TLSA: LLM-Guided Text-Label Space Alignment with Contrastive Learning for Generalized Category Discovery

Wenxi Xu[1,2,3], **Chuan Qin**[1,3*], **Xi Chen**[4,1]
Chuyu Fang[5], **Yuanchun Zhou**[1,2,3*], **Hengshu Zhu**[1,3]
[1]Computer Network Information Center, CAS
[2]School of Advanced Interdisciplinary Sciences, UCAS
[3]University of Chinese Academy of Sciences
[4]University of Science and Technology of China
[5]Baidu Inc.
{wxxu, zyc}@cnic.cn, chenxi0401@mail.ustc.edu.cn
{chuanqin0426, fangchuyu2022, zhuhengshu}@gmail.com

Abstract

Generalized Category Discovery (GCD) aims to classify data from partially labeled datasets by jointly recognizing known categories and discovering novel ones. Despite recent advances, existing methods still suffer from weak text–label alignment, inconsistent objectives across known and novel categories, and poor discrimination of semantically similar clusters. To mitigate these issues, we propose TLSA, a unified framework that enforces contrastive alignment between text and label representations within a shared semantic space. Specifically, we first design a label-semantic aware dual-encoder equipped with a symmetric contrastive objective to achieve text-label alignment. Then, we leverage LLM-based label induction to generate explicit and semantically meaningful names for previously unseen categories, followed by a graph-based refinement strategy that disambiguates semantically overlapping clusters through forced renaming. Finally, a confidence-aware sampling strategy ensures balanced learning across both easy and hard instances. Extensive experiments on four benchmark datasets show that TLSA consistently outperforms state-of-the-art GCD methods. The code is available at https://github.com/Wenxi-Xu/TLSA.

1 Introduction

Deep learning has demonstrated remarkable effectiveness in text classification when trained on large-scale annotated corpora. Despite this success, most prior work has been developed under the closed-set assumption, where all possible categories are known in advance. Such an assumption is rarely satisfied in real-world applications, which often face challenges of limited supervision, domain shifts, and evolving label spaces. To address these issues, a growing body of research has explored semi-supervised and self-supervised approaches as alternatives to supervised learning.

Recently, the task of generalized category discovery (GCD) has emerged as a promising paradigm for learning from partially labeled data (Lin et al., 2020; Zhang et al., 2021). In this setting, a model is trained on a limited set of labeled samples from known classes, alongside a large pool of unlabeled data that encompasses both known and previously unseen categories. The model needs to distinctly separate known classes while discovering and isolating unknown ones. This setting is particularly relevant in real-world scenarios where annotation budgets are limited and new categories continuously emerge over time.

Contemporary methods for GCD predominantly employ a two-stage paradigm. The initial stage involves training a feature encoder on labeled data. Subsequently, unlabeled samples are clustered and iteratively refined using pseudo labels, often augmented with semantic guidance from large language models (An et al., 2024a; Zou et al., 2025; Liang et al., 2024). However, three core limitations remain: (1) Treating labels as mere identifiers (one-hot targets) rather than leveraging their true semantics weakens text–label alignment. (2) Optimizing known and novel categories with different objectives (Zhang et al., 2021; An et al., 2023b), such as a classification loss for labeled samples and contrastive clustering losses for unlabeled ones, introduces an inherent bias toward known classes that becomes more severe when novel classes dominate. (3) For fine-grained label spaces, semantically close categories tend to occupy overlapping representation regions, making clusters less separable and decision boundaries ambiguous.

*Corresponding Authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 19030–19046
July 2-7, 2026

To address these challenges, we propose TLSA, a unified framework that performs **T**ext–**L**abel **S**pace **A**lignment under LLM guidance with contrastive learning. Specifically, we first employ a dual-encoder architecture to embed text and label into a shared semantic space, where a symmetric contrastive objective aligns the two modalities. Building upon these aligned representations, we introduce LLM-driven label induction to automatically derive explicit and interpretable labels for new categories. We further apply graph-based refinement to disambiguate semantically overlapping clusters and employ confidence-aware sampling to ensure balanced learning across easy and hard instances. Extensive experiments on four benchmark datasets demonstrate that TLSA consistently outperforms state-of-the-art methods.

2 Related Work

Generalized Category Discovery Recently, open-world learning has attracted growing attention for its relevance to realistic settings, such as labor-market analytics (Qin et al., 2025b; Chen et al., 2024). In text classification, a representative formulation is open-set text classification (OSTC), which requires models to distinguish in-distribution classes from unknown inputs (Prakhya et al., 2017; Chen et al., 2026). Generalized category discovery (GCD) further extends this setting by uncovering the latent structure among unknown instances while preserving reliable recognition of known classes (Vaze et al., 2022). Existing GCD methods typically adopt a two-stage pipeline: they first train on the labeled set and then cluster or pseudo-label the unlabeled data, with refinements such as alignment-based clustering (Zhang et al., 2021), robust pseudo-label training (An et al., 2023a), and geometry- or structure-aware constraints (Tang et al., 2024; Zhang et al., 2024). More recent studies emphasize knowledge transfer: decoupled prototypical learning improves separability for novel classes while preserving known-class discrimination (An et al., 2023b), and prototype/instance alignment methods further reduce bias between domains (Shi et al., 2024; An et al., 2024b). In parallel, calibration-based methods mitigate the tendency to overfit known classes by adjusting prediction logits, thereby facilitating novel-class discovery (An et al., 2025).

LLMs for GCD LLMs have been widely applied across a range of natural language processing domains (Qin et al., 2025c; Jiang et al., 2024; Tong et al., 2025; Qin et al., 2025a; Huang et al., 2026; Song et al., 2026). Recent studies integrate LLMs into GCD via active learning. ALUP queries uncertain samples and propagates soft labels (Liang et al., 2024); LOOP leverages LLM feedback to refine neighborhoods and name clusters for evaluation (An et al., 2024a); and GLEAN combines similarity judgments, cluster descriptions, and sample-to-cluster assignments with quality control (Zou et al., 2025). Although LOOP and GLEAN generate cluster names, such names are not incorporated into representation learning, serving only for evaluation or querying the LLM. In contrast, our approach explicitly models labels as text and enforces contrastive alignment between text and label representations within a shared semantic space, enhancing semantic coherence and substantially improving performance in GCD.

3 Preliminaries

Problem Definition Given a labeled dataset $\mathcal{D}^l = \{(x, y) \mid y \in \mathcal{Y}^k\}$, models trained under the closed-set assumption are effective in recognizing predefined known categories $\mathcal{Y}^k$. However, in realistic open-world settings, models are inevitably exposed to unlabeled data $\mathcal{D}^u = \{x \mid y \in \mathcal{Y}^k \cup \mathcal{Y}^n\}$, which contains instances from both known categories $\mathcal{Y}^k$ and novel categories $\mathcal{Y}^n$ with $\mathcal{Y}^k \cap \mathcal{Y}^n = \emptyset$. This mismatch often leads to severe failures in identifying novel categories. The task of GCD therefore aims to simultaneously recognize known categories while discovering novel categories by leveraging both $\mathcal{D}^l$ and $\mathcal{D}^u$. Following prior work (An et al., 2023b; Shi et al., 2024; An et al., 2025), we assume the number of novel categories $|\mathcal{Y}^n|$ is known during training. Evaluation is conducted on a test set $\mathcal{D}^t = \{(x, y) \mid y \in \mathcal{Y}^k \cup \mathcal{Y}^n\}$ inductively.

Conventional GCD approaches typically fine-tune a PLM with a classification head on $\mathcal{D}^l$, and then apply clustering on the learned instance representations from $\mathcal{D}^u$ to assign pseudo one-hot labels. These pseudo-labeled samples are then used to augment the classifier, which is trained iteratively. However, since the classification head is trained only on $\mathcal{Y}^k$, the learned representations are inherently biased toward known categories. Moreover, pseudo labels for both known and novel categories remain semantic-free one-hot identifiers that fail to convey category-level meaning, limiting

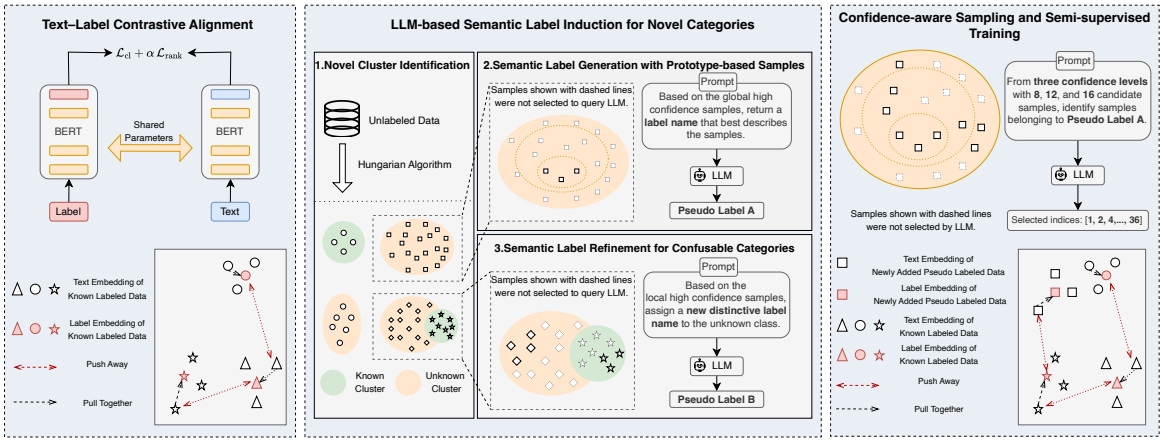

Figure 1: Overall architecture of TLSA. *Text–Label Contrastive Alignment* pre-trains a dual encoder on labeled data via a symmetric contrastive objective. *LLM-based Semantic Label Induction* identifies unknown clusters via clustering and the Hungarian algorithm, then leverages LLM induction to generate and refine pseudo labels. *Confidence-aware Sampling and Semi-supervised Training* perform confidence-stratified sampling and LLM verification for joint training with labeled data.

generalization to unseen classes.

Solution Overview To overcome the limitations of conventional one-hot based GCD approaches, we propose a unified framework TLSA. First, we introduce *Text–Label Contrastive Alignment* to jointly embed texts and labels into a shared semantic space, enabling classification through representation matching. Second, we employ *LLM-based Semantic Label Induction for Novel Categories* to generate and refine semantic labels for novel clusters, facilitating interpretable category discovery. Finally, we design a *Confidence-aware Sampling and Semi-supervised Training* strategy that filters samples with multi-level confidence and iteratively performs semi-supervised training. This unified framework enables the model to move beyond predefined categories, effectively discovering and learning novel classes.

4 Method

4.1 Text–Label Contrastive Alignment

A key limitation of conventional GCD models lies in their reliance on a classification head that maps instances to a fixed one-hot label space, which restricts the model's ability to generalize beyond predefined categories. To address this, we propose a *Label-Semantic Aware Dual-Encoder* architecture that directly aligns texts and labels within a shared semantic space. This design enables the model to bypass the constraints of a fixed classifier, making it naturally extensible to novel categories.

Dual Representations for Text and Label To enable the model to generalize beyond the one-hot label space, we design a dual-encoder architecture that jointly models semantic representations for both texts and labels. Specifically, given an input text x, we encode it with a BERT encoder $f_\theta(\cdot)$ followed by a projection MLP, obtaining the text embedding $h_x = \mathrm{MLP}_t(f_\theta(x))$. Similarly, for each label $y \in \mathcal{Y}^k$, we construct a textual description (e.g., its name or natural language definition), which is encoded by the same BERT with an independent projection head, yielding the label embedding $h_y = \mathrm{MLP}_l(f_\theta(y))$. During training, this dual representation enables semantic alignment between texts and labels through contrastive matching.

Symmetric Difficulty-aware Reweighted Contrastive Loss Given a mini-batch of text–label pairs $\{(x_i, y_i)\}_{i=1}^{B}$, the objective is to align each text embedding h_{x_i} with its corresponding label embedding h_{y_i} while pushing it away from non-matching labels. A standard contrastive objective computes the probability of the correct label under a softmax distribution:

$$\mathcal{L}_{\mathrm{cl}}(x, y) = -\log \frac{\exp(\mathrm{sim}(h_x, h_y)/\tau)}{\sum_{y' \in \mathcal{Y}^k} \exp(\mathrm{sim}(h_x, h_{y'})/\tau)}, \quad (1)$$

where τ is a temperature parameter and $\mathrm{sim}(\cdot, \cdot)$ denotes cosine similarity computed as:

$$\mathrm{sim}(x, y) = \frac{h_x^\top h_y}{\|h_x\| \|h_y\|}. \quad (2)$$

We introduce two key enhancements. First, the loss is made symmetric, enforcing alignment in

both directions to avoid representation collapse and to maintain balanced embedding spaces. Let $\mathbf{Z} \in \mathbb{R}^{B\times B}$ denote the cosine similarity matrix for text-label pairs within a mini-batch of size B, where $\mathbf{Z}_{ij} = \mathrm{sim}(h_{x_i}, h_{y_j})/\tau$. The symmetric contrastive loss is defined as:

$$\mathcal{L}_{\text{cl}}^{t\to l} = \frac{1}{B}\sum_{i=1}^{B}\Big[-\log\frac{\exp(\mathbf{Z}_{ii})}{\sum_{j=1}^{B}\exp(\mathbf{Z}_{ij})}\Big],$$
$$\mathcal{L}_{\text{cl}}^{l\to t} = \frac{1}{B}\sum_{j=1}^{B}\Big[-\log\frac{\exp(\mathbf{Z}_{jj})}{\sum_{i=1}^{B}\exp(\mathbf{Z}_{ij})}\Big],$$
$$\mathcal{L}_{\text{cl}} = \tfrac{1}{2}\big(\mathcal{L}_{\text{cl}}^{t\to l} + \mathcal{L}_{\text{cl}}^{l\to t}\big). \tag{3}$$

Second, we incorporate a difficulty-aware ranking loss. For $i \neq j$, a larger $\mathbf{Z}_{ij}$ means the text x_i is more similar to an incorrect label y_j, hence this negative is more confusing. Accordingly, we impose a margin-based penalty on the top-k negatives with the largest $\mathbf{Z}_{ij}$. For text-to-label alignment, for each i we select the top-k negatives from $\{\mathbf{Z}_{ij}\}_{j\neq i}$ and compute:

$$\mathcal{L}_{\text{rank}}^{t\to l} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{k}\sum_{j\in\mathcal{N}_i^{t\to l}}[\mathbf{Z}_{ij} - \mathbf{Z}_{ii} + m]_+, \tag{4}$$

where $m \geq 0$ is a margin and $[u]_+ = \max(0, u)$. Analogously, for label-to-text alignment we obtain $\mathcal{L}_{\text{rank}}^{l\to t}$ by applying the same procedure on $\mathbf{Z}^\top$. Finally, the difficulty-aware ranking loss is

$$\mathcal{L}_{\text{rank}} = \tfrac{1}{2}(\mathcal{L}_{\text{rank}}^{t\to l} + \mathcal{L}_{\text{rank}}^{l\to t}). \tag{5}$$

Combining the symmetric contrastive loss with the rank loss yields the final training objective:

$$\mathcal{L}_{train} = \mathcal{L}_{\text{cl}} + \alpha\,\mathcal{L}_{\text{rank}}, \tag{6}$$

where α is a weight hyperparameter controlling the contribution of hard-negative reweighting.

This design ensures that the dual-encoder not only learns robust text–label alignment in a symmetric manner, but also pays particular attention to fine-grained semantic distinctions. A warm-up phase is first conducted on the known labeled data.

4.2 LLM-based Semantic Label Induction for Novel Categories

While the dual-encoder enables semantic alignment between texts and labels, it alone does not provide explicit names for unseen categories. To ensure interpretability and facilitate the extension of the label space, we incorporate LLMs to induce descriptive labels for clusters that are not aligned with known categories. This process involves three stages: identifying novel clusters by aligning discovered clusters with known categories, generating semantic labels for novel clusters using representative samples, and refining ambiguous labels through a similarity-aware disambiguation step.

Novel Cluster Identification via Known-class Alignment Given the unlabeled dataset $\mathcal{D}^u$, we first encode each instance with the text encoder and obtain its semantic embedding. K-Means clustering is then applied to partition $\mathcal{D}^u$ into $|\mathcal{Y}^k| + |\mathcal{Y}^n|$ groups, corresponding to both known and novel categories. To distinguish them, we identify clusters corresponding to known categories by aligning their centroids with precomputed labeled known-class representatives. This alignment is formulated as a maximum-weight bipartite matching problem and solved via the Hungarian algorithm (Kuhn, 1955). Details are provided in Appendix B. Clusters matched to known centroids are labeled as *aligned-to-known*, while unmatched clusters are regarded as *novel candidates*.

Semantic Label Generation with Prototype-based Samples We first compute confidence scores for unlabeled samples. Given a sample x in cluster C_i, its confidence score is defined as:

$$\mathrm{conf}(x, C_i) = \frac{d(x, C_j) - d(x, C_i)}{d(C_i, C_j)}, \tag{7}$$

where C_j is the nearest alternative cluster, $d(\cdot,\cdot)$ denotes Euclidean distance, and $d(C_i, C_j)$ is the distance between cluster centroids.

For each candidate novel cluster in C_i, we select a small set of high-confidence representative samples, which are then used as input to prompt an LLM to generate a concise, human-readable label that summarizes the cluster's dominant semantics. Details of the prompt are provided in Appendix E.1.1. By operating directly in natural language space rather than relying on numeric identifiers, this procedure leverages the LLM's prior knowledge to produce informative and interpretable category names. Importantly, the LLM is instructed to avoid duplication with existing known labels, thereby ensuring that the induced labels extend rather than overlap with the predefined label space.

Semantic Label Refinement for Confusable Categories Although the initial LLM-generated labels provide semantic interpretability, clusters with high semantic similarity may still be assigned overlapping or ambiguous names. To mitigate this, we

build a label similarity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex $v_i \in \mathcal{V}$ represents a known or induced label y_i, and each edge $(v_i, v_j) \in \mathcal{E}$ encodes their semantic similarity. Given label embeddings h_{y_i} and h_{y_j} obtained from the label encoder, we define the edge weight as the cosine similarity:

$$\mathrm{sim}(y_i, y_j) = \frac{h_{y_i}^\top h_{y_j}}{\|h_{y_i}\| \|h_{y_j}\|}. \tag{8}$$

An undirected edge is established if $\mathrm{sim}(y_i, y_j) \geq \theta_{\text{sim}}$, forming a semantic graph that connects labels with potentially overlapping meanings. We then identify connected components $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M$ within $\mathcal{G}$, where each component $\mathcal{C}_m$ represents a group of semantically confusable categories.

We distinguish between two scenarios according to the composition of each connected component during training. For *pure novel groups*, where a component contains only novel clusters, we employ the LLM to simultaneously rename all clusters within the group, enforcing pairwise distinctiveness among the induced labels (see Prompt 2 in Appendix E.1.2). For *mixed groups*, where novel clusters co-occur with semantically close known categories, the LLM is instructed to generate new labels that are not only mutually distinctive but also clearly separated from their known neighbors (see Prompt 3 in Appendix E.1.3).

To enhance the reliability of label refinement, we perform local confidence recalibration within each component $\mathcal{C}_m$. Specifically, we restrict the confidence computation to the subset of clusters $\{\tilde{C}_i \mid \tilde{C}_i \in \mathcal{C}_m\}$, and re-evaluate the confidence scores for samples assigned to these clusters as:

$$\tilde{\mathrm{conf}}(x, \tilde{C}_i) = \frac{d(x, \tilde{C}_j) - d(x, \tilde{C}_i)}{d(\tilde{C}_i, \tilde{C}_j)}, \tag{9}$$

where $\tilde{C}_j$ is the nearest alternative cluster in $\mathcal{C}_m$.

This local normalization ensures that the locally recalibrated high-confidence samples used for LLM-based refinement are representative of confusable clusters within the same component, rather than globally dominant outliers. Through this refinement step, semantically entangled clusters are disambiguated and assigned mutually exclusive, interpretable names, significantly improving the consistency and readability of induced label semantics.

4.3 Confidence-aware Sampling and Semi-supervised Training

However, a cluster label may not apply to every instance within the cluster, particularly for samples near decision boundaries. We therefore introduce an LLM-guided, confidence-aware sampling strategy to retain instances that are truly consistent with the cluster label, providing a second-stage verification after cluster discovery and label induction.

LLM-guided Confidence-aware Sampling Based on precomputed $\mathrm{conf}(x, C_i)$, unlabeled samples in each cluster C_i are partitioned into high-, medium-, and low-confidence sets, denoted as $\mathcal{D}^{u,i}_{\text{high}}$, $\mathcal{D}^{u,i}_{\text{mid}}$, and $\mathcal{D}^{u,i}_{\text{low}}$. At epoch e, we rank samples in C_i by $\mathrm{conf}(x, C_i)$ in descending order and partition them by adaptive ratio r^h_e and fixed ratio r^l:

$$\begin{aligned}
\mathcal{D}^{u,i}_{\text{high}} &= \{\, x \in C_i \mid \mathrm{rank}(x) \leq r^h_e \,|C_i|\,\}, \\
\mathcal{D}^{u,i}_{\text{mid}} &= \{\, x \in C_i \mid r^h_e \,|C_i| < \mathrm{rank}(x) \leq (1-r^l)\,|C_i|\,\}, \\
\mathcal{D}^{u,i}_{\text{low}} &= \{\, x \in C_i \mid \mathrm{rank}(x) > (1-r^l)\,|C_i|\,\},
\end{aligned} \tag{10}$$

For each cluster C_i, we randomly sample unlabeled k_h, k_m, and k_l instances from $\mathcal{D}^{u,i}_{\text{high}}$, $\mathcal{D}^{u,i}_{\text{mid}}$, and $\mathcal{D}^{u,i}_{\text{low}}$, respectively, to form the queried candidate set $\widetilde{\mathcal{D}}^u_e$, and send them to the LLM together with their label of the cluster. The LLM returns a binary decision $\mathrm{verify}(x, C_i) \in \{0, 1\}$ for each queried sample. The final verified pseudo-labeled set for epoch e is:

$$\widehat{\mathcal{D}}^u_e = \{(x, \hat{y}) \mid x \in \widetilde{\mathcal{D}}^u_e,\ \mathrm{verify}(x, C_i) = 1\},$$

where $\hat{y}$ denotes the matched known-class label for aligned-to-known clusters or an LLM-induced label for novel clusters. Details of the sample-selection prompt are provided in Appendix E.1.4.

Iterative Semi-supervised Training After pseudo-label filtering, we retrain the model in an iterative semi-supervised fashion. At each epoch e, the training set is defined as the union of labeled and verified pseudo-labeled samples:

$$\mathcal{D}^{\text{train}}_e = \mathcal{D}^l \cup \widehat{\mathcal{D}}^u_e. \tag{11}$$

To incorporate more uncertain yet potentially informative samples, we dynamically anneal the high-confidence ratio r^h_e across epochs. Specifically, r^h_e is linearly increased from an initial conservative value r^h_0 (e.g., 0.15) to a larger value r^h_T (e.g., 0.40) as training proceeds. This adaptive relaxation can be formulated as:

$$r^h_e = (1 - t)\, r^h_0 + t\, r^h_T, \quad t = \frac{e}{E - 1}, \tag{12}$$

where E is the total number of semi-supervised epochs.

During each epoch, the model parameters are optimized with the proposed loss $\mathcal{L}_{\text{train}}$ in Eq. 6.

The process is iterated over epochs, alternating between clustering, label induction, LLM-based verification, and retraining. As the model evolves, cluster boundaries become sharper, confidence estimates more accurate, and pseudo-label quality increasingly reliable. This dynamic ratio scheduling enables the model to gradually transition from conservative pseudo-label selection to more inclusive learning, thereby achieving robust representations for both known and novel categories.

5 Experiment

5.1 Experimental Settings

Datasets Following prior work (Liang et al., 2024; Zou et al., 2025), we evaluate our GCD framework on three widely-used English benchmarks: BANKING (Casanueva et al., 2020), CLINC (Larson et al., 2019), and HWU (Liu et al., 2021). We also incorporate a Chinese dataset, THUCNews.* Further details are provided in Appendix C.2. We followed the standard splitting commonly employed in GCD (Zhang et al., 2024). Specifically, we considered three experimental settings in which 25%, 50% and 75% of the total categories were designated as known classes $\mathcal{Y}^k$, while the remaining were treated as novel classes $\mathcal{Y}^n$. Within the training data, only 10% of the samples from $\mathcal{Y}^k$ were used as labeled training data, and the rest, together with all samples from $\mathcal{Y}^n$, formed the unlabeled training set.

Baselines We compared our approach with two categories of SOTA methods for GCD. (1) **Traditional GCD methods**, including DeepAligned (Zhang et al., 2021), DPN (An et al., 2023b), GeoID (Tang et al., 2024), KTN (Shi et al., 2024), PTJN (An et al., 2023a), SDC (An et al., 2025), TAN (An et al., 2024b), and USNID (Zhang et al., 2024). (2) **LLM-enhanced GCD methods**, including LOOP (An et al., 2024a), ALUP (Liang et al., 2024), and GLEAN (Zou et al., 2025), which integrated LLMs into the discovery process. Further details of these baselines are provided in Appendix C.3.

Implementation Details We adopted dataset-appropriate BERT (Devlin et al., 2019) backbones as the shared encoder: BERT-base-uncased for the English datasets and BERT-base-Chinese for THUCNews, each equipped with two separate two-layer MLP projection heads for text and label encoding. The model was optimized using AdamW. For the symmetric difficulty-aware reweighted contrastive loss (Eq. (6)), we set $\alpha = 0.5$ and top-k with $k = 5$. To further incorporate LLM capabilities, locally deployed DeepSeek-V3 (Liu et al., 2025) was employed as the LLM backend. During training, the high-confidence ratio r_e^h was linearly annealed from 0.15 to 0.40 across epochs. For confidence-aware representative sampling, we set $k_{\text{high}} = 8, k_{\text{mid}} = 12$, and $k_{\text{low}} = 16$. All experiments were conducted with three random seeds on a single NVIDIA RTX 4090 GPU. To ensure a fair comparison, we re-implemented all baseline methods using the same dataset-appropriate BERT backbone as TLSA on each dataset and followed the hyperparameter settings reported in their original papers. For LLM-enhanced methods, DeepSeek-V3 served as the backend to guarantee experimental comparability.

Evaluation Protocol Following prior work on GCD (Zhang et al., 2024), we evaluate performance using three standard metrics: overall accuracy (ACC), Adjusted Rand Index (ARI), and Normalized Mutual Information (NMI). While the LLM is used for training, inference follows the standard GCD protocol using only the text encoder. Predicted clusters are aligned to ground-truth labels using the Hungarian algorithm (Kuhn, 1955). Formal definitions are provided in Appendix C.4.

5.2 Overall Performance

The overall performance under different known-class ratios was presented in Table 1. As shown, our method outperformed all baselines in the vast majority of cases across datasets and known ratios, demonstrating its robustness and effectiveness.

Firstly, our method demonstrated superior performance across all known-class ratios, with particularly strong gains observed at $\rho = 0.25$, where it achieved an average relative improvement of 3.56% in ACC, 4.20% in ARI, and 1.55% in NMI across datasets. These results highlight the effectiveness of our approach under low-resource conditions, where limited labeled information poses significant challenges. Additionally, LLM-enhanced methods generally outperformed traditional approaches in most cases, reinforcing the advantage of leveraging external semantic knowledge. Even when $\rho = 0.75$, where sufficient labeled data were available, our method continued to deliver strong and stable performance. This demonstrated that the

*http://thuctc.thunlp.org/

Known Ratio	Method	BANKING			CLINC			HWU			THUCNews		
		ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI
$\rho = 0.25$	DeepAligned	45.87	34.35	68.65	75.47	65.83	89.43	53.81	42.56	75.17	47.30	32.10	54.76
	DPN	45.27	34.41	72.75	74.46	64.94	92.36	52.07	43.42	78.38	63.91	45.97	68.98
	GeoID	38.31	27.03	60.48	83.80	77.26	93.50	67.80	54.33	80.40	64.03	48.34	67.65
	KTN	41.45	27.46	65.78	70.43	58.87	88.19	59.27	44.21	76.45	73.16	62.09	75.81
	PTJN	51.51	39.31	71.66	79.23	71.01	91.32	56.69	45.02	76.08	61.47	48.46	64.94
	SDC	48.01	36.44	69.21	66.65	55.91	85.95	58.78	45.25	75.75	64.66	47.73	67.19
	TAN	42.16	28.43	64.31	58.86	46.93	81.86	44.25	29.34	66.89	67.22	52.36	69.61
	USNID	67.44	57.16	81.90	82.75	77.68	93.99	72.58	62.19	84.71	67.38	52.40	69.81
	LOOP	70.63	60.71	83.16	87.79	82.45	94.92	71.16	60.74	83.97	40.06	20.93	40.32
	ALUP	72.32	62.33	84.22	87.24	81.96	95.04	70.16	59.42	83.05	73.63	63.81	75.89
	GLEAN	66.36	56.91	82.02	85.79	80.36	94.67	69.93	58.02	83.24	46.71	27.98	49.59
	TLSA	**74.97**	**65.31**	**85.82**	**90.67**	**86.92**	**96.33**	**76.87**	**65.74**	**85.71**	**74.65**	**64.38**	**77.23**
$\rho = 0.5$	DeepAligned	58.22	46.87	76.27	81.07	73.10	91.96	62.92	51.86	79.67	60.53	39.17	59.15
	DPN	57.03	46.99	79.49	79.78	73.53	94.14	61.53	50.06	81.29	67.57	43.78	68.06
	GeoID	68.98	58.07	81.63	87.91	81.90	94.78	69.70	59.35	83.00	70.29	54.94	71.14
	KTN	62.36	49.54	77.88	80.34	74.08	93.33	71.58	59.13	82.61	78.00	62.55	74.93
	PTJN	64.01	51.75	78.59	84.00	76.57	93.04	66.60	54.27	80.36	64.94	50.67	68.08
	SDC	61.54	49.92	77.77	75.01	67.52	90.45	65.15	52.62	79.71	76.82	59.69	73.21
	TAN	56.74	44.90	74.51	72.41	63.67	88.82	54.04	38.98	73.36	67.30	47.39	65.48
	USNID	70.92	62.16	84.82	86.80	81.87	95.28	77.49	67.94	86.95	71.39	52.19	69.46
	LOOP	72.76	63.18	84.58	87.07	82.07	94.99	74.19	64.14	84.99	48.95	26.02	45.43
	ALUP	72.79	64.03	85.33	88.70	84.21	95.57	73.39	62.52	84.25	75.09	60.58	73.62
	GLEAN	72.69	63.64	85.39	83.66	75.61	93.74	77.43	65.98	85.93	55.57	34.21	55.07
	TLSA	**78.64**	**68.21**	**86.82**	**92.27**	**88.29**	**96.68**	**79.72**	**68.92**	**87.02**	**80.91**	**70.61**	**81.39**
$\rho = 0.75$	DeepAligned	66.13	54.69	80.32	86.41	80.06	94.04	71.55	59.61	82.21	82.29	72.55	80.34
	DPN	70.38	60.68	84.04	88.95	84.15	96.03	76.94	66.38	86.53	81.66	68.63	79.65
	GeoID	76.13	66.09	85.68	91.69	86.97	96.11	76.48	66.00	85.51	73.83	61.08	76.16
	KTN	76.54	65.85	85.19	90.78	86.41	96.15	79.14	67.61	85.92	81.89	67.94	77.71
	PTJN	73.33	60.46	82.12	87.47	81.79	94.68	73.06	60.71	82.81	76.98	65.72	76.51
	SDC	72.46	60.28	82.46	83.33	76.97	93.33	72.95	60.42	82.97	82.37	67.76	77.16
	TAN	68.81	56.61	80.93	82.49	75.43	92.58	69.64	56.29	81.03	82.64	72.56	80.25
	USNID	77.00	69.09	87.44	90.06	86.45	96.36	78.65	69.11	87.23	84.02	76.37	83.21
	LOOP	73.80	64.90	85.41	90.65	86.18	95.95	73.74	63.80	85.21	67.57	50.34	63.74
	ALUP	77.19	68.44	86.92	90.95	86.66	96.16	77.39	67.45	86.41	78.86	65.31	76.68
	GLEAN	80.69	71.96	**88.26**	86.84	73.93	93.54	80.26	70.04	87.54	74.26	58.25	72.97
	TLSA	**81.43**	**72.14**	87.83	**95.32**	**92.06**	**97.49**	**81.07**	**70.60**	**87.58**	**85.04**	**77.43**	**83.22**

Table 1: Overall performance on GCD task. Best in **bold**, second-best underlined. Results averaged over 3 seeds.

proposed method effectively sharpened clustering boundaries and enhanced representation discrimination, contributing to consistent performance.

5.3 Ablation Studies

To further assess the contribution of individual components within our method, we conducted an ablation study by introducing four variants: (1) **w/o Semantic Label Refinement for Confusable Categories**: disables the semantic label refinement module designed to disambiguate confusable categories; (2) **w/o Dynamic Confidence Ratios r_e^h**: fixes the confidence ratios across epochs instead of adapting them dynamically; (3) **w/o Symmetric Difficulty-aware Reweighted Contrastive Loss**: replaces the proposed contrastive loss with standard cross-entropy; (4) **w/o Confidence-aware Sampling**: removes the LLM-guided confidence-aware sampling.

The results are shown in Table 2. Removing semantic label refinement consistently degrades performance, as semantically similar clusters remain entangled and blur decision boundaries. Fixing the dynamic confidence ratios across epochs leads to further decline because the model can no longer adapt to evolving cluster densities during training. Replacing the reweighted contrastive loss with standard cross-entropy causes moderate degradation by weakening semantic alignment between known and novel categories. Finally, removing confidence-aware sampling results in the most substantial performance drop, indicating that the model struggles to identify reliable pseudo-labels without intelligent sample selection.

5.4 Impact of Hard Negative Top-k

To investigate the impact of selecting different numbers of hard negatives in the difficulty-aware ranking loss, we evaluated various values of $k \in \{0, 5, 10, 20, 30\}$ on the BANKING and HWU datasets where the known-class ratio ρ is 0.25.

As shown in Table 3, setting $k = 5$ achieved the

	$\rho = 0.25$			$\rho = 0.50$			$\rho = 0.75$		
	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI
TLSA	**74.97**	**65.31**	**85.82**	**78.64**	**68.21**	**86.82**	**81.43**	**72.14**	**87.83**
w/o Semantic Label Refinement	73.48	63.42	85.21	76.37	66.70	86.61	79.53	69.85	87.35
w/o Dynamic Confidence Ratios	73.41	63.80	85.34	76.07	66.46	86.39	79.19	70.23	87.55
w/o Reweighted Contrastive Loss	72.45	62.36	84.86	76.34	66.54	86.29	79.30	69.45	87.36
w/o Confidence-aware Sampling	58.01	45.17	74.82	63.60	51.73	78.47	70.18	58.63	81.86

Table 2: Ablation studies on the BANKING dataset.

Top-k	BANKING			HWU		
	ACC	ARI	NMI	ACC	ARI	NMI
0	73.64	64.20	85.58	76.33	65.58	85.61
5	**74.97**	**65.31**	**85.82**	**76.87**	**65.74**	**85.71**
10	74.31	65.06	85.60	75.03	64.22	85.48
20	74.13	64.25	85.73	75.53	63.97	85.15
30	73.81	64.42	85.71	75.15	64.61	85.41

Table 3: Performance comparison across different top-k values on BANKING and HWU ($\rho = 0.25$).

α	BANKING			HWU		
	ACC	ARI	NMI	ACC	ARI	NMI
0.1	72.83	62.90	84.98	75.19	64.52	85.39
0.5	**74.97**	**65.31**	**85.82**	**76.87**	**65.74**	**85.71**
1	73.70	64.67	85.73	75.24	64.71	85.51
2	72.80	63.64	85.50	76.09	65.61	85.65

Table 4: Performance comparison across different α values on BANKING and HWU ($\rho = 0.25$).

best performance. When $k = 0$, the ranking loss is disabled, limiting the model's ability to perform fine-grained discrimination. Moderate values of k (e.g., 5 or 10) effectively concentrated the ranking penalty on the most confusing negatives. In contrast, larger values tended to diffuse the learning signal across too many samples, weakening the model's focus on critical boundary cases and degrading performance.

5.5 Impact of Ranking Loss Weight α

We further examined the effect of the ranking loss weight α in Eq. (6), which controlled the trade-off between the symmetric contrastive loss $\mathcal{L}_{\text{cl}}$ and the difficulty-aware ranking loss $\mathcal{L}_{\text{rank}}$ for hard negatives. To analyze its influence, we conducted experiments on the BANKING and HWU datasets under a low known-class ratio setting where $\rho = 0.25$, varying α among 0.1, 0.5, 1, and 2.

The results, summarized in Table 4, showed that $\alpha = 0.5$ achieved the best performance, suggesting that this configuration achieved an effective balance between the contrastive loss and the hard-negative loss. When α is too small, the ranking loss was underweighted, reducing the model's ability to capture subtle semantic distinctions among confusing negatives. Conversely, excessively large values placed disproportionate emphasis on hard negatives, which can lead to overfitting on challenging samples. These observations underscore the importance of appropriately balancing the two loss components to ensure both robust generalization and fine-grained discriminative capability.

(k_h, k_m, k_l)	BANKING			HWU		
	ACC	ARI	NMI	ACC	ARI	NMI
Balanced Configurations						
(2, 4, 6)	72.39	62.25	84.09	72.93	61.19	83.51
(4, 8, 12)	73.13	63.59	85.11	74.64	62.94	84.72
(8, 12, 16)	**74.97**	**65.31**	85.82	**76.87**	**65.74**	**85.71**
(12, 18, 24)	74.44	65.18	**86.00**	75.74	64.78	85.65
(15, 30, 45)	72.87	63.76	85.49	72.16	60.68	84.19
Single-level Configurations						
(36, 0, 0)	64.41	51.43	78.39	70.15	55.75	80.58
(0, 36, 0)	71.95	59.87	83.29	70.90	58.78	82.66
(0, 0, 36)	73.36	63.51	85.42	74.13	62.97	84.72

Table 5: Performance comparison across different sampling configurations on BANKING and HWU ($\rho = 0.25$).

5.6 Impact of Confidence-aware Sampling Strategies

To investigate how different sampling configurations affect model performance, we conducted experiments on the BANKING and HWU datasets with known-class ratio $\rho = 0.25$. Table 5 summarized the results across five balanced configurations with varying sample sizes, as well as three single-level settings that utilize only one confidence subset, where k_h, k_m, and k_l denoted the numbers of samples drawn from the high-, medium-, and low-confidence subsets, respectively.

Among all configurations, the balanced setting of $(8, 12, 16)$ achieved the best performance. Balanced configurations outperformed single-level variants, suggesting that combining samples of varying confidence provides complementary sig-

Strategy	ACC	ARI	NMI
Direct	74.97	65.31	85.82
ICL	75.34	65.50	85.99
CoT	76.14	65.96	**86.18**
Self-Refine	**76.22**	**66.43**	86.13

Table 6: Performance comparison across prompting strategies on BANKING ($\rho = 0.25$).

nals: higher-confidence instances anchor stable supervision, while lower-confidence instances capture harder yet still label-consistent cases near boundaries. Additionally, both excessively small and large balanced configurations led to diminished effectiveness. Smaller configurations restrict feedback diversity, limiting the model's exposure to representative variations, whereas larger configurations introduce noisy samples.

5.7 Impact of Different LLMs and Prompts

Recently, numerous studies have sought to enhance the text comprehension capabilities of LLMs through the design of diverse prompting strategies. To evaluate the robustness and generalizability of TLSA, we conducted experiments under four representative prompting paradigms: Direct, In-Context Learning (Brown et al., 2020), Chain-of-Thought (Wei et al., 2022), and Self-Refine (Madaan et al., 2023), where the Direct strategy served as the default setting for the main experiment. Detailed prompt designs are provided in Appendix E.2. As shown in Table 6, our method achieved consistently strong performance across multiple advanced prompting strategies, indicating that while LLM prompting incurs higher cost, it also brings notable improvements in performance. We also evaluated our approach with different LLM backbones, and the detailed results are provided in Appendix D.

6 Conclusion

In this work, we proposed TLSA, a novel GCD framework that integrates a dual-encoder with LLM-driven semantic refinement. Our method explicitly models the semantics of texts and labels and aligns them in a shared embedding space to achieve robust contrastive alignment. Furthermore, we introduced LLM-guided semantic label refinement and confidence-aware sampling. Extensive experiments on four benchmark datasets show that TLSA achieves the best overall average performance, demonstrating its effectiveness.

Limitations

Despite the strong performance, our method has several limitations. First, while evaluations in Appendix D demonstrate that smaller open-source models can achieve competitive results, the framework's overall effectiveness remains inevitably influenced by the inherent performance and biases of the chosen LLM. This sensitivity may be more pronounced in highly specialized domains where model knowledge may be limited. Second, integrating LLMs introduces additional challenges in terms of data privacy, LLM interaction latency, and request overhead. In this study, we attempted to alleviate these concerns by employing a locally deployed DeepSeek-V3 model to enhance security and minimize communication delays.

Ethical Considerations

This study is conducted entirely on publicly available benchmark datasets released for research use. We do not collect new personal data, and our experiments operate on the datasets as provided by their original sources. We follow standard research practices regarding data usage, privacy, and responsible reporting.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) (Grant No. 62506352) and the National Natural Science Foundation of China (NSFC) (No.92470204).

References

Wenbin An, Haonan Lin, Jiahao Nie, Feng Tian, Wenkai Shi, Yaqiang Wu, Qianying Wang, and Ping Chen. 2025. Unleashing the potential of model bias for generalized category discovery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(15):15365–15373.

Wenbin An, Wenkai Shi, Feng Tian, Haonan Lin, QianYing Wang, Yaqiang Wu, Mingxiang Cai, Luyan Wang, Yan Chen, Haiping Zhu, and Ping Chen. 2024a. Generalized category discovery with large language models in the loop. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8653–8665, Bangkok, Thailand. Association for Computational Linguistics.

Wenbin An, Feng Tian, Ping Chen, Qinghua Zheng, and Wei Ding. 2023a. New user intent discovery with robust pseudo label training and source domain joint training. *IEEE Intelligent Systems*, 38(4):21–31.

Wenbin An, Feng Tian, Wenkai Shi, Yan Chen, Yaqiang Wu, Qianying Wang, and Ping Chen. 2024b. Transfer and alignment network for generalized category discovery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):10856–10864.

Wenbin An, Feng Tian, Qinghua Zheng, Wei Ding, Qianying Wang, and Ping Chen. 2023b. Generalized category discovery with decoupled prototypical network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):12527–12535.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Xi Chen, Chuan Qin, Chuyu Fang, Chao Wang, Chen Zhu, Fuzhen Zhuang, Hengshu Zhu, and Hui Xiong. 2024. Job-sdf: A multi-granularity dataset for job skill demand forecasting and benchmarking. *Advances in Neural Information Processing Systems*, 37:129329–129356.

Xi Chen, Chuan Qin, Ziqi Wang, Shasha Hu, Chao Wang, Hengshu Zhu, and Hui Xiong. 2026. Beyond the known: An unknown-aware large language model for open-set text classification. In *The Fourteenth International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaohan Huang, Meng Xiao, Chuan Qin, Qingqing Long, Jinmiao Chen, Yuanchun Zhou, and Hengshu Zhu. 2026. Scihorizon-gene: Benchmarking llm for life sciences inference from gene knowledge to functional understanding. *Preprint*, arXiv:2601.12805.

Feihu Jiang, Chuan Qin, Kaichun Yao, Chuyu Fang, Fuzhen Zhuang, Hengshu Zhu, and Hui Xiong. 2024. Enhancing question answering for enterprise knowledge bases using large language models. In *Database Systems for Advanced Applications: 29th International Conference, DASFAA 2024, Gifu, Japan, July 2–5, 2024, Proceedings, Part IV*, page 273–290, Berlin, Heidelberg. Springer-Verlag.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Jinggui Liang, Lizi Liao, Hao Fei, Bobo Li, and Jing Jiang. 2024. Actively learn from LLMs with uncertainty propagation for generalized category discovery. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7845–7858, Mexico City, Mexico. Association for Computational Linguistics.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8360–8367.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. *Benchmarking Natural Language Understanding Services for Building Conversational Agents*, pages 165–183. Springer Singapore, Singapore.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Sridhama Prakhya, Vinodini Venkataram, and Jugal Kalita. 2017. Open set text classification using CNNs. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*,

pages 466–475, Kolkata, India. NLP Association of India.

Chuan Qin, Xin Chen, Chengrui Wang, Pengmin Wu, Xi Chen, Yihang Cheng, Jingyi Zhao, Meng Xiao, Xiangchao Dong, Qingqing Long, Boya Pan, Han Wu, Chengzan Li, Yuanchun Zhou, Hui Xiong, and Hengshu Zhu. 2025a. Scihorizon: Benchmarking ai-for-science readiness from scientific data to large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 5754–5765, New York, NY, USA. Association for Computing Machinery.

Chuan Qin, Chuyu Fang, Kaichun Yao, Xi Chen, Fuzhen Zhuang, and Hengshu Zhu. 2025b. Cotr: Efficient job task recognition for occupational information systems with class-incremental learning. *ACM Trans. Manage. Inf. Syst.*, 16(2).

Chuan Qin, Le Zhang, Yihang Cheng, Rui Zha, Dazhong Shen, Qi Zhang, Xi Chen, Ying Sun, Chen Zhu, Hengshu Zhu, and Hui Xiong. 2025c. A comprehensive survey of artificial intelligence techniques for talent analytics. *Proceedings of the IEEE*, 113(2):125–171.

Wenkai Shi, Wenbin An, Feng Tian, Yan Chen, Yaqiang Wu, Qianying Wang, and Ping Chen. 2024. A unified knowledge transfer network for generalized category discovery. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18961–18969.

Zhiheng Song, Jingshuai Zhang, Chuan Qin, Chao Wang, Chao Chen, Longfei Xu, Kaikui Liu, Xiangxiang Chu, and Hengshu Zhu. 2026. Mobility-bench: A benchmark for evaluating route-planning agents in real-world mobility scenarios. *Preprint*, arXiv:2602.22638.

Kai Tang, Junbo Zhao, Xiao Ding, Runze Wu, Lei Feng, Gang Chen, and Haobo Wang. 2024. Learning geometry-aware representations for new intent discovery. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5641–5654, Bangkok, Thailand. Association for Computational Linguistics.

Zhenyu Tong, Chuan Qin, Chuyu Fang, Kaichun Yao, Xi Chen, Jingshuai Zhang, Chen Zhu, and Hengshu Zhu. 2025. From missteps to mastery: Enhancing low-resource dense retrieval through adaptive query generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, page 1373–1384, New York, NY, USA. Association for Computing Machinery.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. 2022. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7492–7501.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021. Discovering new intents with deep aligned clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14365–14373.

Hanlei Zhang, Hua Xu, Xin Wang, Fei Long, and Kai Gao. 2024. A clustering framework for unsupervised and semi-supervised new intent discovery. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):5468–5481.

Henry Peng Zou, Siffi Singh, Yi Nian, Jianfeng He, Jason Cai, Saab Mansour, and Hang Su. 2025. Glean: Generalized category discovery with diverse and quality-enhanced llm feedback. *Preprint*, arXiv:2502.18414.

A LLM Usage Statement

We used large language models (LLMs) in two specific ways during this research. First, LLMs were employed to polish the writing and improve the clarity of this paper. Second, LLMs were integrated into our proposed method as part of the label generation, refinement, and sample filtering processes. No LLMs were used for idea conception, model architecture design, algorithm development, or experimental evaluation. All methodological decisions and empirical analyses were carried out independently by the authors.

B Hungarian Algorithm Details for Known-class Alignment

To distinguish known categories from novel ones within the discovered clusters, we formulate the alignment between unlabeled cluster centroids and labeled known-class anchors as a maximum-weight bipartite matching problem. Let $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ be the set of centroids obtained from K-Means clustering on the unlabeled data $\mathcal{D}^u$, where $K = |\mathcal{Y}^k| + |\mathcal{Y}^n|$ is the total number of clusters. Let $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{|\mathcal{Y}^k|}\}$ be the precomputed prototypes for known classes, where each $\mathbf{p}_j$ is the mean embedding of labeled instances belonging to class $y_j^k \in \mathcal{Y}^k$ extracted by the encoder. The similarity between an unlabeled cluster centroid $\mathbf{c}_i$ and a known-class prototype $\mathbf{p}_j$ is defined by cosine similarity:

$$S_{i,j} = \frac{\mathbf{c}_i^\top \mathbf{p}_j}{\|\mathbf{c}_i\| \|\mathbf{p}_j\|}. \tag{13}$$

We construct a similarity matrix $\mathbf{S} \in \mathbb{R}^{K \times |\mathcal{Y}^k|}$. The objective is to find an optimal injective mapping $\sigma : \{1, \ldots, |\mathcal{Y}^k|\} \to \{1, \ldots, K\}$ that maximizes the aggregate similarity:

$$\max_{\sigma} \sum_{j=1}^{|\mathcal{Y}^k|} S_{\sigma(j),j}, \tag{14}$$

subject to $\sigma(j) \neq \sigma(j')$ for $j \neq j'$. To solve this using the Hungarian algorithm (Kuhn, 1955), we augment $\mathbf{S}$ into a square matrix $\tilde{\mathbf{S}} \in \mathbb{R}^{K \times K}$ by adding $K - |\mathcal{Y}^k|$ dummy columns with zero entries. The cost matrix $\tilde{\mathbf{C}}$ is then defined as $\tilde{\mathbf{C}}_{i,j} = -\tilde{\mathbf{S}}_{i,j}$. The algorithm identifies a permutation π of $\{1, \ldots, K\}$ that minimizes $\sum_{i=1}^{K} \tilde{\mathbf{C}}_{i,\pi(i)}$. Clusters matched to the original $|\mathcal{Y}^k|$ columns are assigned to the corresponding known classes, while the remaining clusters are identified as candidates for novel categories.

C Experimental Details

C.1 Experimental Setup Details

Our experiments use dataset-appropriate BERT backbones: BERT-base-uncased for the English datasets and BERT-base-Chinese for THUCNews. All experiments can be executed on a single consumer-grade GPU. The overall running time is largely influenced by the latency of LLM queries. We build our framework with open-source software packages, including PyTorch, Transformers, scikit-learn, and NumPy. Default hyperparameter configurations and additional training details are provided in the released code repository.

C.2 Datasets

(1) **BANKING** (Casanueva et al., 2020), an intent recognition dataset containing 13,073 customer service queries labeled with 77 fine-grained intents; (2) **CLINC** (Larson et al., 2019), an out-of-scope intent detection dataset comprising 22,500 utterances spanning 150 intents; (3) **HWU** (Liu et al., 2021), a spoken language understanding dataset consisting of 25,716 examples across 64 intents; and (4) **THUCNews**, a Chinese news classification dataset from which we randomly sampled 8,348 titles covering 14 distinct categories for GCD. For this Chinese dataset, both the baselines and our method replaced the English BERT-base-uncased with BERT-base-Chinese to better capture its linguistic features. To ensure fairness and reproducibility, all datasets were pre-partitioned into fixed training, validation, and testing sets, which were strictly maintained across all experiments and baselines.

C.3 Baselines

We provide detailed introductions of the baselines used in our experiments. (1) **Traditional GCD methods.** These methods mainly rely on clustering, representation learning, or knowledge transfer without the involvement of large language models. They include DeepAligned (Zhang et al., 2021), USNID (Zhang et al., 2024), DPN (An et al., 2023b), TAN (An et al., 2024b), GeoID (Tang et al., 2024), KTN (Shi et al., 2024), PTJN (An et al., 2023a), and SDC (An et al., 2025). They represent the evolution from early clustering-based meth-

ods with alignment and pseudo-label refinement, to recent transfer- and geometry-aware methods that explicitly enhance knowledge sharing across categories. (2) **LLM-enhanced GCD methods.** These approaches leverage large language models to provide additional information. They include LOOP (An et al., 2024a), ALUP (Liang et al., 2024), and GLEAN (Zou et al., 2025), which integrate LLMs into the discovery process through active querying, uncertainty propagation, and feedback quality control.

DeepAligned Deep Aligned Clustering (Zhang et al., 2021) leverages limited labeled intents to guide clustering of both known and novel ones. It pre-trains a BERT encoder and applies K-Means clustering to generate pseudo labels. To stabilize training, cluster centroids across epochs are aligned via the Hungarian algorithm, producing consistent pseudo labels instead of noisy reinitialization.

USNID USNID (Zhang et al., 2024) is a unified clustering framework for unsupervised and semi-supervised new intent discovery. It learns robust intent representations with contrastive pre-training, stabilizes clustering via centroid-guided alignment, and jointly optimizes cluster-level and instance-level objectives.

DPN The Decoupled Prototypical Network (An et al., 2023b) decouples prototypes of known and novel categories. Known prototypes are aligned with labeled classes via bipartite matching, while unmatched prototypes represent novel categories. It further introduces semantic-aware prototypical learning to capture semantic relations and mitigate pseudo-label noise, and regularizes known prototypes with labeled supervision.

TAN The Transfer and Alignment Network (An et al., 2024b) explicitly transfers supervision from known to novel categories. It learns transferable prototypes and aligns them across domains to reduce distribution shift. TAN alternates between supervised training on labeled data and pseudo-label refinement on unlabeled data, effectively mitigating bias towards known categories and improving clustering for novel intents.

GeoID GeoID (Tang et al., 2024) learns geometry-aware representations for new intent discovery, inspired by Neural Collapse. It fixes classifier weights to a Simplex Equiangular Tight Frame (ETF) and aligns cluster assignments via optimal transport and k-means matching. Dual pseudo-labeling and contrastive learning enforce intra-class compactness and inter-class separation, producing near-optimal cluster geometry.

KTN The Knowledge Transfer Network (Shi et al., 2024) transfers knowledge from known to novel categories via entropy-based soft differentiation and prototype-based transfer weights. Logit- and label-level adjustments reduce confusion for known samples and propagate logits to semantically similar novel categories. KTN combines cross-entropy, consistency, and InfoNCE losses, and supports online inference with fast speed.

PTJN PTJN (An et al., 2023a) introduces an Extractor–Generator–Corrector architecture for robust pseudo-label training. Generator produces epoch-level pseudo labels, while Corrector generates iteration-level refinements to mitigate mismatches. Joint training on labeled data preserves source knowledge, and adaptive voting integrates multiple predictions at inference.

SDC Self-Debiasing Calibration (An et al., 2025) leverages the bias of a pre-trained model instead of discarding it. Category Bias Mitigation subtracts biased logits from known categories, while Category Confusion Mitigation transfers biased logits from semantically similar known to novel categories. An entropy-based weighting mechanism adapts debiasing to known vs. novel samples.

LOOP LOOP (An et al., 2024a) is an active learning framework with LLMs in the loop. It selects misclustered samples via Local Inconsistent Sampling, queries LLMs to refine neighbor assignments, and trains with Refined Neighborhood Contrastive Learning. Finally, it queries LLMs to generate semantic names for novel clusters, combining improved clustering with interpretability.

ALUP ALUP (Liang et al., 2024) integrates LLMs with active learning and uncertainty propagation. Uncertainty Propagation selects representative uncertain samples, Comparison-based Prompting queries LLMs with comparative judgments, and Soft Feedback Propagation propagates soft labels to neighbors.

GLEAN GLEAN (Zou et al., 2025) improves LLM-based GCD by incorporating diverse and quality-enhanced feedback. It combines similar instance selection, category characterization, and

pseudo-category alignment with quality investigation and filtering. It also introduces mechanisms to assess and enhance the reliability of LLM feedback before propagation to prevent error accumulation.

C.4 Evaluation Metrics

In the experiments, we employ three standard evaluation metrics to evaluate the performance of GCD models: ACC, ARI, and NMI. Their definitions are as follows.

Clustering Accuracy (ACC). ACC measures the best one-to-one correspondence between predicted clusters and ground-truth classes. It is defined as

$$\mathrm{ACC} = \max_{m \in \mathcal{M}} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\big(m(\hat{y}_i) = y_i\big),$$

where N is the total number of samples, $\hat{y}_i$ and y_i denote the predicted and true labels of the i-th sample, respectively, and m ranges over all possible one-to-one mappings between predicted clusters and true classes. Higher ACC indicates better alignment between predicted clusters and true categories.

Adjusted Rand Index (ARI). ARI measures the similarity between two partitions (predicted and true labels) by considering all pairs of samples and counting those that are assigned consistently. It corrects the Rand Index (RI) for chance:

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max(\mathrm{RI}) - \mathbb{E}[\mathrm{RI}]},$$

where $\mathrm{RI} = \frac{a+b}{\binom{N}{2}}$, with a denoting the number of sample pairs placed in the same cluster in both partitions, and b denoting the number of pairs placed in different clusters in both partitions. ARI ranges from -1 to 1, where 1 indicates perfect agreement and values close to 0 indicate random assignment.

Normalized Mutual Information (NMI). NMI evaluates the mutual dependence between the predicted clustering and the true labels, normalized by their entropies:

$$\mathrm{NMI} = \frac{I(\hat{Y}; Y)}{\sqrt{H(\hat{Y})H(Y)}},$$

where $I(\hat{Y}; Y)$ denotes the mutual information between predicted labels $\hat{Y}$ and true labels Y, and $H(\cdot)$ represents entropy. NMI ranges from 0 to 1, with higher values indicating stronger correspondence between the predicted and ground-truth label distributions.

LLM	ACC	ARI	NMI
DeepSeek-V3	74.97	65.31	85.82
GPT-5-Mini	**76.53**	65.72	**85.99**
Kimi-K2	75.94	**65.79**	85.90
Grok-4-Fast	74.99	64.09	85.44
Gemma-3-27B	74.02	63.70	85.28
Qwen3-Next-80B-A3B	73.10	63.42	85.09

Table S1: Performance comparison across different LLMs on BANKING ($\rho = 0.25$).

D Impact of Different LLMs

To assess generalizability, we evaluate six models on BANKING with $\rho = 0.25$ (Table S1). While GPT-5-Mini and Kimi-K2 achieve top performance, open-source models like DeepSeek-V3, Gemma-3-27B, and Qwen3-Next-80B-A3B remain highly competitive. These open-source alternatives are particularly advantageous for practical deployment: they allow for local hosting to ensure data privacy, reduce latency, and lower operational costs during training. The robust results across models of varying scales underscore our framework's broad applicability and efficiency.

E Prompting Design Strategies

This appendix details our standard prompting implementation and compares it against three advanced strategy variants: Chain-of-Thought, In-Context Learning, and Self-Refine.

E.1 Our Standard Prompts

Our standard framework employs a multi-stage prompting pipeline within each training epoch to name, refine novel categories and select high-quality samples for training. Details are as follows:

E.1.1 Prompt 1: Novel Label Generation

This is the first step in the label generation process. After clustering identifies a new group of unlabeled data, this prompt is used to generate a single, concise label for it. The prompt emphasizes adherence to naming conventions and avoidance of duplication with any existing known labels.

Prompt 1: Novel Label Generation

System: You are an expert at categorizing text data and creating concise, meaningful labels.
User:
Context: Generalized Category Discovery / New Intent Recognition. Some intents are already known. You will propose ONE new intent label for the samples below.
Make the label:
- One intent only: the label must describe a single coherent

intent. - Handle outliers: base the label on the strongest common intent across samples. - Form: lowercase words joined by underscores; <=5 words; concise and label-like. - Style: express a single, specific action or outcome; reject broad expressions. - Do NOT duplicate or be synonymous with any KNOWN LABEL.
Output (strict JSON only; no code fences; no extra text): `{"label": "<label_text>"}`
KNOWN LABELS (for conflict/style reference): `{known_labels}`
SAMPLES (numbered):
`{numbered_samples}`

Variable

- `{known_labels}`: All known category labels from the training set.
- `{numbered_samples}`: Representative texts from a novel cluster, randomly sampled from the high-confidence subset.

E.1.2 Prompt 2: Rename Novel Label for Pure Novel Groups

Following initial label generation, this prompt addresses the scenario where hierarchical clustering identifies a "pure component" containing multiple, semantically similar novel clusters. It instructs the LLM to generate distinct labels for all groups in a single batch, enforcing pairwise distinctiveness to refine the initial names.

Prompt 2: Rename Novel Label for Pure Novel Groups

System: You answer only with minimal strict JSON.
User:
Context: You are given `{G}` groups of user queries. Each group needs ONE distinct intent label.
IMPORTANT: These `{G}` groups are semantically SIMILAR. You must add strong qualifiers (object/channel/phase/state) to ensure clear separation.
Task: Propose EXACTLY `{G}` labels.
Constraints:
- Pairwise distinctiveness: the `{G}` labels MUST be mutually exclusive. - GLOBAL-FORBIDDEN: do NOT duplicate or be synonymous. - Form: lowercase words joined by underscores; <=5 words. - Style: follow GLOBAL-FORBIDDEN naming style.
Output (strict JSON only; no code fences; no extra text): `{"labels": ["<label_1>", ..., "<label_{G}>"]}`
GLOBAL-FORBIDDEN: `{forbidden_global_str}`
GROUPS:
`{groups_joined}`

Variable

- `{G}`: The number of novel clusters within the pure component.
- `{forbidden_global_str}`: All known labels in the dataset.
- `{groups_joined}`: Representative samples for each of the 'G' novel clusters, formatted as numbered lists.

E.1.3 Prompt 3: Rename Novel Label for Mixed Groups

This prompt runs in parallel with Prompt 2 for a different scenario. It is designed for "mixed components" where novel clusters are semantically close to one or more known labels. It requires the LLM to generate new labels that are not only distinct from each other but also clearly distinguished from their semantically similar known neighbors.

Prompt 3: Rename Novel Label for Mixed Groups

System: You answer only with minimal strict JSON.
User:
Context: You are given `{G}` groups of user queries. Each group needs ONE distinct intent label.
IMPORTANT: These `{G}` groups are semantically SIMILAR to FORBIDDEN-LOCAL labels below. You must find subtle differences and add strong qualifiers to distinguish them.
Task: Propose EXACTLY `{G}` labels.
Constraints:
- Pairwise distinctiveness: the `{G}` labels MUST be mutually exclusive. - FORBIDDEN-LOCAL: do NOT duplicate or be synonymous; study sample texts to understand boundaries. - GLOBAL-FORBIDDEN: also avoid duplications.
Output (strict JSON only; no code fences; no extra text): `{"labels": ["<label_1>", ..., "<label_{G}>"]}`
FORBIDDEN-LOCAL (semantically SIMILAR; must distinguish): `{forbidden_local_str}`
`{known_joined}`
GLOBAL-FORBIDDEN: `{forbidden_global_str}`
GROUPS:
`{groups_joined}`

Variable

- `{G}`: The number of novel clusters within the mixed component.
- `{forbidden_local_str}`: Known labels from the same mixed component.
- `{known_joined}`: Sample texts for each 'forbidden_local_str' label, providing concrete boundary examples.
- `{forbidden_global_str}`: All known labels in the dataset.
- `{groups_joined}`: Representative samples for each of the 'G' novel clusters.

E.1.4 Prompt 4: Sample Selection

As the final step in the pipeline before training, this prompt is used to refine the data for both known and newly labeled categories. It instructs the LLM to select samples from a candidate pool that truly belong to a given 'target_label', effectively filtering out noise and ambiguous cases before they are used in contrastive learning.

Prompt 4: Sample Selection

System: You are an expert at judging whether a user query belongs to a given category label. You should make precise selections.
User:
You are given ONE known label and a list of numbered user queries.
Goal: select only the queries whose primary intent truly matches the known label.
Rules:
- One-intent focus: focus on the main intent of each query, not other aspects of the query. Semantic match only (not keyword/substring match). - Treat off-topic or rare mentions as outliers; EXCLUDE them. - Boundary with OTHER LABELS: if a query fits any OTHER LABEL better, EXCLUDE it. - Do not invent any new label. No explanation.
Note: quality over quantity; returning [] is acceptable !
KNOWN LABEL: `{target_label}`
OTHER LABELS (for boundary reference): `{known_labels_str}`
Output (strict JSON only; no code fences; no extra text): `{"selected_indices": [i1, i2, ...]}`
QUERIES (numbered):
`{numbered_queries}`

Variable

- `{target_label}`: A single category label (can be known or newly generated).
- `{known_labels_str}`: All other labels, used as a boundary reference to prevent overlap.
- `{numbered_queries}`: Candidate samples from the cluster associated with the ‘target_label‘.

E.2 Comparison with Advanced Prompting Strategies

We evaluated three advanced prompting strategies by modifying our core framework’s prompts. The key differences are highlighted below.

E.2.1 Chain-of-Thought (CoT)

The CoT strategy was applied to all four standard prompts. It modifies them by instructing the LLM to provide step-by-step reasoning before its final JSON output, making the decision-making process more transparent.

Example An instruction is added to the output format section of each prompt. For instance, the modification for Prompt 1 (Novel Label Generation) is shown below. The original prompt content is in light gray, and the added instruction is highlighted in green.

Make the label:
- One intent only: ...
- ...
- Do NOT duplicate or be synonymous with any KNOWN LABEL.

Please reason step by step.
Output format: First provide your step-by-step reasoning and analysis. Then provide your final answer in JSON format: `{"label": "<label_text>"}`

KNOWN LABELS (for conflict/style reference): ...
SAMPLES (numbered): ...

E.2.2 In-Context Learning (ICL)

The ICL strategy was applied to Prompt 1 (Novel Label Generation) and Prompt 4 (Sample Selection). For these tasks, injecting few-shot examples helps guide the LLM’s behavior. This strategy was not applied to the batch renaming prompts (Prompts 2 and 3) due to the high complexity and variability of their inputs, which makes crafting representative and non-confusing few-shot examples challenging.

Examples

1. For Prompt 1 (Novel Label Generation), 2-3 examples are inserted before the final ‘KNOWN LABELS‘ section. The added part is highlighted in blue.

... (rules section) ...
Output (strict JSON only; no code fences; no extra text):
`{"label": "<label_text>"}`

Example 1:
Samples:
`{example_samples_1}`
Label: `{example_label_1}`
Example 2:
Samples:
`{example_samples_2}`
Label: `{example_label_2}`

KNOWN LABELS (for conflict/style reference): ...

2. For Prompt 4 (Sample Selection), few-shot examples demonstrating correct selections are added.

... (rules section) ...
Output (strict JSON only; no code fences; no extra text):
`{"selected_indices": [...]}`

Example 1:
KNOWN LABEL: `{example_label_1}`
SAMPLES:
`{example_samples_1}`
Output: `{"selected_indices": [1, 3]}`
Example 2:
KNOWN LABEL: `{example_label_2}`
SAMPLES:
`{example_samples_2}`
Output: `{"selected_indices": [2, 4, 5]}`

KNOWN LABEL: `{target_label}` ...

E.2.3 Self-Refine

The Self-Refine strategy was applied to all four standard prompts. It transforms each LLM call into a two-round process: an initial generation followed by a critique-and-refine step.

Example This is a procedural change. For any given task, such as Prompt 1 (Novel Label Generation), the process is as follows:

1. **Round 1**: The standard prompt is sent to the LLM, producing an 'initial_response'.
2. **Round 2**: A new prompt is constructed, containing the original task, the 'initial_response', and a set of review criteria. The LLM is asked to critique its own work and provide a 'refined_answer'. The final output is parsed from this second response.

The review-and-refine prompt for label generation is shown below, with the review instructions highlighted in orange.

Original Task: `{initial_prompt}`
Previous Answer: `{initial_response}`

Please carefully review your generated label: 1. Does this label accurately summarize the common theme? 2. Is the label concise and easy to understand? 3. Does the label duplicate any known labels?
Output format: Feedback: [Your analysis and feedback] Refined Answer: `{"label": "improved label"}`