NJUST_KMG at SemEval 2026 Task 10 PsyCoMark—Subtask 2: Conspiracy Detection

Yuhan Zheng, Yang Yang
Nanjing University of Science and Technology
Nanjing, China
zhengyuhan@njust.edu.cn, yyang@njust.edu.cn

Abstract

This paper describes our system designed for SemEval-2026 Task 10: PsyCoMark—Subtask 2: Conspiracy Detection. We proposed a two-stage approach that leverages large-scale pre-trained models and a fine-tuned smaller model to detect conspiracy theories in text. In the first stage, we utilize a large model to test all the test samples and filter out those that are clearly unrelated to conspiracy theories. For the remaining samples, we apply a retrieval-enhanced custom prompt strategy combined with the Roberta-Large model in the second stage. This allows us to fine-tune the model with weighted predictions based on relevant retrieved information, enhancing detection accuracy. Our system achieved first place on the leaderboard, with an impressive F1 Score of 0.8874. We also present a brief analysis of the effectiveness of the methods used, including the advantages and limitations of large model-based filtering and retrieval-augmented fine-tuning.

1 Introduction

The PsyCoMark task (Ghosh et al., 2026) addresses the gap in benchmarks for "automatic detection of conspiracy content in everyday online language" by combining "conspiracy detection" (classification) with "psycholinguistic reasoning tasks." The dataset is available on Zenodo and open for community collaboration via GitHub.

The sub-task is divided into two types: determining if a "submission statement" expresses a conspiracy-related claim (Yes/No), or categorizing it as "Can't tell." The evaluation metric is the weighted F1 score, which averages the weighted form to assess classification performance across tasks.

The dataset is constructed using Reddit submission statements, helping train the task to constrain the target corpus, as conspiracy theories appear in various forms. This training set differs from real platform data and requires adaptation through examples like Pushshift and Project Arctic Shift.

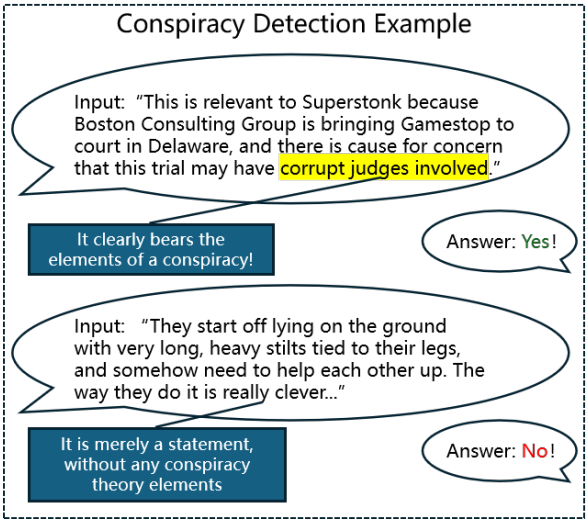

Figure 1: Example of conspiracy theory detection. The first input clearly includes conspiracy elements, whereas the second input is simply a statement without any conspiracy-related content.

Additionally, the task integrates psycholinguistic "conspiracy detection" spans (e.g., Actor, Action, Evidence, Condition, Term), illustrating the knowledge structure and reasoning process in the task execution.

We adopt multimodal learning and retrieval enhancement techniques, which have been widely applied across domains. Zhang et al. (Zhanga et al.) proposed a multimodal semantic decoupled prompt for zero-shot referring expression comprehension. Li et al. (Li et al., 2025) explored noise self-correction in cross-modal retrieval, improving robustness. Yang et al. (Yang et al., 2025) introduced a probe-and-rebalance approach for multimodal learning, while Yang et al. (Yang et al., 2024) improved multimodal classification by dynamically learning modality gaps. Yang et al. (Yang et al., 2023) discussed the benefits of out-of-distribution data in open-set active learning,

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 932–937

and Jiang et al. (Jiang et al., 2025) rethought multimodal learning to address classification ability disproportion, crucial for balancing model outputs in our approach.

2 Related Work

In the past five years, research on conspiracy theory detection has progressed along key lines: (1) supervised text classification (Transformer fine-tuning), (2) cross-platform/cross-lingual shared tasks, (3) large model prompting and weak supervision, and (4) retrieval augmentation and cascading inference (Pustet et al., 2024). Besides the SemEval task, academic efforts have organized evaluations for various platforms and languages. For example, the ACTI task in EVALITA 2023 advanced evaluation on non-news, non-encyclopedic dialog corpora, using Transformer-based models and enhancement techniques like data augmentation and sentence vectors (Russo et al., 2023). Studies highlight that supervised models can be biased by keywords on specific platforms, suggesting large models to improve robustness beyond keyword-based detection (Pustet et al., 2024).

Recent work on large language models (LLMs) has explored techniques like zero-shot/few-shot prompting, prompt-based learning, and instruction fine-tuning. For instance, GPT-3-based models showed the feasibility of using large models for various tasks with minimal training, providing strong baselines for competitive scenarios (Brown et al., 2020). In conspiracy theory detection, studies are focusing on task-specific instruction data and emphasizing emotion, narrative, and psychological factors in detection (Liu et al., 2024).

Additionally, retrieval augmentation with LLMs, like REALM and RAG, improves factuality and updatability by combining knowledge bases with generative models (Guu et al., 2020). Retrieval-augmented prompting (RAP) adapts static templates into instance-specific contexts, proving useful for tasks with high ambiguity and evidence dependency. Lastly, cascading inference strategies, as seen in FrugalGPT, balance performance and cost by using simpler models for easier cases and reserving more complex models for harder cases (Chen et al., 2023).

3 System Overview

As shown in Figure 2, our system consists of four stages. Stage 1 enhances data relevance through retrieval and reranking. Stage 2 applies pre-trained large models (PLMs) like BERT, RoBERTa, and DeBERTa, using techniques such as class weight adjustment and data augmentation to address imbalance and overfitting. Advanced models like Qwen2.5 and ChatGPT5.2 are also incorporated to boost performance. In Stage 3, a designed prompt interacts with large language models (LLMs) for predictions. Finally, Stage 4 combines model outputs using ensemble learning for better robustness and generalization in conspiracy detection.

3.1 Model Architecture

Pre-trained Language Models (PLMs): We fine-tuned two Transformer-based models, RoBERTa and DeBERTa, for our sequence classification task. RoBERTa improves performance with better training methods and larger datasets, while DeBERTa enhances contextual understanding with its disentangled attention mechanism and mask decoder. Both PLMs are used as encoders connected to a classification layer for final predictions.

Large Language Models (LLMs): We also fine-tuned two advanced LLMs, Qwen2.5 and ChatGPT5.2. After pre-processing with their respective tokenizers, we fine-tuned the models and applied them to the test set. Unlike PLMs, LLMs generate predictions directly through a generative approach. Model performance is evaluated by comparing predictions to true labels.

This section further details the two-stage pipeline in Section 3: Stage 1 uses LLMs for global pre-filtering, while Stage 2 applies retrieval-enhanced prompts and RoBERTa-Large weighted fusion for hard examples, including threshold selection, probability calibration, and implementation details. For related methods, see studies on cascade inference, retrieval enhancement, and calibration.

3.2 LLM Global Pre-filtering

Motivation (Why filtering). In PsyCoMark subtask 2, the official binary evaluation uses {`No`, `Yes`} and the weighted F1 score as the core metric. Thus, Stage 1 is designed as a *high-precision negative class filter*, where samples clearly not containing conspiracy elements are classified as `No`, focusing resources on ambiguous samples needing evidence. This approach aligns with the cost-performance trade-off in LLM cascade/triage.

Notation and Output Constraints. Let the input text be x and the label space be $y \in \{\texttt{No}, \texttt{Yes}\}$.

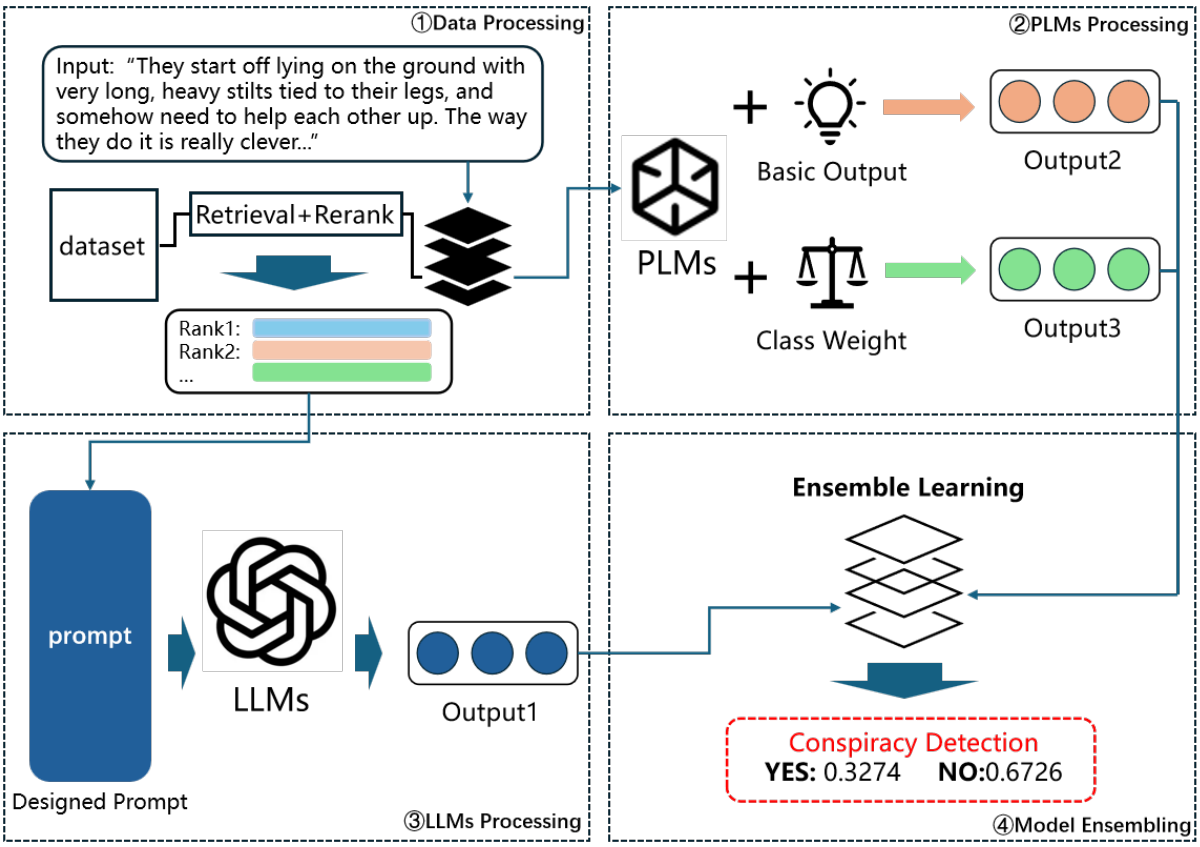

Figure 2: the architecture of our proposed system. In the first stage, data processing involves retrieval and reranking to enhance the relevance of the input dataset. The second stage applies pre-trained large models (PLMs) such as RoBERTa, fine-tuned with class weight adjustment to address data imbalance. The third stage utilizes large language models (LLMs), where a designed prompt is used to generate predictions. Finally, in Stage 4, the system aggregates predictions from multiple models using ensemble learning, ensuring improved robustness and accuracy for the conspiracy detection task.

The LLM in Stage 1 generates a structured output $\mathrm{Resp}^{(1)}(x)$ under the prompt template $\mathcal{P}_{\text{filter}}$. To reduce ambiguity, the output is constrained to a fixed format (e.g., JSON with `label` and `p_yes` fields); specific model types and sampling parameters **are not specified**.

Probability Extraction and Routing Branch. The LLM output is mapped to the positive class probability:

$$p^{(1)}_{\text{LLM}}(x) = g\Big(\mathrm{Resp}^{(1)}(x)\Big) \in [0,1], \quad (1)$$

where $g(\cdot)$ represents parsing and necessary post-processing. The routing rule for Stage 1 is:

$$\text{route}(x) = \begin{cases} \texttt{predict No}, & \text{if } p^{(1)}_{\text{LLM}}(x) < \tau_{\text{filter}}, \\ \texttt{Stage2}, & \text{otherwise.} \end{cases} \quad (2)$$

where τ_{filter} **is not specified**. A smaller τ_{filter} makes Stage 1 more conservative, reducing false negatives, while a larger threshold increases filtering aggressiveness but raises the risk of false negatives.

Threshold Selection and (Optional) Calibration. We perform a grid search or ranking-based threshold scan for τ_{filter} on the development set, using weighted F1 after Stage 2 for optimization. If $p^{(1)}_{\text{LLM}}(x)$ is from the LLM's "self-reported probability," its reliability may vary with prompt and sampling fluctuations. Optionally, temperature scaling or logistic regression calibration can be applied on the development set to improve interpretability.

Engineering Implementation Key Points (Efficiency and Robustness). To improve efficiency, we use the following strategies:

- **Caching**: Cache $\mathrm{Resp}^{(1)}(x)$ and $p^{(1)}_{\text{LLM}}(x)$ at the sample level to avoid redundant LLM calls.
- **Batch Processing and Length Limiting**: Use batch processing for LLMs that support it, and control input length through truncation/summary.
- **Parsing Fault Tolerance**: If parsing fails, de-

grade the sample to "enter Stage 2" to reduce false negatives.

3.3 Retrieval-Enhanced Prompting and RoBERTa-Large Weighted Fusion

Motivation. For the "hard examples" retained by Stage 1, relying solely on superficial word triggers is prone to topic drift and semantic ambiguity; retrieval enhancement can inject relevant background and similar examples for each sample, improving prompt alignment and consistency. Meanwhile, fine-tuned RoBERTa-Large as a discriminative encoder typically provides more stable probability signals. Therefore, we combine the LLM signal from retrieval-enhanced prompting with the RoBERTa-Large probability through weighted fusion to balance semantic coverage and discriminative stability.

Stage 2.1: Retrieval and Reranking. Let the retrieval corpus be $\mathcal{C}$ (sources and scale **are not specified**). Given input x, construct a query $q(x)$ (which may be the original text, key phrases, or a compressed query; specific strategy **not specified**), and perform two-stage retrieval:

$$\mathcal{D}_k(x) = \mathrm{Retrieve}(q(x), \mathcal{C}, k)\,, \tag{3}$$

$$\mathcal{E}_m(x) = \mathrm{SelectTop}(\mathrm{Rerank}(x, \mathcal{D}_k(x)),\, m)\,, \tag{4}$$

where k is the number of recall candidates, and m is the number of final evidence entries, both **not specified**; the retrieval type (sparse/dense/mixed) and reranker type (e.g., cross-encoder) **are not specified**. The two-stage "first recall, then rerank" motivation is to obtain higher relevance evidence at a lower cost.

Stage 2.2: Retrieval-Enhanced Prompt Construction and Evidence Injection. We inject $\mathcal{E}_m(x)$ into the prompt template $\mathcal{P}_{\mathrm{RAG}}$ in the form of "citation blocks/point lists," and explicitly define in the prompt: (i) task definition and label semantics; (ii) evidence usage boundaries (only using evidence as auxiliary, no fabrication of non-existent information); (iii) output format constraints (suggesting output of `p_yes` and a brief justification). The LLM output is mapped as:

$$p^{(2)}_{\mathrm{LLM}}(x) = h(\mathrm{LLM}(x, \mathcal{E}_m(x); \mathcal{P}_{\mathrm{RAG}})) \in [0,1] \tag{5}$$

where $h(\cdot)$ represents the parsing function. The LLM model, prompt details, context length budget, and whether to use self-consistency **are not specified**.

Stage 2.3: RoBERTa-Large Fine-Tuning and Probability Calibration. The RoBERTa-Large sequence classifier outputs binary logits $z(x) = [z_{\texttt{No}}(x), z_{\texttt{Yes}}(x)]$, corresponding to the probability:

$$\begin{aligned} p_{\mathrm{RoB}}(y|x) &= \mathrm{softmax}(z(x))_y, \\ p^{(2)}_{\mathrm{RoB}}(x) &= p_{\mathrm{RoB}}(\texttt{Yes}|x). \end{aligned} \tag{6}$$

During training, we use weighted cross-entropy (or equivalent binary log loss):

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \alpha_{y_i} \log p_\theta(y_i|x_i), \tag{7}$$

where α_y is the class weight (determined by class frequency or validation set tuning; value **not specified**). To improve probability usability, temperature scaling calibration can be applied on the development set:

$$\tilde{p}_{\mathrm{RoB}}(y|x) = \mathrm{softmax}\left(\frac{z(x)}{T}\right)_y, \quad T > 0, \tag{8}$$

where T is obtained by minimizing the negative log-likelihood on the development set; the final value of T **is not specified**. Training hyperparameters such as learning rate, epoch, batch size, and maximum length **are not specified**.

Stage 2.4: Weighted Fusion and Final Decision. For the same sample x, we combine the two probabilities, possibly calibrated, as follows:

$$\begin{aligned} p_{\mathrm{final}}(x) = w_{\mathrm{RoB}} \cdot \tilde{p}^{(2)}_{\mathrm{RoB}}(x) + w_{\mathrm{LLM}} \cdot \tilde{p}^{(2)}_{\mathrm{LLM}}(x), \\ w_{\mathrm{RoB}} + w_{\mathrm{LLM}} = 1, \quad w_{\mathrm{RoB}}, w_{\mathrm{LLM}} \geq 0. \end{aligned} \tag{9}$$

where the fusion weights w_{RoB} and w_{LLM} **are not specified** (suggested to be determined via grid search or simple linear regression fitting on the development set). The final binary classification decision is:

$$\hat{y}(x) = \begin{cases} \texttt{Yes}, & p_{\mathrm{final}}(x) \geq \tau_{\mathrm{final}}, \\ \texttt{No}, & \text{otherwise}, \end{cases} \tag{10}$$

The threshold τ_{final} is selected on the development set to maximize the official weighted F1 score, and τ_{final} **is not specified**.

Engineering Implementation Key Points (Retrieval and Inference Efficiency).

- **Pre-build Vector/Inverted Indexes**: Offline construction of retrieval indexes; chunking and deduplication of evidence fragments (chunking granularity **is not specified**).

- **Two-level Caching**: Cache $\mathcal{D}_k(x)$ and $\mathcal{E}_m(x)$, as well as $p^{(2)}_{\mathrm{LLM}}(x)$, to reduce redundant retrieval and LLM calls; batch RoBERTa inference.

- **Length Budget Control**: Truncate and compress evidence (e.g., retaining the first n words/sentences for each piece of evidence; n **is not specified**), to avoid excessive length in the prompt that leads to LLM degradation.

4 Experimental Setup and Results

4.1 Dataset Description

For our experiments, we use the PsyCoMark dataset, which is specifically designed for conspiracy theory detection in textual data. The dataset consists of a diverse set of textual samples sourced from multiple platforms, including social media posts, articles, and forum discussions. Each sample in the dataset is labeled as either `Yes` (conspiracy theory) or `No` (non-conspiracy theory). The dataset is split into training, validation, and test subsets, with the training set consisting of approximately 80% of the total data, while the validation and test sets make up 10% each.

4.2 Metrics

We use the F1 score as the main evaluation metric, which is the harmonic mean of precision and recall. In multi-class classification, the F1 score is calculated for each class, and then averaged to obtain the overall score. This approach treats each class equally, regardless of class distribution, providing a balanced measure of model performance.

5 Results

Table 2 presents the effectiveness of our method, evaluated using the Macro F1 score. Starting with RoBERTa as the baseline, we achieved a score of 0.7451. Incorporating data processing improved the score to 0.7812, highlighting the impact of optimizing the input data. Adding class weight further boosted the score to 0.8191, addressing class imbalance. The inclusion of LLMs with a designed prompt led to a significant improvement, with the score rising to 0.8442, demonstrating the benefit of leveraging large models for more accurate predictions. Finally, the combination of these methods with ensemble learning achieved the highest score of 0.8874, showing the effectiveness of aggregating multiple models to enhance robustness and generalization.

Component	Setting
Stage-1 LLM	GPT-5.2,Qwen2.5-7B-Instruct
Temperature / Top-p	1.0 / 1.0
Max generation tokens	128
Filtering threshold τ_{filter}	0.3
Retrieval corpus	official training set
Corpus size	official training set passages
Retriever	BM25 + dense retrieval
Embedding model	bge-large-en-v1.5
Recall candidates k	20
Reranker	bge-reranker-large
Final evidence entries m	3
RoBERTa backbone	roberta-large
Learning rate	2×10^{-5}
Epochs	10
Weight decay	0.01
Batch size	10
Fusion weights	$w_{\mathrm{RoB}} = 0.35$, $w_{\mathrm{LLM}} = 0.65$
Final threshold τ_{final}	0.73

Table 1: Implementation details of our submitted system.

These results highlight the substantial impact of each component on the overall performance of the system, with ensemble learning being the most influential factor. By combining multiple models, we are able to capture diverse patterns in the data, significantly improving the model's ability to detect conspiracy-related content.

Settings	**Macro F1**
RoBERTa	0.7451
+ data processing	0.7812
+ class weight	0.8191
+ LLMs designed prompt	0.8442
+ ensemble learning	0.8874

Table 2: Effectiveness of the proposed method.

Our method performed exceptionally well in the competition, securing the top position with a Macro F1 score of 0.89. This result demonstrates the effectiveness of our approach, which benefited from techniques such as threshold search, class weighting, data augmentation, and ensemble learning. The table below shows the final rankings for the top 5 teams in the leaderboard.

6 Conclusion

We proposed an effective approach for conspiracy theory detection, combining LLMs with techniques like retrieval-enhanced prompting, class weighting, and ensemble learning. Our method achieved the top Macro F1 score of 0.89 on the PsyCoMark task.

Rank	Team	Macro F1
1	**NJUST_KMG**	**0.89**
2	mdok-style	0.78
3	dangphuduy	0.78
4	VARH-AI	0.78
5	UMUTeam	0.77

Table 3: Results of top 5 teams for the competition task on the test set.

Future work will focus on improving robustness, exploring other modalities, and refining the method for broader misinformation detection.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Debanjan Ghosh, Kai North, Ekaterina Kochmar, Mamoru Komachi, and Marcos Zampieri, editors. 2026. *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*. Association for Computational Linguistics, San Diego, United States.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

QingYuan Jiang, Longfei Huang, and Yang Yang. 2025. Rethinking multimodal learning from the perspective of mitigating classification ability disproportion. *arXiv preprint arXiv:2502.20120*.

Ruoxuan Li, Xiangyu Wu, and Yang Yang. 2025. Noise self-correction via relation propagation for robust cross-modal retrieval. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 4748–4757.

Zhiwei Liu, Boyang Liu, Paul Thompson, Kailai Yang, and Sophia Ananiadou. 2024. Conspemollm: Conspiracy theory detection using an emotion-based large language model. *arXiv preprint arXiv:2403.06765*.

Milena Pustet, Elisabeth Steffen, and Helena Mihaljević. 2024. Detection of conspiracy theories beyond keyword bias in german-language telegram using large language models. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 13–27.

Giuseppe Russo, Niklas Stoehr, and Manoel Horta Ribeiro. 2023. Acti at evalita 2023: Overview of the conspiracy theory identification task. *arXiv preprint arXiv:2307.06954*.

Yang Yang, Fengqiang Wan, Qing-Yuan Jiang, and Yi Xu. 2024. Facilitating multimodal classification via dynamically learning modality gap. *Advances in Neural Information Processing Systems*, 37:62108–62122.

Yang Yang, Xixian Wu, and Qing-Yuan Jiang. 2025. Towards equilibrium: An instantaneous probe-and-rebalance multimodal learning approach. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 3552–3560.

Yang Yang, Yuxuan Zhang, Xin Song, and Yi Xu. 2023. Not all out-of-distribution data are harmful to open-set active learning. *Advances in Neural Information Processing Systems*, 36:13802–13818.

Yuxuan Zhanga, Longfei Huanga, and Yang Yanga. Multimodal semantic decoupled prompt for zero-shot referring expression comprehension.