

Look and check: A multi-label classification pipeline via multi-agent cooperation

Mingyu Fu¹ , Wei Suo¹, Yukun Li, Lingyan Ran* , Peng Wang

School of Computer Science, Northwestern Polytechnical University, 1 Dongxiang Road, Chang'an District, Xi'an, 710129, Shaanxi, China

ARTICLE INFO

Communicated by X. Zhu

Keywords:

Multi-label classification
Multi-agent cooperation
Large pre-trained models

ABSTRACT

Multi-label classification is an important task in computer vision with many real-world applications. Recent works attempt to leverage rich knowledge from large pre-trained models (e.g., CLIP, ChatGPT) to address the issue by generating diverse label descriptions to enrich semantic information and matching image patches with predefined labels. However, due to obscured and small-sized objects in complex scenarios, existing methods suffer from misclassification and omission. Meanwhile, such methods mostly overlook the role of contextual scenarios, which play an important role in recognizing objects in given images. To address these issues, we propose a new pipeline via multi-agent cooperation. Specifically, the pipeline consists of two stages: Look and Check. In the first stage, a multi-modal large model (e.g., LLaVA) is used as a looker, to search for objects in the image using a group of location-guided instructions. In the second stage, a large language model (e.g., LLaMA) serves as a checker, which is applied to understand the scenario and make inferences to assist the looker. Finally, the predicted results are obtained by summarizing the two stages. Extensive experiments are conducted, demonstrating the superior performance of our method on four widely used benchmarks.

1. Introduction

Multi-label classification, also known as image tagging is a fundamental task in computer vision. Instead of assigning just one label to each image, as in traditional single-label classification, multi-label classification involves predicting a set of labels for each image. Thus, it is more challenging and practical, particularly in complex real-world scenarios. Multi-label classification plays a vital role in numerous real-world applications, including recommendation systems, autonomous driving, and scene understanding.

To recognize multiple objects in the image, some works [1–4] attempt to capture relationships between labels by building graphs or applying graph convolutional networks (GCNs) [5]. Other methods [6–8] explore how to identify regions of interest via attention mechanisms. Recent works [9–13] utilize the strong capabilities of vision and language pre-training (VLP) models (e.g., CLIP [14]) to match different areas in the image with labels in a predefined vocabulary. Among them, RAM++ [13] leverages large language models (e.g., ChatGPT) to produce diverse label descriptions to enrich semantic information.

Although these works have made great progress, they are still restricted by the following challenges: First, such methods struggle to recognize obscured and small-sized objects under complex scenarios. As shown in Fig. 1, only a small part of the window is exposed and it is difficult to recognize for previous methods due to its incomplete display. Second, existing works mostly ignore the influence of contextual scenarios, which play an important role in capturing objects in the given images. As displayed in Fig. 1, the corresponding scenario (i.e., washroom) could assist the model in inferring the reasonable objects (e.g., towel) and filtering out unreasonable items (e.g., pan). Finally, previous methods mostly depend on a specific model to deal with the multi-label classification task. This would inevitably lead to the trained model overfitting the corresponding data distribution and obtaining sub-optimal results. Meanwhile, the label space they use in evaluation covers limited categories (e.g., 80 for the COCO dataset [15]), which restricts their applications in the real world.

To tackle the issues mentioned above, we propose a new classification pipeline called the Multi-Agent Cooperation (MAC) paradigm.

* Corresponding author.

Email addresses: fumy@mail.nwpu.edu.cn (M. Fu), suowei1994@mail.nwpu.edu.cn (W. Suo), yukunli2319@gmail.com (Y. Li), lingyanran@gmail.com (L. Ran), peng.wang@nwpu.edu.cn (P. Wang).

¹ The two authors contribute equally to this work.

<https://doi.org/10.1016/j.neucom.2026.133626>

Received 3 June 2025; Received in revised form 26 January 2026; Accepted 8 April 2026

Available online 12 April 2026

0925-2312/© 2026 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

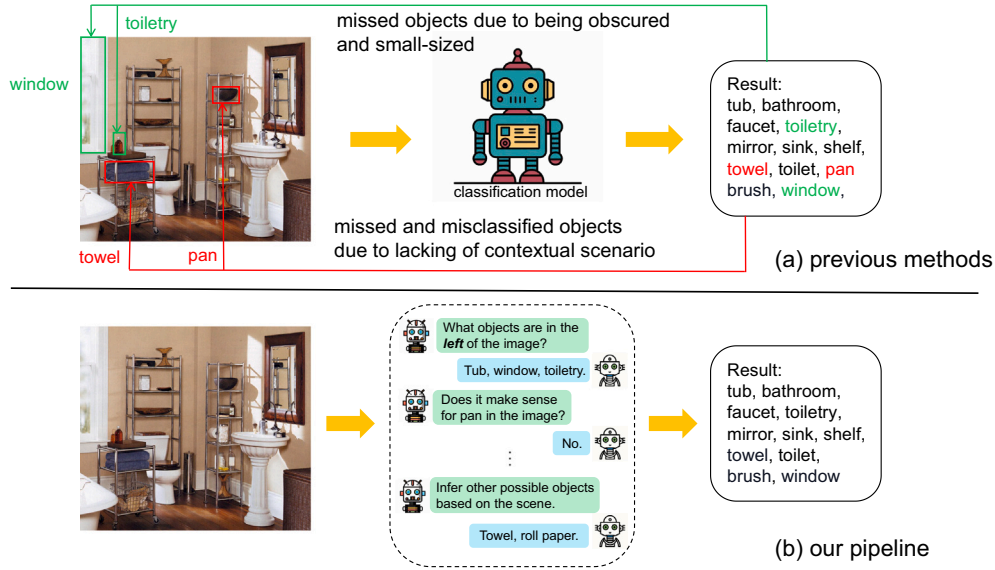


Fig. 1. Comparison of previous methods and our pipeline. (a) Previous methods suffer from omission (e.g., window, towel and toiletary) and misclassification (e.g., pan) due to obscured objects and a lack of contextual scenarios. (b) Instead, our pipeline tackles these issues by utilizing the cooperation of multiple agents.

Different from previous works, our method applies multiple agents to cooperate and complement each other in the classification process. More importantly, a proposed approach is a training-free paradigm without the predefined vocabulary or complex optimization process. Specifically, our pipeline consists of two stages. In the first stage, a Multi-modal Large Language Model (MLLM) is used to look for objects and items in the image. To find obscured and small-sized objects easily, a group of instructions with explicit location guidance is designed to guide the MLLM to focus on every area in the image evenly. In the second stage, to further filter out incorrect objects found by the MLLM and supplement missed items, we adopt a Large Language Model (LLM) to capture the corresponding scenario and make inferences to assist the MLLM. Benefiting from its rich knowledge, the LLM can point out unreasonable objects in the scene and offer proposals about other items possibly present in the image. At last, a list of unreasonable objects and potentially existing items is delivered for the MLLM to verification. Notably, while our pipeline benefits from the strong capacity of large pre-trained models, directly applying an MLLM without location guidance or the assistance of an LLM is suboptimal, given the complexity of real-world scenarios. In addition, we adopt a large-vocabulary evaluation setting with a sufficiently large label space to validate the effectiveness of the proposed pipeline. Overall, the constructed pipeline is simple yet effective. It outperforms previous methods by a large margin and establishes a new state-of-the-art in the large-vocabulary setting on four widely used benchmarks. Meanwhile, the pipeline achieves competitive results in the small-vocabulary setting.

Our main contributions can be summarized as follows:

(1) To the best of our knowledge, we are the first to achieve multi-label classification by constructing multi-agent cooperation. The proposed pipeline is a training-free paradigm without predefined vocabulary or back-propagation.

(2) Our pipeline involves two different stages (i.e., Look and Check). In the first stage, a group of location-guided instructions is designed to search for obscured and small-sized objects. Moreover, an LLM is adopted to capture the corresponding scenario and make inferences to assist with the multi-label classification.

(3) Extensive experiments show that our pipeline significantly surpasses previous methods and establishes a new state-of-the-art on four public benchmarks.

2. Related works

2.1. Multi-label classification

<https://www.sciencedirect.com/science/article/pii/S0926641025001111>

Multi-label classification is a fundamental task in computer vision, aiming to predict the presence of multiple objects, attributes, or concepts within a given image. It arises relatively early because of its practicality in many real world applications. In recent years, multi-label classification has drawn more interest with the development of deep learning. Most methods [8,16–18] use multiple binary classifiers, one for each category, to solve the problem. Among them, Query2Label [17] is the first to use a Transformer decoder architecture in multi-labels classification to query the existence of class label. Ridnik et al. [16] proposed a unified architecture called ML-Decoder for multi-label, zero-shot, and single-label classification, which removes self-attention layers in the decoder and uses a group-decoding scheme to improve efficiency and scalability. Ridnik et al. [18] proposed an asymmetric loss (ASL) to mitigate the negative-positive label imbalance problem when using a binary classifier. They improved focal loss to implement it differently on positive and negative samples. Some works [1–4] try to model label correlations by constructing graphs or using graph convolutional networks (GCN) [5] due to intrinsic semantic dependencies between labels in the image. Other methods [6–8] explore how to locate regions of interest using Long Short-Term Memory (LSTM) [19] networks or attention mechanisms. You et al. [7] raised an adjacency-based similarity graph embedding method and generated cross-modal attention maps from label embeddings. Narayan et al. [8] put forward a bi-level attention module (BiAM) to improve the features by involving both local and global context information.

In the past few years, vision and language pre-training (VLP) models (e.g., CLIP [14] and ALIGN [20]) have made great progress due to their powerful image-text matching ability. Researchers have applied VLP models in a wide range of tasks, including few-shot learning [21,22], continual learning [23,24], open-vocabulary detection [25,26], and open-vocabulary segmentation [27,28]. Recent works in multi-label classification have explored how to utilize VLP models to recognize multiple labels in an image. Sun et al. [9] proposed Dual Context Optimization (DualCoOp) for partial-label and zero-shot multi-label classification, which encodes positive and negative label names as

prompts and uses class specific region feature aggregation after the image encoder of CLIP. MKT [10] adopts knowledge distillation to maintain consistency between image and label embeddings and prompt tuning to improve label embeddings, with a two-stream module to obtain both local and global features. ADDS [29] uses a dual-modal decoder to align visual and textual features and a pyramid-forwarding method to handle high resolution inputs. LMPT [11] is proposed for long-tailed multi-label classification, which adopts general prompts and instance prompts to better align text and image modalities, and a class-specific embedding loss with class-aware soft margin and re-weighting to address imbalanced data distribution. Tag2Text [30] integrates image tagging and image captioning into a complete framework and uses tags parsed from text to supervise the classifier. RAM [12] enhances Tag2Text's open-set classification ability by introducing CLIP into the framework. RAM++ [13] further improves the open-set capacity with large language models to produce diverse label descriptions used during the training phase.

Although existing methods have achieved great progress, they have three main problems. First, previous methods suffer from misclassification and omission due to sheltered and small-sized objects in complex scenes. Second, the impact of contextual scenarios has been ignored, which is important in the classification process. Finally, existing works are restricted to a limited label space, which impairs their applications in these real world. By contrast, our pipeline can tackle the issues through the collaboration of multiple agents and achieve impressive performance in a large-vocabulary evaluation setting.

2.2. Large language models

Large Language Models (LLMs) are a category of artificial intelligence models that leverage advanced natural language processing techniques, with the capacity to understand, generate human-like text and follow human's instructions to solve problems. LLMs have profoundly influenced the field of natural language processing (NLP) and other industries in the past few years. LLMs are typically pre-trained on vast amounts of diverse and unlabeled text data. A copious volume of training data gives the LLMs the ability called "emergent capabilities", which means they often exhibit capabilities that weren't explicitly programmed but emerge from their understanding of pre-training data. Thus, LLMs are adept at transfer learning, where they can leverage knowledge gained from pre-training on large datasets to perform well on specific downstream tasks with relatively smaller labeled datasets in fine-tuning. Instruction tuning and reinforcement learning with human feedback (RLHF) further boost LLMs' ability to follow human's instructions. As a result, LLMs like ChatGPT, GPT-4, LLaMA [31], Alpaca [32], Falcon [33] and ChatGLM [34] showcase impressive intelligence in understanding and generation capabilities, pushing artificial intelligence a large step forward. The development of LLMs also energizes other areas. However, few researchers have explored utilizing LLMs in multi-label classification. To our knowledge, we are the first to explore the cooperation of LLMs and MLLMs (Multi-modal Large Language Models) in this field.

2.3. Multi-modal large language models

Multi-modal Large Language Models (MLLMs) represent a cutting-edge fusion of natural language processing and multi-modal capabilities, enabling these models to seamlessly comprehend and generate information across diverse data types including image, video and audio. Unlike traditional language models, MLLMs can integrate textual information with visual and auditory inputs, offering a more holistic understanding of content. Typically built upon transformer architectures, MLLMs demonstrate unprecedented performance in tasks such as image recognition, image captioning, and visual question answering. Leveraging their immense scale and pre-training on extensive datasets, MLLMs can capture intricate relationships between words and visual

elements, demonstrating versatility in a range of applications. MLLMs such as BLIP-2 [35], InstructBLIP [36], LLaVA [37], GPT-4 Vision and MiniGPT-4 [38] have attracted more and more researchers to study and make use of them in a variety of downstream tasks. Although MLLMs obtain impressive results in a series of computer vision tasks, their performance is still limited under complex scenarios in multi-label classification due to obscured and small-sized objects. Thus a group of instructions containing location guidance is designed for the MLLM to mitigate misclassification and omission in our pipeline.

3. Methods

3.1. Preliminary

Consider a set of image samples $X = \{x_1, x_2, \dots, x_N\}$, where each instance x_i is associated with a set of labels $y_i \subseteq Y$. Here, $Y = \{c_1, c_2, \dots, c_M\}$ is the label space containing all M categories. To solve the multi-label image classification problem, the model needs to learn a mapping function $f : X \rightarrow 2^Y$. In this paper, we utilize multiple agents to cooperate and complement each other to solve the problem. Unlike previous methods, it is implemented without a training phase and a predefined vocabulary.

3.2. The overall pipeline

As shown in Fig. 2, MAC consists of two stages: Look stage and Check stage. For a given image x_i , an MLLM serves as a looker to search for objects. Meanwhile, an LLM acts as a checker to assist the MLLM. With the collaboration of the looker and checker, the final result y_i can be obtained through multiple rounds of dialogue. In the first stage, the MLLM is tasked with looking for objects in the image. To facilitate the identification of obscured and small-sized objects, a group of instructions is designed to guide the MLLM. Considering that the contextual scenario is important for filtering out incorrect labels and supplementing other objects that may potentially exist, we adopt an LLM to capture the scenario and make inferences to assist the MLLM in the second stage. A set of templates is designed for the LLM to understand the scenario, point out unreasonable objects and provide other items. Since the LLM cannot directly interact with the image, the proposals made by it are sent to the MLLM for verification. Specifically, we employ a template to ask the MLLM to confirm whether these 'unreasonable' objects and potentially existing items are present in the image. Finally, the result is obtained by summarizing the dialogue from the two stages.

In this work, we adopt a static, two-agent architecture with predefined roles (i.e., Looker for visual perception and Checker for reflection and reasoning) for the following reasons: (a) Our primary objective is to investigate the effectiveness of the multi-round dialogue and reflection mechanism in multi-label classification. By fixing both the number and roles of agents, we can more clearly attribute the observed performance gains to the proposed collaboration mechanism, rather than to dynamic team formation strategies. (b) When addressing the multi-label image classification task, humans naturally decompose it into two sequential subtasks: (1) comprehensive visual discovery and (2) reflection upon and validation of the discovered results. Our design mirrors this human cognitive process: the Looker is responsible for thoroughly searching for objects in the image, while the Checker identifies implausible predictions and supplements potentially missing items. This explicit and static division of labor not only provides good interpretability but also serves as a foundational baseline.

3.3. Stage one: look via MLLM

In the first stage, an MLLM is adopted to search for objects in the image. Inspired by Zhu et al. [39], we use a prompting mechanism for the MLLM, including a task prompt, dialogue history and an answer prompt. First, the task prompt is delivered to the MLLM at the beginning of the dialogue to introduce the task and the requirements. Then,

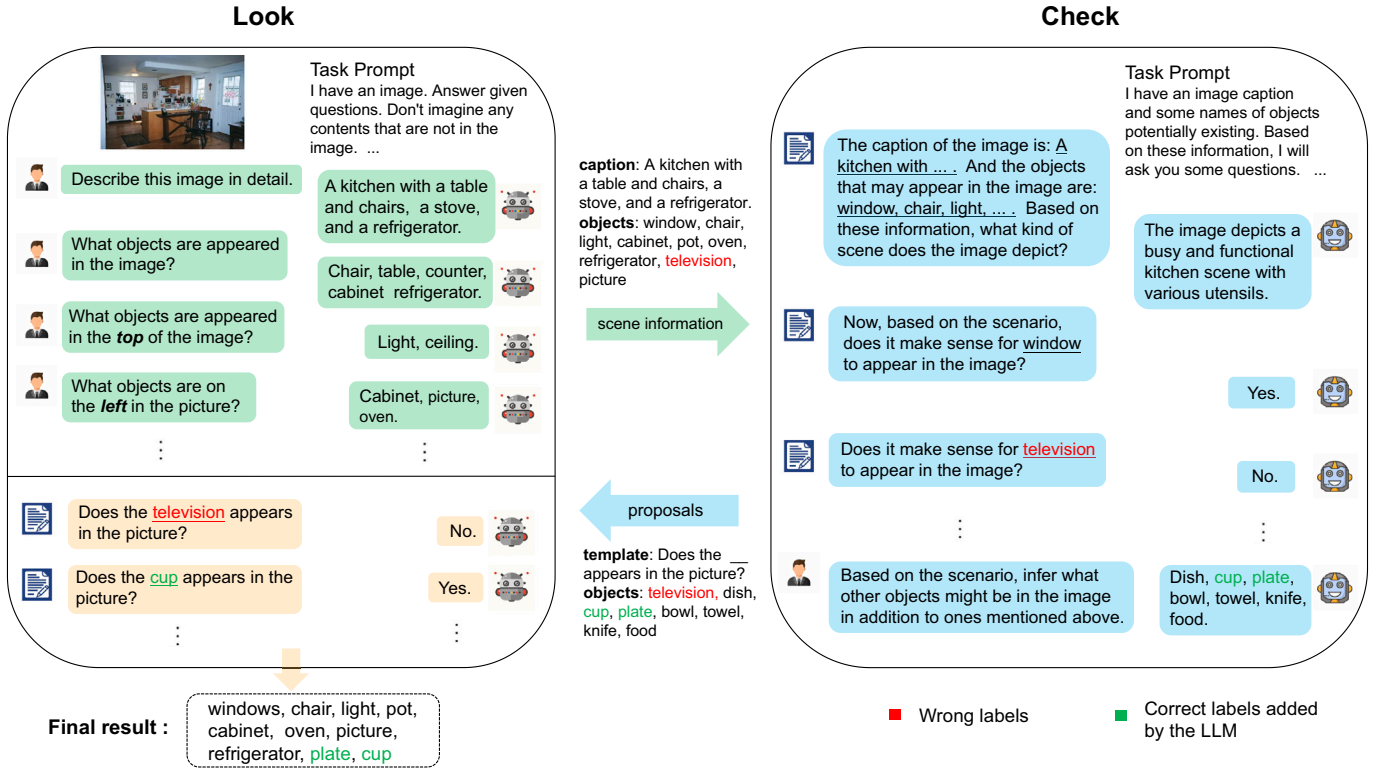


Fig. 2. Illustration of the overall pipeline. Our pipeline consists of two stages: Look stage and Check stage. In the first stage (in green), we design a group of instructions with location guidance for an MLLM to recognize obscured and small-sized objects. In the second stage (in blue), for filtering out incorrect labels and supplementing potential objects, an LLM is adopted to capture the corresponding scenario via the scene information (summarized from the first stage) and make inferences to assist the MLLM. Then proposals made by the LLM are sent to the MLLM for verification (in orange). The final result can be obtained by summarizing the two stages.

we design a group of instructions in a hierarchical structure inspired by MKT [10]. The instructions have three components: a caption question, general questions and location-guided questions. To be specific, we ask the MLLM with a caption question: “Describe this image in detail.” to get an image caption as the scene information for the second stage. And some general questions asking the MLLM to look for objects are used, e.g., “What objects appear in the image?” By the above instructions, a base label set could be obtained, i.e., kitchen, table, chair, ..., as shown on the left side (in green) of Fig. 2. Since there are many objects occluded or of small size under complex scenarios, instructions only including general questions lead to the omission of items. Therefore, an image is further divided into four areas: left, right, top and bottom, for brevity and clarity. Some questions with location guidance are employed as follows: “What objects appear at the top of the image?”, “What objects are on the left in the picture?” ... All aforementioned question-answer pairs are stored in the dialogue history, which allows the MLLM to know the found objects and avoid repeated answers. Finally all answers from the MLLM are summarized as the scene information and delivered to the LLM for understanding the scenario after the first stage.

3.4. Stage two: check via LLM

Although many obscured and small-sized objects are identified after the first stage, there are still omissions and misclassifications due to the lack of necessary contextual scenarios. To tackle the issue, an LLM is used to understand the corresponding scenario and make inferences in the second stage. First we introduce the task and requirements to the LLM in a task prompt. A set of templates is employed to interact with the LLM, as shown on the right side (in blue) of Fig. 2. Since the LLM cannot directly interact with the image, the scene information

summarized from the Look stage is filled in the first question in the templates: “The caption of the image is: And the objects that may appear in the image are: Based on this information, what kind of scene does the image depict?” Here we do not employ a scene classifier [40–43] to capture the corresponding scenario. This is because such classifiers are limited to predefined categories during the training phase which weakens their generalization ability in the real world, as shown in the Experiments.

Based on the captured scenario, the LLM is able to filter out incorrect labels by judging which objects are unreasonable to appear in the scene. In particular, a template is adopted as follows: “Does it make sense for to appear in the image?”. As demonstrated in the right side (on blue) of Fig. 2, the LLM could infer that a television is usually not found in the kitchen with the answer “No”. On the other hand, the LLM is required to supplement other objects potentially existing. As shown in the Fig. 2, the LLM provides a list of items relevant to the kitchen scene with the instruction: “Based on the scenario, infer what other objects might be in the image in addition to the ones mentioned above.”.

Then, we notice that some works [44,45] point out that the pre-trained knowledge of deep learning models limits their performance due to inherent biases in the training data. Therefore the proposals made by the LLM (containing “unreasonable” objects and potential items) are sent to the MLLM for verification. We take the verification phase as part of the Check stage since it is intended for validating the inferred result of the LLM. As shown in the left side (on orange) of Fig. 2, the MLLM is asked to confirm whether these objects are in the image, using the template: “Does the appear in the picture?”. As a result, some objects (e.g., cup, plate) not recognized in the Look stage could be found. Finally, the result is summarized from two parts. One part consists of the reasonable objects with the LLM answering “Yes” in the Check stage (the blue

Table 1

Comparison of our pipeline with state-of-the-art methods on COCO, ADE-20K, Pascal Context and Pascal VOC in a **large-vocabulary** setting. The results are reported in terms of Overall Precision (OP), Overall Recall (OR) and Overall F1 Score (OF1). All metrics are in %.

Methods	COCO			ADE-20K			Pascal Context			Pascal VOC		
	OP	OR	OF1	OP	OR	OF1	OP	OR	OF1	OP	OR	OF1
Tag2Text [30]	18.5	30.5	23.0	24.5	23.9	24.2	16.4	35.3	22.4	10.0	64.6	17.3
RAM [12]	27.3	28.5	27.9	34.8	22.7	27.5	23.7	31.2	26.9	15.8	64.8	25.4
RAM++ [13]	27.6	28.5	28.0	34.9	22.3	27.2	24.4	31.2	27.4	16.2	64.8	25.9
LLaVA (Baseline)	31.2	28.8	30.0	33.0	28.0	30.3	28.3	28.9	28.6	16.4	56.6	25.4
MAC (Ours)	32.8	38.0	35.2	34.7	36.2	35.4	28.8	39.5	33.3	19.4	88.2	31.8

part of Fig. 2). The other part consists of the objects with the MLLM answering “Yes” in the verification phase (the orange part of Fig. 2).

4. Experiments

4.1. Experimental settings

4.1.1. Benchmarks

To validate the capacity of our pipeline to recognize multiple objects under complex scenarios, we choose four datasets as our benchmarks: COCO [15], ADE-20K [46], Pascal Context [47] and Pascal VOC 2007 [48]. COCO and ADE-20K are widely used in object detection and segmentation. Recently both datasets have become popular in the multi-label classification task [9,12,16,17,29,30]. Following [12], the validation set of the COCO 2017 version with 5000 images containing 133 categories is chosen. For ADE-20K, 2000 images from the validation set with 150 semantic categories are used in evaluation. Pascal Context is an extension of the Pascal VOC dataset and is built for semantic segmentation, with 459 categories in total. Following [47,49], we use 59 categories with the highest frequency as semantic labels and label the remaining categories as background. A total of 5105 images in the validation set are adopted. Pascal VOC 2007 is a popular dataset in multi-label classification, which has 20 categories. We use the test set containing 4952 images as our benchmark.

4.1.2. Evaluation

On four benchmarks, we report the Overall Precision (OP), Overall Recall (OR) and Overall F1 score (OF1) following [50].

The three metrics we used are defined as follows:

$$OP = \frac{\sum_m N_m^{tp}}{\sum_m N_m^{p}}, \quad OR = \frac{\sum_m N_m^{tp}}{\sum_m N_m} \quad (1)$$

$$OF1 = \frac{2 \times OP \times OR}{OP + OR} \quad (2)$$

in which N_m^{tp} is the number of true positive samples for each class m . N_m^p refers to the number of positive predictions per class and N_m represents the number of ground truth samples of class m .

For fair and convenient comparison, we employ a large-vocabulary setting for evaluation. In particular, following RAM (Recognize Anything Model) [12], we adopt a label space that contains 4585 categories (i.e., $M = 4585$) spanning objects, scenes, attributes, and actions collected from various sources and covering most of the objects in daily life. Given the breadth and diversity of the label space, we consider this setting a reasonable approximation to the open-world setting for practical purposes. For the comparing methods that adopt a decoder-style architecture, all labels in the label space are used as candidate queries. Since the responses of the agents are open-ended in our pipeline, the label space is used to filter the object list of each image. Therefore, adopting a consistent label space can align the output space of our pipeline and the comparing methods. Notably, we use the label space solely for the convenience of calculating evaluation metrics and making comparisons.

The label space is not essential in inference to avoid limiting the generalization of our method in certain scenarios (e.g., when recognizing rare species that fall outside the scope of the label space).

4.1.3. Implementation details

In our main experiments, LLaVA and LLaMA-2 are adopted in the first and second stage respectively. For LLaVA, we employ the LLaVA-v1.5 with a Vicuna-7B model and a ViT-L/336px model from CLIP [14]. For LLaMA-2, the 7b-chat version is chosen. All images are resized to 336×336 resolution. The instructions in the first stage include a caption question, three general questions and four location-guided questions (top, bottom, left and right). The dialogue history has the format as “Question: { } Answer: { }”. To obtain an object list after the first stage, SceneGraphParser [51] is utilized to parse the dialogue history of the MLLM and all nouns are collected. Specifically, we first instruct the MLLM to report only the objects present in the image, without introducing any additional terms, to prevent interference from non-object nouns, using the prompt: “Your answers should be brief and to the point. Directly state what objects are present in the image without adding any qualifiers or modifiers.” Then we apply the SceneGraphParser to parse each response of the MLLM. Owing to the strong parsing capability of the SceneGraphParser, the “lemma_head” field in the “entity” section of the parsing results captures all nouns in the responses and automatically normalizes them to singular, lowercase forms. All extracted nouns are subsequently consolidated by applying WordNet from the Natural Language Toolkit (NLTK) [52] to identify synonyms following [39]: any pair of nouns with a Wu–Palmer similarity [53] greater than 0.9 is treated as equivalent and merged. This procedure yields the final object list for the LLM to check. In evaluation, we use the label space to filter the result list of each image. To be specific, tags within the label space are retained. For those beyond the label space, we use WordNet to find their synonyms in the same manner as mentioned earlier.

4.2. Comparison with state-of-the-art methods

We compare our pipeline with state-of-the-art methods in Table 1. For fair and convenient comparison, we choose comparing methods that are pre-trained on large-scale data and evaluated on downstream datasets directly. Among the methods, RAM++ [13] achieves the state-of-the-art, which aligns image-text and image-tags in a framework with image-tags-text triplets. Tag2Text [30] unifies image tagging and captioning into a whole framework. RAM [12] introduces open-vocabulary recognition into Tag2Text via CLIP and adopts distillation to align image-text features. ‘LLaVA (Baseline)’ is our baseline which means applying LLaVA with instructions to output the objects in the image without location guidance and the assistance of LLM. Different from these methods depending on a single model, our pipeline utilizes multiple agents cooperating with each other. As a result, our pipeline (MAC) outperforms RAM++ by a large margin on the COCO dataset, with an absolute gain of 5.2% on OP, 9.5% on OR and 7.2% on OF1. Compared with LLaVA (Baseline), MAC achieves an improvement of 1.6% on OP, 9.2% on OR and 5.2% on OF1. Both improvements in

Table 2

Impact of different components in our pipeline. ‘Look’ represents the Look stage in the pipeline. ‘caption’ means the image caption question (i.e., “Describe this image in details.”). ‘general’ means general questions (e.g., “What objects appear in the image?”). ‘guidance’ means location-guided questions (e.g., “What objects are on the left in the picture?”). ‘Check’ represents the Check stage. **Notably, the second row is our baseline.** ‘Total Tokens’ denotes the sum of input tokens and output tokens for each sample. ‘Rel. Latency’ means the multiple of inference time compared to the baseline.

Components				COCO			Inference Cost			
caption	Look general	guidance	Check	OP	OR	OF1	MLLM Calls	LLM Calls	Total Tokens	Rel. Latency
✓				30.5	19.9	24.1	1	0	637.7	0.77 ×
✓	✓			31.2	28.8	30.0	2	0	1223.9	1.0 ×
✓	✓	✓		32.4	30.3	31.3	6	0	3568.3	2.13 ×
✓	✓		✓	32.5	36.3	34.3	5.7	4.2	3485.8	2.87 ×
✓	✓	✓	✓	32.8	38.0	35.2	9.1	5.4	5496.7	3.61 ×

precision and recall validate the role of our proposed location-guided instructions and applying an LLM to capture the scenario, showing that applying large pre-trained models in multi-label classification is not trivial. Moreover, MAC surpasses the methods consistently on ADE-20K, Pascal Context and Pascal VOC, with an absolute gain of around 5% on OF1. The consistent improvements across four benchmarks demonstrate the effectiveness of our pipeline in the large-vocabulary evaluation setting.

4.3. Ablation studies

To explore the impact of each component, we conduct ablation studies with different stages in the pipeline and different instructions in the Look stage in Table 2. The experiments are conducted on COCO [15] due to its wide usage and challenge. LLaVA and LLaMA-2 are adopted in the Look stage and Check stage respectively. In the first row of the table, the pipeline contains only the Look stage with an image caption question as the instruction. The caption contains few objects in the image, leading to a much lower OR. The second row is our baseline which has caption and general questions as the instructions in the Look stage. Benefiting from the strong capacity of MLLMs, general questions significantly improve OR. As shown in the third line of Table 2, the introduction of location-guided questions boosts both OP and OR, with an absolute gain of 1.2% and 1.5%. This proves that location guidance could compel the MLLM to focus on different areas of the image and recognize more objects under complex scenarios. Further, the Check stage improves the performance in both OP and OR (in the fourth line of Table 2). This can be attributed to the important role of contextual scenarios in the classification process. Specifically, since the LLM could capture the corresponding scene, it can then infer to point out unreasonable objects and provide possible items, reducing misclassifications and omissions. Finally, combining location guidance and the assistance of the LLM, our whole pipeline surpasses the baseline by a large margin, 1.6% on OP, 9.2% on OR and 5.2% on OF1.

4.4. Efficiency and cost analysis

We measure the inference cost of different components in our pipeline in Table 2. Here we report proxy metrics for inference cost, including the average number of MLLM/LLM calls, total token consumption, and relative latency per sample, which are highly correlated with actual computational overhead in multi-agent systems. As shown in Table 2, our baseline only involves two MLLM calls with 1223.9 total tokens, where the token consumption mainly stems from the visual tokens of the input prompt (576 tokens for LLaVA [37]). The location-guided instructions include four questions to query four areas of an image (i.e., left, right, top and bottom). Thus, the number of MLLM calls increases to six with $2.13 \times$ latency compared to the baseline, as shown in the third row. When the Check stage is applied, our pipeline reaches an OF1 score of 35.1%, accompanied by $7.25 \times$ total calls, $4.49 \times$ token consumption and $3.61 \times$ latency compared to the baseline. Despite the

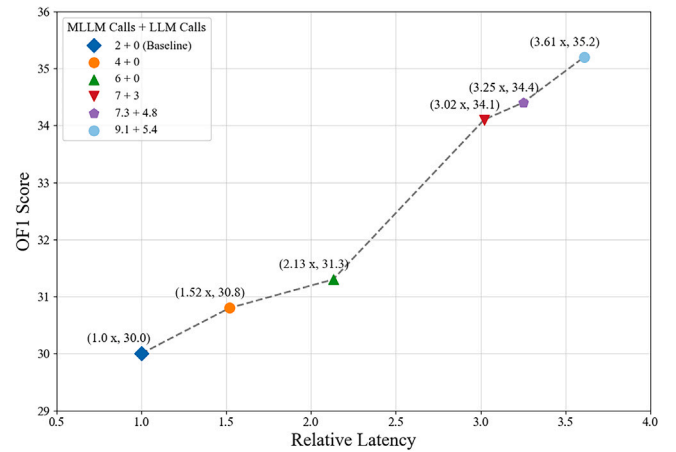


Fig. 3. Performance-Cost trade-off curve. We demonstrate a trade-off curve between OF1 score and average relative latency per sample on COCO, by adjusting the number of MLLM/LLM calls. The ‘4 + 0’ point denotes the variant that has 4 MLLM calls, having two location-guided (i.e., center and edge) questions, without the Check stage. The ‘7 + 3’ point represents that we check and verify All objects via a single call, respectively, with 7 MLLM calls and 3 LLM calls. The ‘7.3 + 4.8’ point corresponds to the variant with two two location-guided questions, leading to 7.3 MLLM calls and 4.8 LLM calls. The other two variants can be found in Table 2.

increased computational cost, this trade-off is acceptable in scenarios where accuracy and reliability are prioritized over strict real-time constraints. In Section 5, we discuss potential optimization directions to improve the efficiency of our method, thereby enhancing its practicality in latency-sensitive scenarios.

In Fig. 3, we show a trade-off curve between performance (OF1 score) and computational overhead (relative latency), achieved by adjusting the number of MLLM/LLM calls within the pipeline. As shown in the first three points in Fig. 3, fine-grained location-guided questions improve performance, while incurring more MLLM calls and higher latency ($2.13 \times$). After the Check stage is applied, some unreasonable objects and missing items can be inferred and further verified. Thus, an obvious improvement can be observed (OF1 score increasing from 31.3% to 34.1%), which validates the effectiveness of our designed multi-round dialogue and reflection mechanisms, with relatively higher latency (from $2.13 \times$ to $3.02 \times$). After we further check and verify each object in a single call, our pipeline achieves an OF1 score of 35.2%, with $3.61 \times$ latency. Although our pipeline incurs higher latency compared to the baseline, the inference cost is explicitly controllable via the number of MLLM/LLM calls, enabling a clear performance-cost trade-off. As shown in Fig. 3, the method remains practical under moderate budgets, while higher budgets can be employed when accuracy is prioritized.

Table 3
Different combinations of MLLMs and LLMs in our pipeline.

MLLM + LLM	COCO		
	OP	OR	OF1
LLaVA + Vicuna	32.5	35.3	33.8
LLaVA + ChatGLM2	32.2	33.6	32.9
BLIP-2 + LLaMA-2	34.4	34.4	34.4
InstructBLIP + LLaMA-2	37.7	32.2	34.7
LLaVA + LLaMA-2 (default)	32.8	38.0	35.2

4.5. Alternative settings

To examine how different agents influence our pipeline, we explore five combinations of MLLMs and LLMs in Table 3. For the LLM component, we examine Vicuna [54], ChatGLM2 [34] and LLaMA-2 [55]. Among them, LLaMA-2 demonstrates stronger instruction-following ability and larger model capacity, enabling it to generate more accurate proposals that assist the MLLM in both removing incorrect labels and supplementing missing objects. Consequently, LLaMA-2 achieves consistently higher OP and OR compared to ChatGLM2 and Vicuna. For the MLLM component, we explore BLIP-2 [35], InstructBLIP [36] and LLaVA [37]. InstructBLIP extends BLIP-2 with instruction tuning, which substantially enhances its ability to follow our customized prompts during the verification phase. This leads to more precise filtering of incorrect labels and, in turn, a higher OP (37.7%) relative to both BLIP-2 and LLaVA. LLaVA employs an MLP-based multimodal connector that produces 576 visual tokens, whereas BLIP-2 and InstructBLIP rely on the Q-Former and generate only 32 visual tokens (i.e., query embeddings). The increased number of visual tokens enables LLaVA to preserve more fine-grained visual information, which in turn enhances its ability to recognize a greater number of objects, resulting in a higher OR (38.0%). Overall, the ‘LLaVA + LLaMA-2’ combination achieves the best OF1 score (35.2%), and we adopt it as the default configuration. More broadly, differences in architectural design and training regimes introduce characteristic biases across model combinations within the pipeline. Therefore, model selection should be aligned with the task’s primary objective. For example, InstructBLIP is more suitable when reliability and precision are paramount, whereas LLaVA is preferable when maximizing recall is the priority.

To study the role of dialogue history and location guidance, we conduct experiments with different settings in the Look stage. As shown in the first row of Table 4, the pipeline without dialogue history results in an obvious drop in OP and OR. Since the MLLM could not know the objects already found without the dialogue history, the same objects will be repeated in the answers, which limits the MLLM ability to capture more objects in the image. Guidance with two locations (center and edge) leads to a lower OR than the default setting with four locations (top, bottom, left and right). We believe fine-grained guidance helps the MLLM to capture more objects that are small-size or obscured in the image.

Moreover, we try another source of the scene information. In the default setting, the image caption and object list summarized from the Look stage are adopted as the scene information. Here, a scene classifier [40] is used to predict the corresponding scenario and provide scene information for the LLM. As shown in the third row of Table 4, the scene information from the scene classifier leads to a drop in both OP and OR compared to the default setting. The scene classifiers pretrained on predefined categories could not provide a detailed and accurate description of the image. Therefore the LLM could not adequately understand the scenario and make inferences to assist the MLLM.

To validate the robustness of our method, we test the sensitivity of the pipeline to the specific prompts used in Table 5. Specifically, we choose the location guided prompt (i.e., “What objects appear at the top of the image?”) in the Look stage, the check prompt (i.e., “Does it make sense for __ to appear in the image?”) and the verification

Table 4

Alternative settings in our pipeline. “without dialogue history” means the MLLM could not see previous question-answer pairs in the dialogue of the Look stage. “guidance with two locations” means the instructions of the first stage contain two location-guided questions, i.e., “What objects are in the **center** of the picture?” and “What objects are on the **edge** of the image?”. “with scene classifier” means a scene classifier is adopted to predict the corresponding scenario and the prediction is used as the scene information for the Check stage. ‘default’ means our default setting.

Alternative settings	COCO		
	OP	OR	OF1
without dialogue history	31.2	37.0	33.9
guidance with two locations	32.3	36.8	34.4
with scene classifier	32.2	36.5	34.2
default	32.8	38.0	35.2

Table 5

Sensitivity of our pipeline to prompts. We replace the location guided prompt, check prompt and verification prompt with two variants that convey the same meaning, respectively. **Location guided prompt A**: “What items are visible in the upper portion of the image?”. **Location guided prompt B**: “Which objects can be found at the top of the image?”. The underlined content here can be replaced with other nouns for location, e.g., left, right and bottom. **Check prompt A**: “Is it reasonable for __ to be shown in the image?”. **Check prompt B**: “Does the presence of __ in the image seem appropriate?”. **Verification prompt A**: “Is __ visible in the picture?”. **Verification prompt B**: “Can __ be seen in the image?”.

Prompt variants	COCO		
	OP	OR	OF1
Location guided prompt A	32.9	37.7	35.1
Location guided prompt B	32.7	38.1	35.2
Check prompt A	33.1	37.9	35.3
Check prompt B	32.5	37.5	34.8
Verification prompt A	32.7	37.4	34.9
Verification prompt B	32.6	37.9	35.1
default	32.8	38.0	35.2

prompt (i.e., “Does the __ appear in the picture?”) in the Check stage. For each of these prompts, we create two alternative variants that convey the same meaning and replace them with the variants, respectively. For example, the location guided prompt variants are “What items are visible in the upper portion of the image?” and “Which objects can be found at the top of the image?”. As shown in Table 5, different ways of expressing these prompts lead to only marginal differences in performance (OF1 score ranging from 34.8% to 35.3%). Owing to the strong capacity of both the MLLMs and LLMs, our agents can reliably understand diverse prompt formulations and execute the required instructions. As a result, our pipeline exhibits high robustness to prompt wording variations.

4.6. Discussion

To validate the generalization of the pipeline, we also conduct experiments in the small-vocabulary setting on four benchmarks. In this setting, the category list of each benchmark is used as the corresponding label space, for example, the 150 categories for ADE-20K. As shown in Table 6, MAC attains comparable OF1 scores across the four benchmarks, yet exhibits lower OR on COCO, ADE-20K, and Pascal Context. The differing performance observed under the large-vocabulary and small-vocabulary settings is primarily attributed to architectural differences among the methods. All comparing methods (Tag2Text [30], RAM [12] and RAM++ [13]) adopt a decoder-style architecture that explicitly queries each category in a predefined label list. This design performs effectively in the small-vocabulary setting with a limited label space (e.g., 80 COCO classes), enabling these methods to systematically verify each category and thus achieve higher recall. In contrast, the agent

Table 6

Comparison of our pipeline with state-of-the-art methods on COCO, ADE-20K, Pascal Context and Pascal VOC in a **small-vocabulary** setting. The results are reported in terms of Overall Precision (OP), Overall Recall (OR) and Overall F1 Score (OF1). All metrics are in %.

Methods	COCO			ADE-20K			Pascal Context			Pascal VOC		
	OP	OR	OF1	OP	OR	OF1	OP	OR	OF1	OP	OR	OF1
Tag2Text [30]	65.2	53.0	58.5	49.6	54.3	51.8	60.2	57.3	58.7	70.4	85.8	77.3
RAM [12]	66.8	51.3	58.0	54.4	49.1	51.6	63.7	52.9	57.8	79.2	87.7	83.2
RAM++ [13]	66.8	51.6	58.2	55.3	48.6	51.7	63.9	53.3	58.1	79.6	87.5	83.4
MAC (Ours)	69.0	48.6	57.0	55.8	43.5	48.9	68.1	50.7	58.1	82.3	88.3	85.2

Table 7

Comparison of our pipeline with the variant in which two agents engage in direct dialogue on four benchmarks. ‘Chat directly’ denotes the aforementioned variant, where an LLM generates questions and an MLLM answers.

Methods	COCO			ADE-20K			Pascal Context			Pascal VOC		
	OP	OR	OF1	OP	OR	OF1	OP	OR	OF1	OP	OR	OF1
Chat directly	35.5	18.8	24.6	39.9	18.1	24.9	32.8	23.2	27.2	18.6	41.2	25.6
MAC (Ours)	32.8	38.0	35.2	34.7	36.2	35.4	28.8	39.5	33.3	19.4	88.2	31.8

responses in MAC are open-ended and numerous long-tail concepts may appear. When a small label space is used to post-process the output of MAC, categories with ambiguous semantics are more likely to be filtered out. As a result, our method attains lower recall in the small-vocabulary evaluation setting. Overall, our approach shows clear advantages in the large-vocabulary setting while maintaining competitive performance in the small-vocabulary scenario, which demonstrates its robustness across different evaluation regimes.

Our pipeline consists of the Look stage and the Check stage, in which the MLLM and the LLM do not have a direct dialogue, but collaborate with each other via the scene information and the designed instructions. To decouple the impact of the proposed collaboration mechanism from model capacity, we explore a setting in which an MLLM and an LLM engage in direct dialogue to recognize objects in the image inspired by [39]. In this way the LLM is responsible for generating questions to explore the image content and the MLLM is tasked with answering the given questions in each round. The final result can be obtained by summarizing the whole dialogue. For fair comparison, we keep the same dialogue rounds as those in MAC. However, the result is unsatisfactory, as shown in Table 7. MAC surpasses chat directly by a large margin on OR and achieves an obvious improvement of more than 6% on OF1 across four benchmarks. We observe that there are two main reasons. The first is that the LLM tends to pay more attention to details of objects already recognized such as color and texture, which is not our main focus. As a result, the generated questions could not compel the MLLM to find more objects in limited dialogue rounds. The second is that the LLM usually generates questions based on the objects already found. This amplifies the influence of false positive predictions and incurs more hallucinations [56–58] in the responses of the MLLM. By contrast, our pipeline adopts a group of designed instructions to restrict the MLLM and employs an LLM to reflect on and reason over identified items, achieving superior performance, which indicates that the observed performance gains of MAC are not merely a consequence of increased model capacity, but arise from the proposed collaboration mechanism.

4.7. Qualitative assessment

For visually understanding the effectiveness of our pipeline, we visualize the predictions of a few samples from ADE-20K in Fig. 4. MAC (Look and Check) consistently predicts more correct labels than RAM++ [13] under different scenes. We observe that the Look stage could achieve better results compared to RAM++. For example, the first sample of Fig. 4, depicts a modern kitchen with lots of supplies.

RAM++ could only recognize a few items (*i.e.*, cabinet, kitchen and vegetable) due to obscured and small-sized objects in such a complex scenario. However, with the explicit location guidance, the Look stage of MAC (Look only) could focus on different areas of the image and find objects that are occluded or small in size: window, plate, stove and sink. With an LLM to capture the scenario and make inferences, MAC (Look and Check) further discovers some items by utilizing contextual information (*e.g.*, a pot on the stove, bowls and dishes in the cabinet).

5. Limitations and future work

Despite the promising results, our work has several limitations that warrant further investigation. In this section, we discuss these limitations and outline potential directions for future work. **1) Inference Efficiency.** Our method currently incurs relatively high latency and token cost due to multi-round dialogues, which hinders its application in resource-constrained scenarios. Meanwhile, multi-turn dialogues may be redundant, which is not explicitly optimized in this work. Recent works, such as AgentDropout [59], propose eliminating redundant interactions in multi-agent systems by dynamically evaluating and removing agents with low contribution. Similarly, we could adaptively control the dialogue budget (*i.e.*, the number of MLLM/LLM calls) based on the difficulty level of the samples. In addition, since the token consumption of our pipeline mainly stems from visual tokens (95.4%), integrating visual token pruning strategies [60,61] into our framework is a promising optimization direction. **2) Agent Initialization.** While our static multi-agent framework with predefined roles effectively validates the benefits of iterative dialogue and reflection mechanisms, how the team structure itself influences collaborative effectiveness has not been investigated. Recent studies, such as AgentInit [62], dynamically initialize agent teams with task-specific expertise and diversity, which has emerged as a promising direction for improving both collaboration efficiency and effectiveness. A natural future direction is to combine these two paradigms by first constructing a task-aware expert team through dynamic initialization, and then applying our multi-round dialogue and reflection mechanism to facilitate deep and efficient collaboration. This integration would result in a complete and efficient multi-agent pipeline spanning from team formation to team collaboration. **3) Towards True Open-World Evaluation.** While our current experiments are conducted under a large-vocabulary setting, a natural and important future direction is to evaluate the proposed method in a true open-world scenario across a broader range of benchmarks and tasks. Exploring such a setting would provide a more comprehensive understanding of the generality and scalability of our approach.



Fig. 4. Comparison of predictions between our pipeline and RAM++ [13] on ADE-20K. Look only represents the Look stage of the pipeline. Look and Check refers to our whole pipeline (MAC). True positive samples are shown in green and false positive samples are shown in red. Labels in black are relevant ones but not included in the ground truth.

6. Conclusions

We propose a simple yet effective pipeline via the cooperation of multi-agents for multi-label classification, called Multi-Agent Cooperation (MAC). The pipeline consists of two stages: Look and Check. In the first stage, an MLLM identifies objects in the image with explicit location guidance to find obscured and small-sized objects. In the second stage, to filter out unreasonable labels and supplement other possible objects, an LLM is used to capture the corresponding scenario and make inferences to assist the MLLM. Our pipeline surpasses state-of-the-art methods significantly on four widely used benchmarks in a large-vocabulary setting, showing its effectiveness under complex scenarios. To our best knowledge, this is the first work to study how to make use of multiple agents in multi-label classification and we hope it could encourage more researchers to explore such a new paradigm.

CRedit authorship contribution statement

Mingyu Fu: Writing – original draft, Methodology, Investigation. **Wei Suo:** Writing – review & editing, Supervision, Methodology. **Yukun Li:** Validation, Software. **Lingyan Ran:** Writing – review & editing, Supervision. **Peng Wang:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62476226).

Data availability

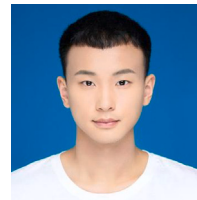
Data will be made available on request.

References

- [1] Z.-M. Chen, X.-S. Wei, P. Wang, Y. Guo, Multi-label image recognition with graph convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5177–5186.
- [2] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: a real-world web image database from national university of Singapore, in: Proceedings of the ACM International Conference on Image and Video Retrieval, 2009, pp. 1–9.
- [3] Y. Wang, D. He, F. Li, X. Long, Z. Zhou, J. Ma, S. Wen, Multi-label classification with label graph superimposing, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 12265–12272.
- [4] J. Ye, J. He, X. Peng, W. Wu, Y. Qiao, Attention-driven dynamic graph convolutional networks for multi-label image recognition, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 649–665.
- [5] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907, 2016.
- [6] B.-B. Gao, H.-Y. Zhou, Learning to discover multi-class attentional regions for multi-label image recognition, IEEE Trans. Image Process. 30 (2021) 5920–5932.
- [7] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, S. Wen, Cross-modality attention with semantic graph embedding for multi-label classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 12709–12716.
- [8] S. Narayan, A. Gupta, S. Khan, F.S. Khan, L. Shao, M. Shah, Discriminative region-based multi-label zero-shot learning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8731–8740.
- [9] X. Sun, P. Hu, K. Saenko, Dualcoop: fast adaptation to multi-label recognition with limited annotations, in: Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 30569–30582.
- [10] S. He, T. Guo, T. Dai, R. Qiao, X. Shu, B. Ren, S.-T. Xia, Open-vocabulary multi-label classification via multi-modal knowledge transfer, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 808–816.
- [11] P. Xia, D. Xu, L. Ju, M. Hu, J. Chen, Z. Ge, Lmpt: Prompt tuning with class-specific embedding loss for long-tailed multi-label visual recognition, arXiv preprint arXiv:2305.04536, 2023.
- [12] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu, et al., Recognize anything: A strong image tagging model, arXiv preprint arXiv:2306.03514, 2023.
- [13] X. Huang, Y.-J. Huang, Y. Zhang, W. Tian, R. Feng, Y. Zhang, Y. Xie, Y. Li, L. Zhang, Open-set image tagging with multi-grained text supervision, arXiv e-prints arXiv:2310.15200, 2023.
- [14] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.
- [15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: common objects in context, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 740–755.
- [16] T. Ridnik, G. Sharir, A. Ben-Cohen, E. Ben-Baruch, A. Noy, Ml-decoder: scalable and versatile classification head, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 32–41.

- [17] S. Liu, L. Zhang, X. Yang, H. Su, J. Zhu, Query2label: A simple transformer way to multi-label classification, arXiv preprint arXiv:2107.10834, 2021.
- [18] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, L. Zelnik-Manor, Asymmetric loss for multi-label classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 82–91.
- [19] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
- [20] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, T. Duerig, Scaling up visual and vision-language representation learning with noisy text supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 4904–4916.
- [21] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, H. Li, Tip-adapter: training-free adaption of CLIP for few-shot classification, in: Proceedings of the European Conference on Computer Vision, 2022.
- [22] X. Zhu, R. Zhang, B. He, A. Zhou, D. Wang, B. Zhao, P. Gao, Not all features matter: enhancing few-shot CLIP with adaptive prior refinement, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.
- [23] Z. Zheng, M. Ma, K. Wang, Z. Qin, X. Yue, Y. You, Preventing zero-shot transfer degradation in continual learning of vision-language models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.
- [24] X. Liu, X. Cao, H. Lu, J.-W. Xiao, A.D. Bagdanov, M.-M. Cheng, Class incremental learning with pre-trained vision-language models, arXiv preprint arXiv:2310.20348, 2023.
- [25] X. Gu, T.-Y. Lin, W. Kuo, Y. Cui, Open-vocabulary object detection via vision and language knowledge distillation, in: International Conference on Learning Representations, 2022.
- [26] S. Wu, W. Zhang, S. Jin, W. Liu, C.C. Loy, Aligning bag of regions for open-vocabulary object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15254–15264.
- [27] M. Xu, Z. Zhang, F. Wei, H. Hu, X. Bai, Side adapter network for open-vocabulary semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.
- [28] Z. Zhou, Y. Lei, B. Zhang, L. Liu, Y. Liu, Zegclip: towards adapting CLIP for zero-shot semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023.
- [29] S. Xu, Y. Li, J. Hsiao, C. Ho, Z. Qi, A dual modality approach for (zero-shot) multi-label classification, arXiv preprint arXiv:2208.09562, 2022.
- [30] X. Huang, Y. Zhang, J. Ma, W. Tian, R. Feng, Y. Zhang, Y. Li, Y. Guo, L. Zhang, Tag2text: Guiding vision-language model via image tagging, arXiv preprint arXiv:2303.05657, 2023.
- [31] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971, 2023.
- [32] R. Tao, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T.B. Hashimoto, Stanford alpaca: an instruction-following Llama model, 2023.
- [33] E. Almazrouei, H. Alabdulhadi, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Alhammedi, M. Daniele, D. Heslow, J. Launay, Q. Malartic, et al., The Falcon Series of Language Models: Towards Open Frontier Models, Hugging Face Repository, 2023.
- [34] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, J. Tang, Glm: General language model pretraining with autoregressive blank infilling, arXiv preprint arXiv:2103.10360, 2021.
- [35] J. Li, D. Li, S. Savarese, S. Hoi, Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, arXiv preprint arXiv:2301.12597, 2023.
- [36] W. Dai, J. Li, D. Li, A.M.H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, S. Hoi, Instructblip: Towards general-purpose vision-language models with instruction tuning, arXiv 2023, arXiv preprint arXiv:2305.06500, 2023.
- [37] H. Liu, C. Li, Y. Li, Y.J. Lee, Improved baselines with visual instruction tuning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 26296–26306.
- [38] D. Zhu, J. Chen, X. Shen, X. Li, M. Elhoseiny, Minigpt-4: Enhancing vision-language understanding with advanced large language models, arXiv preprint arXiv:2304.10592, 2023.
- [39] D. Zhu, J. Chen, K. Haydarov, X. Shen, W. Zhang, M. Elhoseiny, Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions, arXiv preprint arXiv:2303.06594, 2023.
- [40] A. López-Cifuentes, M. Escudero-Vinolo, J. Bescós, Á. García-Martín, Semantic-aware scene recognition, Pattern Recognit. 102 (2020) 107256.
- [41] A. Aminimehr, A. Molaei, E. Cambria, Entri: Ensemble learning with tri-level representations for explainable scene recognition, arXiv preprint arXiv:2307.12442, 2023.
- [42] R. Pereira, T. Barros, L. Garrote, A. Lopes, U.J. Nunes, A deep learning-based global and segmentation-based semantic feature fusion approach for indoor scene classification, Pattern Recognit. Lett. 179 (2024) 24–30.
- [43] D. Zeng, M. Liao, M. Tavakolian, Y. Guo, B. Zhou, D. Hu, M. Pietikäinen, L. Liu, Deep learning for scene classification: A survey, arXiv preprint arXiv:2101.10531, 2021.
- [44] Z. Yue, H. Zhang, Q. Sun, X.-S. Hua, Interventional few-shot learning, Adv. Neural Inf. Process. Syst. 33 (2020) 2734–2746.
- [45] R. Cadene, C. Dancette, M. Cord, D. Parikh, et al., Rubi: reducing unimodal biases for visual question answering, Adv. Neural Inf. Process. Syst. 32 (2019).
- [46] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 633–641.
- [47] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, A. Yuille, The role of context for object detection and semantic segmentation in the wild, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2014, pp. 891–898.
- [48] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2010) 303–338.
- [49] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
- [50] Y. Gong, Y. Jia, T. Leung, A. Toshev, S. Ioffe, Deep convolutional ranking for multilabel image annotation, arXiv preprint arXiv:1312.4894, 2013.
- [51] H. Wu, J. Mao, Y. Zhang, Y. Jiang, L. Li, W. Sun, W.-Y. Ma, Unified visual-semantic embeddings: bridging vision and language with structured meaning representations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6609–6618.
- [52] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, "O'Reilly Media, Inc.", 2009.
- [53] Z. Wu, M. Palmer, Verb semantics and lexical selection, in: 32nd Annual Meeting of the Association for Computational Linguistics, 1994, pp. 133–138.
- [54] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al., Judging llm-as-a-judge with mt-bench and chatbot arena, arXiv preprint arXiv:2306.05685, 2023.
- [55] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288, 2023.
- [56] Z. Bai, P. Wang, T. Xiao, T. He, Z. Han, Z. Zhang, M.Z. Shou, Hallucination of multimodal large language models: A survey, arXiv preprint arXiv:2404.18930, 2024.
- [57] S. Leng, H. Zhang, G. Chen, X. Li, S. Lu, C. Miao, L. Bing, Mitigating object hallucinations in large vision-language models through visual contrastive decoding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 13872–13882.
- [58] Q. Huang, X. Dong, P. Zhang, B. Wang, C. He, J. Wang, D. Lin, W. Zhang, N. Yu, Opera: alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 13418–13427.
- [59] Z. Wang, Y. Wang, X. Liu, L. Ding, M. Zhang, J. Liu, M. Zhang, Agentdropout: Dynamic agent elimination for token-efficient and high-performance llm-based multi-agent collaboration, arXiv preprint arXiv:2503.18891, 2025.
- [60] L. Chen, H. Zhao, T. Liu, S. Bai, J. Lin, C. Zhou, B. Chang, An image is worth 1/2 tokens after layer 2: plug-and-play inference acceleration for large vision-language models, in: European Conference on Computer Vision, Springer, 2024, pp. 19–35.
- [61] Y. Shang, M. Cai, B. Xu, Y.J. Lee, Y. Yan, Llava-pruner: adaptive token reduction for efficient large multimodal models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025, pp. 22857–22867.
- [62] C. Tian, Y. Wang, X. Liu, Z. Wang, L. Ding, M. Zhang, Agentinit: initializing llm-based multi-agent systems via diversity and expertise orchestration for effective and efficient collaboration, in: Findings of the Association for Computational Linguistics: EMNLP 2025, 2025, pp. 11870–11902.

Author biography



Mingyu Fu received the B.E. degree from Northwestern Polytechnical University, Xi'an, China, in 2021, where he is currently pursuing the Ph.D. degree with the School of Computer Science. His research interests include computer vision and pattern recognition.



Wei Suo received the B.E. degree from the North University of China, Taiyuan, China, in 2017. He received the M.S. degree and the Ph.D. degree from Northwestern Polytechnical University, Xi'an, China, in 2020 and 2025, respectively. He is currently an Associate Professor with the School of Computer Science, Northwestern Polytechnical University. His research interests include computer vision, vision and language, and pattern recognition.



Yukun Li received the B.E. degree from Lanzhou University, Lanzhou, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, continual learning and pattern recognition.



Lingyan Ran received his B.E. degree and Ph.D. degree from Northwestern Polytechnical University, Xi'an, China, in 2011 and 2018. Earlier, he was a visiting scholar at Stevens Institute of Technology (SIT), Hoboken, NJ, from 2013 to 2015. His research interests include image classification and semantic segmentation.



Peng Wang received the B.E. degree in electrical engineering and automation and the Ph.D. degree in control science and engineering from Beihang University, Beijing, China, in 2004 and 2011, respectively. He is currently a Professor with the School of Computer Science, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision, machine learning, and artificial intelligence.