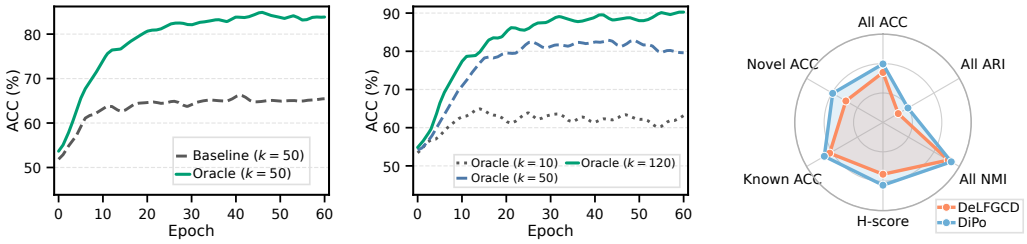
BEYOND LOCAL POSITIVES: LLM-VERIFIED DIVERSE POSITIVE MINING FOR GENERALIZED CATEGORY DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalized Category Discovery (GCD) aims to recognize known categories and discover novel ones from unlabeled data given limited labeled examples of known categories. A central difficulty is constructing effective training signals for unlabeled instances, especially for novel categories. Recent LLM-assisted methods primarily improve the semantic reliability of these signals, but under-emphasize positive diversity: the coverage of same-class pairs that are less similar under the current representation. An oracle study with ground-truth filtering confirms this gap: even when all positives are same-class pairs, restricting them to highly similar candidates still limits discovery. However, expanding the candidate pool also introduces false-positive relations when ground-truth labels are unavailable. To address this trade-off, we propose **DiPo** (**Di**verse **Po**sitive mining), a budgeted two-path framework for neighborhood contrastive learning. For a fraction of anchors, DiPo expands the candidate pool and uses LLM same-or-different judgments to retain semantically valid, less-similar positives; for the remaining anchors, it uses the nearest cluster-consistent mutual neighbor as a reliable local positive. Extensive experiments show that DiPo attains the best average performance among strong LLM-assisted baselines and is particularly effective on novel-class discovery. Code is available at DiPo.



(a) Oracle vs. baseline. (b) Oracle neighborhood size. (c) DiPo vs. DeLFGCD.

Figure 1: **Why diverse positives matter** (BANKING, KCR=25%). (a) Removing false-positive neighbors (oracle) improves ACC over NCL. (b) When all positives are constrained to be same-class, a small neighborhood (K=10) yields limited gains, whereas larger pools (K=50 and K=120) continue to improve ACC. (c) DiPo outperforms DeLFGCD on overall ACC/ARI/NMI, H-score (known/novel harmonic mean), and known-/novel-class ACC.

1 INTRODUCTION

In many real-world applications, the category space is not fixed: new user requests, discussion topics, and business categories continually emerge, while only a subset of categories can be annotated. Generalized Category Discovery (GCD) (Vaze et al., 2022; Cao et al., 2022) addresses this setting: a model must recognize known categories and discover novel ones from partially labeled data. The central difficulty is constructing training signals for unlabeled instances, especially for novel categories, for which no labeled exemplars are available (Han et al., 2019; Fini et al., 2021).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A common strategy is to supplement the limited labeled data with pseudo-supervision derived from the unlabeled instances, such as contrastive relations, clustering assignments, or neighborhood structures (Han et al., 2019; Lin et al., 2020; Zhang et al., 2021; Vaze et al., 2022; Zhang et al., 2022; Wen et al., 2023; Caron et al., 2018; Van Gansbeke et al., 2020). However, when these signals are inferred from the evolving representation, they can be unreliable. Semantically inconsistent relations may be introduced as positive supervision and repeatedly reinforced during representation refinement, causing incorrect structures to persist, particularly for novel categories. Recent GCD methods have explored large language models (LLMs) (Brown et al., 2020) as an external semantic signal to improve the reliability of this representation-derived pseudo-supervision.

However, reliability alone does not make a supervision relation sufficient for effective discovery. We investigate this issue through controlled oracle experiments in GCD. First, replacing representation-derived relations with ground-truth-consistent relations improves discovery performance, confirming the importance of semantic reliability (Fig. 1a). Yet, even when all positive relations are guaranteed to be semantically correct, restricting them to a narrow neighborhood remains suboptimal: expanding the candidate neighborhood can yield further gains (Fig. 1b). These gains cannot be attributed to a difference in purity, since the oracle keeps the retained relations ground-truth consistent at every neighborhood size. Instead, broader neighborhoods expose same-class relations that are inaccessible when supervision is confined to a small neighborhood in the current representation. Thus, positive supervision in GCD faces a tension between reliability and diversity: expanding the candidate pool provides more diverse positive relations, but also introduces more potentially incorrect relations when semantic labels are unavailable.

We propose **DiPo** (**Di**verse **Po**sitive mining) to address this tension with selective LLM supervision. DiPo treats the learned neighborhood as a candidate pool rather than as the positive set. For a fraction of anchors, it expands the neighborhood so that more diverse same-class pairs enter the candidate pool, and applies LLM same-or-different judgments to that pool so those pairs can be kept as positives. For the remaining anchors, it selects the nearest cluster-consistent mutual neighbor, so the positive stays reliable without an LLM judgment. DiPo improves overall discovery and narrows the performance gap between known and novel categories on standard text GCD benchmarks. Our contributions are threefold:

(1) We show through a controlled oracle study that positive supervision in text GCD requires both semantic reliability and diversity: even ground-truth-consistent positives remain weak when drawn only from a narrow neighborhood.

(2) We propose DiPo, which decouples candidate exploration from positive selection: retrieved neighborhoods serve as candidate pools, and selective semantic verification identifies reliable positives beyond the local neighborhood.

(3) On BANKING, CLINC, and StackOverflow, DiPo attains the best average performance among strong LLM-assisted baselines, with larger gains on novel categories that narrow the gap between known and novel categories.

2 Rethinking Positive Construction in GCD

Problem Formulation. Let $\mathcal{C} = \mathcal{C}_k \cup \mathcal{C}_u$ with $\mathcal{C}_k \cap \mathcal{C}_u = \emptyset$ and $C = |\mathcal{C}|$, where $\mathcal{C}_k$ and $\mathcal{C}_u$ denote the known and novel categories. The known category ratio (KCR) is $|\mathcal{C}_k|/C$, the fraction of categories treated as known. The training data consist of labeled instances $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{N_l}$ with $y_i \in \mathcal{C}_k$, and unlabeled instances $\mathcal{D}_u = \{x_i\}_{i=1}^{N_u}$, whose labels may belong to either $\mathcal{C}_k$ or $\mathcal{C}_u$. Only labels from $\mathcal{C}_k$ are observed during training, while no class names or exemplars from $\mathcal{C}_u$ are provided (Vaze et al., 2022). The goal is to learn an encoder f_θ that maps each instance to an embedding $\mathbf{z}_i = f_\theta(x_i)$ such that instances from all C categories can be grouped. At test time, the learned embeddings are clustered into C groups (Appendix A).

2.1 Neighborhood-based Contrastive Learning

Our baseline, Neighborhood Contrastive Learning (NCL) (Zhong et al., 2021), uses neighbors in the current representation as positives. Given an anchor instance x_i with L2-normalized embedding $\mathbf{z}_i$,

we retrieve its K nearest neighbors and denote them by

$$\mathcal{N}_K(x_i) = \{x_{i,1}, \ldots, x_{i,K}\}.$$

One neighbor $x_i^+ \in \mathcal{N}_K(x_i)$ is drawn uniformly at random and used as a positive for the anchor, in addition to same-label pairs when both instances are labeled. The anchor and this neighbor are encoded separately and used in a supervised contrastive objective (Khosla et al., 2020; Chen et al., 2020; van den Oord et al., 2018; Gao et al., 2021):

$$\mathcal{L}_{\text{SupCon}} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \log \frac{\exp(\mathbf{z}_i^\top \mathbf{z}_p / \tau)}{\sum_{a \in \mathcal{B} \setminus \{i\}} \exp(\mathbf{z}_i^\top \mathbf{z}_a / \tau)},$$

where $\mathcal{B}$ is a mini-batch of encoded views, $\mathcal{P}(i)$ denotes the positive views of i in $\mathcal{B}$, and τ is the temperature. For every K, this draw returns one neighborhood positive. The neighborhood size K changes only the candidate set.

2.2 Semantic Reliability of Neighborhood Relations

The neighbor drawn above provides pseudo-supervision from the current representation, so its reliability depends on representation quality. In particular, the neighbor may be semantically inconsistent with the anchor when similarity does not reflect class membership, especially for novel categories without labeled exemplars.

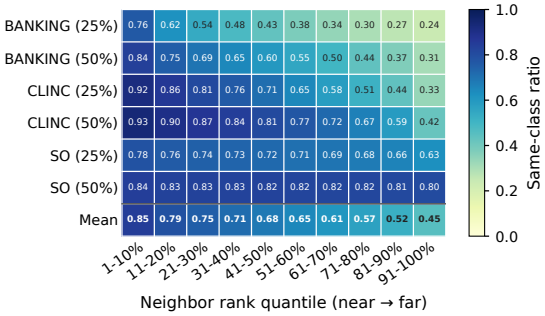


Figure 2: **Neighborhood purity by rank quantile.** Same-class neighbor ratio in successive Top-K rank bins across datasets and KCR settings. Mean purity drops from 0.85 (nearest 10%) to 0.45 (farthest 10%).

To quantify the reliability of representation-derived neighborhoods, we measure neighborhood purity, defined as the proportion of same-class neighbors, across successive rank intervals within neighborhoods from the pretrained encoder, for BANKING, CLINC, and StackOverflow at KCR=25% and 50%. As shown in Fig. 2, purity falls steadily with neighbor rank in every setting, and even the nearest band still admits cross-class relations, so representation similarity is an imperfect proxy for class membership. The degradation is sharper for novel-class anchors than for known-class anchors (Appendix D.1).

To isolate the effect of semantic reliability, we construct an oracle variant that uses the same Top-K as the baseline and retains only same-class neighbors according to the ground-truth labels. As shown in Fig. 1a for BANKING and Fig. 3 for CLINC and StackOverflow, this ground-truth filtering yields a large ACC gain over that baseline. The gap shows that cross-class neighbors harm discovery, confirming that semantic reliability matters for neighborhood contrastive learning.

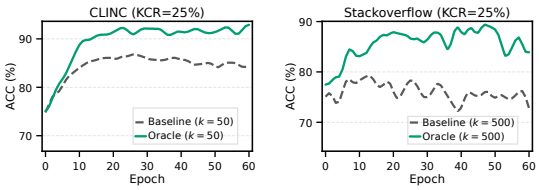
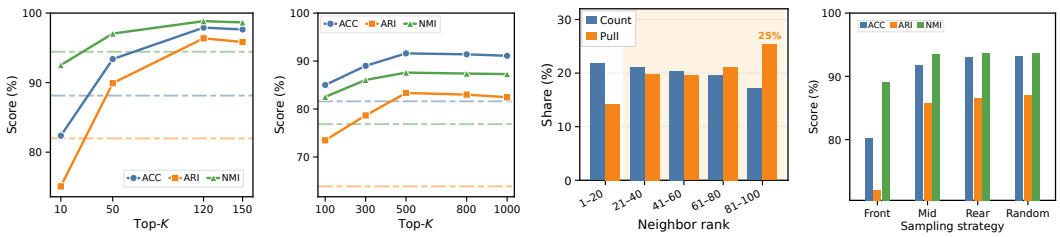


Figure 3: **Oracle vs. baseline training dynamics.** Removing semantically inconsistent neighbors improves ACC on CLINC (left) and StackOverflow (right).

2.3 Beyond a Narrow Neighborhood

A natural next question is whether any same-class neighbor is equally useful once purity is perfect, or whether neighborhood *size* still matters. We therefore keep the ground-truth filter and vary only K. As in Sec. 2.1, each anchor still contributes one neighborhood positive, so changing K changes only the candidate set from which that positive is drawn.

At KCR=25%, oracle ACC/ARI/NMI rise with K and then level off on BANKING (Fig. 1b), CLINC (Fig. 4a), and StackOverflow (Fig. 4b). On all three datasets, a small pool can leave the oracle below the NCL baseline: restricting same-class positives to the nearest ranks can hurt performance more than admitting the false positives that NCL allows. Every oracle neighbor is from the



(a) CLINC. (b) StackOverflow. (c) Count vs. pull. (d) Sampling strategy.

Figure 4: **Oracle neighborhood size, pull, and sampling** (KCR=25%). (a)–(b) Oracle ACC/ARI/NMI vs. Top-K on CLINC and StackOverflow (dashed: NCL). (c) Count vs. InfoNCE pull by rank bin on BANKING (K=100). (d) Front/Mid/Rear/Random sampling of M candidates on BANKING.

same class as its anchor, so the gains from larger K are not from higher purity. What K controls is access to same-class neighbors at larger ranks. Under K=100 on BANKING, selected oracle positives span many rank bins, but farther neighbors dominate the InfoNCE attractive weight (Fig. 4c). Appendix E.6 defines this weight as *pull* (van den Oord et al., 2018): ranks 81–100 contribute more than ranks 1–20, despite fewer selections.

To test where useful same-class relations lie under a fixed candidate budget M, we compare Front, Mid, Rear, and Random sampling of M candidates from a fixed Top-K, with ground-truth filtering applied only to the sampled candidates (Fig. 4d). Front draws from the nearest third of Top-K, Mid and Rear from progressively less local thirds, and Random uniformly from the full neighborhood. Front underperforms Mid, Rear, and Random; the latter three remain close. The same ordering holds on CLINC and StackOverflow (Appendix D.2). Because each strategy still contributes one neighborhood positive per anchor, the gap is not about how many relations enter training, but about where the candidates come from.

Sec. 2.2 showed that false neighbors hurt; the experiments above show that even perfect same-class neighbors help more when they are not restricted to the nearest ranks. We therefore treat the retrieved neighborhood as a *candidate pool* rather than a fixed positive set, and take *positive diversity* to mean drawing same-class neighbors from a wider range of ranks under the current embedding.

3 Methodology

3.1 Overview

Building on Sec. 2, DiPo addresses the tension between reliability and diversity by treating the retrieved neighborhood as a *candidate pool* and decoupling candidate exploration from positive selection. Under a limited LLM budget, for a random fraction ρ of anchors, DiPo verifies candidates from an expanded pool to recover reliable same-class positives beyond the local neighborhood. The remaining anchors use the nearest cluster-consistent mutual neighbor, so positives stay reliable without an LLM judgment. Both paths feed the same supervised contrastive objective; only the positive source changes (Fig. 5).

3.2 LLM-Assisted Diverse Positive Mining

Random anchor selection. We form the LLM-path anchor set $\mathcal{A}$ by uniform sampling over the $N = N_l + N_u$ training instances, with $|\mathcal{A}| = \lfloor \rho N \rfloor$. For each anchor $x_i \in \mathcal{A}$, candidates are explored and verified as follows. Appendix E.4 compares this rule with entropy-based alternatives under the same query budget.

Expanded candidate exploration. The candidate pool is the K-nearest-neighbor set $\mathcal{N}_K(x_i)$ defined in Sec. 2.1; a larger K extends it beyond a narrow neighborhood for semantic verification and subsequent positive construction. From this neighborhood, we sample M candidates according to a sampling policy π:

$$\mathcal{S}_i = \mathrm{Sample}_M(\mathcal{N}_K(x_i); \pi).$$

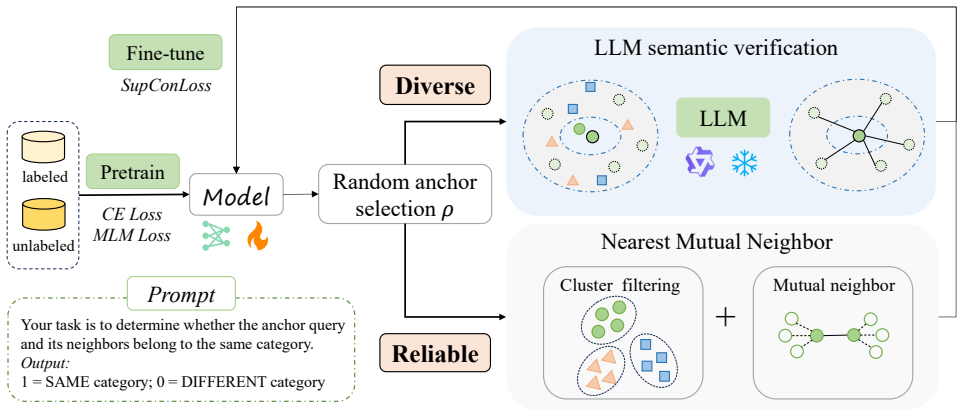


Figure 5: **Overview of DiPo.** A random fraction of anchors take the LLM path: expand the neighborhood, sample M candidates, and retain same-class relations via LLM verification. The remaining anchors take the unqueried path: keep the nearest cluster-consistent mutual neighbor. Both paths feed the same supervised contrastive objective.

Here, π specifies how the M candidates are drawn from the ranked neighborhood for verification. By default we sample uniformly at random from the full $\mathcal{N}_K(x_i)$; for ablation we also restrict sampling to the front or the rear of this ranking.

The neighborhood size K determines the breadth of candidate relations available for exploration, whereas M controls the number of candidates submitted for semantic verification. Verifying all K neighbors with the LLM is impractical. Sampling M candidates from this larger pool for verification is sufficient to achieve strong performance.

LLM relational verification. We use an LLM to verify the semantic consistency between each anchor and its sampled candidates. Given an anchor x_i and candidate set $\mathcal{S}_i$, we formulate the task as a binary relational judgment:

$$g(x_i, x_j) \in \{0, 1\},$$

where $g(x_i, x_j) = 1$ indicates that the LLM judges x_i and x_j to belong to the same semantic category. The LLM is not asked to rank candidates or assign category labels; instead, it directly determines whether each candidate is semantically consistent with the anchor, based on their underlying meaning rather than surface-form similarity.

The verified candidate set is

$$\mathcal{C}_i^{\text{LLM}} = \{x_j \in \mathcal{S}_i \mid g(x_i, x_j) = 1\}.$$

Only candidates judged to be semantically consistent with the anchor are retained for subsequent positive construction. If $\mathcal{C}_i^{\text{LLM}}$ is empty, we sample one neighbor uniformly from $\mathcal{N}_K(x_i)$; empty-pool rates are reported in Appendix F.4. A simplified example of the verification interface is shown below:

LLM semantic verification

Anchor: x_a
Candidates: x_1, x_2, x_3, x_4
Task: Identify candidates that share the same underlying semantic category as the anchor. Judge by the underlying meaning rather than surface-form similarity.
Output: `"same": [1,3]`

The full prompt is in Appendix F.6.

Efficient LLM supervision. Because LLM verification is substantially more expensive than model-based neighborhood construction, we perform semantic annotation only at a small number of early training stages rather than throughout training. Each stage samples a new anchor set $\mathcal{A}$ with $|\mathcal{A}| = \lfloor \rho N \rfloor$, and the verified relations from all stages are accumulated in one cache. This cache is reused in subsequent training stages, avoiding redundant LLM queries (Appendix F.4).

Table 1: Main comparison (ACC/ARI/NMI, %). Best in **bold**, second best underlined. Avg. is the mean of the nine scores per KCR. DiPo reports the mean over three seeds; seed-wise standard deviations are in Appendix E.9. KCR=5%/10% are in Appendix E.3; some baseline numbers are from (Zou et al., 2026).

KCR	Method	CLINC			BANKING			StackOverflow			Avg.
		ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	
25%	DeepAligned (Zhang et al., 2021)	74.07	64.63	88.97	49.08	37.62	70.50	54.50	37.96	50.86	58.69
	MTP-CLNN (Zhang et al., 2022)	83.26	76.20	93.17	65.06	52.91	80.04	74.70	54.80	73.35	72.61
	GCD (Vaze et al., 2022)	82.31	75.45	92.94	69.64	58.30	82.17	81.60	65.90	78.76	76.34
	ProbNID (Zhou et al., 2023)	71.56	63.25	89.21	55.75	44.25	74.37	54.10	38.10	53.70	60.48
	USNID (Zhang et al., 2024)	83.12	77.95	94.17	65.85	56.53	81.94	75.76	65.45	74.91	75.08
	SimGCD (Wen et al., 2023)	84.44	77.53	93.44	69.55	57.86	81.71	79.80	65.19	79.09	76.51
	CsePL (Liang & Liao, 2023)	86.16	79.65	94.07	71.06	60.36	83.22	79.47	64.92	74.88	77.09
	ALUP (Liang et al., 2024b)	88.40	82.44	94.84	74.61	62.64	84.06	82.20	64.54	76.58	78.92
	Loop (An et al., 2024)	86.58	80.67	94.38	71.40	60.95	83.37	82.20	66.29	79.10	78.33
	DeLFGCD (Zou et al., 2026)	91.51	87.07	96.27	76.98	66.00	85.62	84.10	71.01	80.90	82.16
	DiPo (ours)	**93.57**	**89.31**	**96.75**	**79.02**	**69.41**	**86.92**	**87.00**	**74.34**	**81.90**	**84.25**
50%	DeepAligned (Zhang et al., 2021)	80.70	72.56	91.59	59.38	47.95	76.67	74.52	57.62	68.28	69.92
	MTP-CLNN (Zhang et al., 2022)	86.18	80.17	94.30	70.97	60.17	83.42	80.36	62.24	76.66	77.16
	GCD (Vaze et al., 2022)	86.53	81.06	94.60	74.42	63.83	84.84	85.60	72.20	80.12	80.36
	ProbNID (Zhou et al., 2023)	82.62	75.27	92.72	63.02	50.42	77.95	73.20	62.46	74.54	72.47
	USNID (Zhang et al., 2024)	87.22	82.87	95.45	73.27	63.77	85.05	82.06	71.63	78.77	80.01
	SimGCD (Wen et al., 2023)	87.24	81.65	94.83	74.42	64.17	85.08	82.00	70.67	80.44	80.06
	CsePL (Liang & Liao, 2023)	88.66	83.14	95.09	76.94	66.66	85.65	85.68	71.99	80.28	81.57
	ALUP (Liang et al., 2024b)	90.53	84.84	95.97	79.45	68.78	86.79	86.70	73.85	81.45	83.15
	Loop (An et al., 2024)	90.98	85.15	95.59	75.06	65.70	85.43	85.90	72.45	80.56	81.87
	DeLFGCD (Zou et al., 2026)	94.53	90.79	97.12	80.26	70.40	87.65	**89.40**	**78.92**	**85.04**	86.01
	DiPo (ours)	**95.58**	**92.26**	**97.60**	**83.00**	**73.36**	**88.68**	88.85	78.11	83.92	**86.82**

3.3 Reliable Positive Mining for Unqueried Anchors

For unqueried anchors $x_i \notin \mathcal{A}$, let $\hat{y}_i$ be the current cluster pseudo-label of x_i. We restrict $\mathcal{N}_K(x_i)$ to same-cluster neighbors in rank order and set x_i^{mut} to the nearest mutual neighbor in this pool, i.e., the first cluster-consistent mutual neighbor (Brito et al., 1997; Van Gansbeke et al., 2020). If no mutual neighbor exists, we use the nearest same-cluster neighbor.

3.4 Contrastive Objective

We use the same supervised contrastive objective introduced in Sec. 2.1. The only modification is how the positive relation is selected. For LLM-path anchors we uniformly sample one verified candidate; for unqueried anchors we use the first cluster-consistent mutual neighbor:

$$x_i^+ = \begin{cases} \text{Sample}(\mathcal{C}_i^{\text{LLM}}), & x_i \in \mathcal{A}, \\ x_i^{\text{mut}}, & \text{otherwise.} \end{cases}$$

We then encode the anchor and its selected neighbor as a positive pair and optimize them with the same objective as NCL.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate on three standard text GCD benchmarks, BANKING (Casanueva et al., 2020), CLINC (Larson et al., 2019), and StackOverflow (Xu et al., 2015), under KCR=25% and 50%. Following prior work (Vaze et al., 2022; Zhang et al., 2022; Zou et al., 2026), within each known category we label 10% of the training instances and use the same train/validation/test splits. We report clustering accuracy (ACC), adjusted Rand index (ARI), and normalized mutual information (NMI); dataset statistics and metric definitions are in Appendix A.

Baselines. Baselines include conventional GCD methods without LLM feedback (GCD (Vaze et al., 2022), SimGCD (Wen et al., 2023), DeepAligned (Zhang et al., 2021), MTP-CLNN (Zhang et al., 2022), ProbNID (Zhou et al., 2023), USNID (Zhang et al., 2024), and CsePL (Liang & Liao, 2023)) and recent LLM-assisted methods (ALUP (Liang et al., 2024b), Loop (An et al., 2024), and DeLFGCD (Zou et al., 2026)).

Implementation Details. We use BERT (Devlin et al., 2019) as the backbone and Qwen3.5-Flash as the default LLM annotator. Following prior text GCD setups (Zhang et al., 2022; An et al., 2024; Zou et al., 2026), we first pretrain with cross-entropy on labeled known-class samples and masked language modeling on all training instances, then run discovery-stage contrastive learning. Key defaults include annotation ratio ρ (0.2 on BANKING; 0.1 on CLINC/StackOverflow), candidate size M=30, Top-K of 100/120/500 on BANKING/CLINC/StackOverflow, three early LLM rounds, and cluster-consistent mutual neighbors for unqueried anchors; full settings are in Appendix B.

4.2 Main Results

In Table 1, DiPo attains the highest nine-metric average at both KCR=25% and 50%, ahead of the strongest LLM-assisted baseline DeLFGCD. On CLINC and BANKING it improves every reported ACC/ARI/NMI entry over DeLFGCD; the gains are larger on BANKING, consistent with its fine-grained intents and less reliable local neighborhoods (Sec. 2.2). On StackOverflow at KCR=25%, DiPo remains best on all three metrics. At KCR=50% it is slightly behind DeLFGCD on StackOverflow alone (e.g., 88.85 vs. 89.40 ACC), yet still ranks second on that dataset and keeps the best overall average. The smaller StackOverflow headroom matches later ablations: verify+expanded yields a smaller ACC gain there than on CLINC or BANKING (Table 2). StackOverflow uses coarser topic categories, so less-local positives have less room to help once local neighborhoods are already reliable.

4.3 Ablation and Analysis

Table 2 studies the trade-off between reliability and diversity introduced in Sec. 2 at KCR=25%. We cross *Verify* (LLM relational verification on vs. off) with *Pool* (local vs. expanded Top-K). Without verification the setup reduces to NCL; with verification it follows DiPo's LLM path over the chosen pool, while unqueried anchors keep the default mutual neighbor.

Table 2: Reliability×diversity cross ablation at KCR=25% (%). Verify: LLM verification vs. none (NCL). Pool: local / expanded Top-K (50/100 on BANKING, 50/120 on CLINC, 300/500 on StackOverflow). ΔACC vs. NCL+local on the same dataset. Bold: verify + expanded (default DiPo).

Dataset	Verify	Pool	ACC	ARI	NMI	ΔACC
CLINC	No	local	87.16	81.35	94.51	–
	No	expanded	88.04	81.10	94.43	+0.9
	Yes	local	86.80	80.41	94.12	−0.4
	Yes	expanded	**93.69**	**89.57**	**96.80**	**+6.5**
BANKING	No	local	68.93	58.63	82.35	–
	No	expanded	68.28	56.02	81.21	−0.7
	Yes	local	73.67	63.37	84.90	+4.7
	Yes	expanded	**79.38**	**69.53**	**87.29**	**+10.5**
StackOverflow	No	local	83.40	66.94	78.62	–
	No	expanded	82.20	67.02	77.79	−1.2
	Yes	local	83.80	69.62	80.65	+0.4
	Yes	expanded	**87.60**	**75.58**	**82.30**	**+4.2**

Reliability and diversity are complementary. Without verification, expanding the pool does not help and can hurt (BANKING −0.7, StackOverflow −1.2 ACC), matching the purity drop in Sec. 2.2. Verification with a local pool helps less than with an expanded pool: on CLINC, verify+local even falls slightly below NCL+local, since a narrow pool yields few same-class candidates beyond the nearest ranks and verification can discard useful near neighbors. Verify+expanded is best on every dataset. The step from verify+local to verify+expanded yields the largest additional gains (CLINC +6.9, BANKING +5.7, StackOverflow +3.8 ACC).

Does the LLM path help? In the "w/o LLM path" setting of Table 3, every anchor uses the nearest cluster-consistent mutual neighbor, with no LLM path. Scores drop on all three datasets: mutual retrieval gives reliable local positives, but this signal alone is not enough for strong discovery.

How should unqueried anchors be supervised? With the LLM path fixed at default, we vary only the unqueried positive: Mutual† (nearest cluster-consistent mutual neighbor), Random,

Table 3: Path and sampling ablation at KCR=25% (ACC/ARI/NMI, %). †Default DiPo; best per block and column in **bold**.

Component	Setting	CLINC			BANKING			StackOverflow		
		ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI
w/o LLM path	Mutual	87.64	80.92	94.45	68.67	56.98	81.49	84.80	70.19	79.07
Unqueried path	Mutual†	**93.69**	**89.57**	**96.80**	**79.38**	**69.53**	**87.29**	**87.60**	**75.58**	**82.30**
	Random	92.04	86.75	96.05	74.94	64.68	85.06	82.60	66.02	76.94
	Front	92.67	88.03	96.37	78.73	68.47	86.43	83.30	66.80	78.15
Candidate sampling	Random†	**93.69**	**89.57**	**96.80**	79.38	**69.53**	**87.29**	**87.60**	**75.58**	**82.30**
	Front	91.73	86.82	96.22	74.29	63.92	85.02	85.40	71.29	79.73
	Rear	90.93	86.11	95.72	**79.81**	69.45	86.70	86.30	73.22	80.74

or Front (same-cluster without mutual retrieval; Front draws from the closest third of ranks). Mutual is strongest; Random and Front underperform, most clearly on BANKING and StackOverflow. Both alternatives remove the mutual-neighbor requirement and keep only a same-cluster neighbor, and Random further widens the pool, admitting more noise. The Mutual advantage is larger when known categories are scarce (Appendix E.5).

Where should LLM candidates be explored? With unqueried Mutual fixed, we draw M LLM-path candidates from Top-K by Random† (uniform), Front (closest third), or Rear (farthest third). Front is consistently weaker than Random, matching Sec. 2.3: useful same-class candidates are not confined to the nearest ranks. Rear stays close to Random on BANKING and StackOverflow. Default DiPo pairs random sampling of M candidates on the LLM path with mutual neighbors on the unqueried path.

Less-local positives come from the LLM path. Fig. 6 shows positive ranks under default DiPo on CLINC at KCR=25%. Unqueried positives concentrate at rank 1, so the mutual rule stays local. LLM-path positives keep a long rank tail: nearly all selected neighbors beyond rank 20 come from this path. These less-local pairs are only 18.3% of selected positives on CLINC, yet they contribute 60.4% of the InfoNCE pull (Appendix E.6 reports all three datasets). The two paths play complementary roles. Mutual neighbors supply reliable local positives, while LLM verification over randomly sampled candidates recovers the less-local same-class pairs that carry most of the attractive weight. This matches the Front gap in Table 3 and the diversity claim in Sec. 2.3.

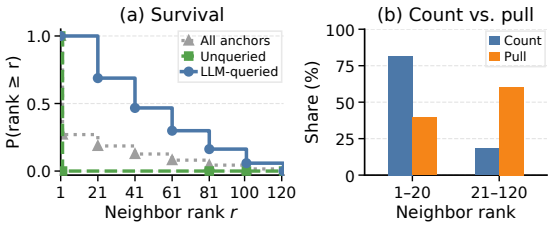


Figure 6: **Where DiPo positives come from** (CLINC, KCR=25%). (a) Selected-positive rank survival for all, LLM-path, and unqueried anchors. (b) Count vs. InfoNCE pull share. Unqueried positives concentrate at rank 1; the rank>20 tail is from the LLM path and carries most of the pull.

Known versus novel discovery. Table 4 reports H-score and known-/novel-class ACC on BANKING at KCR=25% and 50% under Loop's protocol (An et al., 2024) with default DiPo. DiPo attains the best H-score and both subset accuracies at KCR=25%, and leads on every reported entry at 50%. Under DiPo, novel-class ACC stays close to known-class ACC; Loop and DeLFGCD show larger gaps between known and novel categories at KCR=25% (7.74 and 6.35 ACC). This pattern matches Sec. 2.2: novel-class neighborhoods are less pure, so verified less-local positives matter most where labeled exemplars are absent. Full known/novel breakdowns and method-level BANKING curves are in Appendix E.1 and Appendix E.2; additional KCR settings relative to NCL are in Appendix E.3.

Table 4: Known vs. novel discovery on BANKING (%). Best in each column in **bold**. "–": DeLFGCD unavailable at KCR=50% under matched reporting.

Method	KCR=25%			KCR=50%		
	H-score	Known	Novel	H-score	Known	Novel
GCD (Vaze et al., 2022)	44.31	63.68	33.97	47.48	71.64	35.51
MTP-CLNN (Zhang et al., 2022)	51.18	67.28	41.30	56.71	76.43	45.08
ALUP (Liang et al., 2024b)	73.18	75.39	71.10	77.69	79.87	75.62
Loop (An et al., 2024)	72.80	76.88	69.14	75.48	82.54	69.53
DeLFGCD (Zou et al., 2026)	77.62	80.92	74.57	–	–	–
DiPo (ours)	**81.35**	**83.03**	**79.74**	**84.61**	**85.13**	**84.10**

Budget and candidate size. Fig. 7 varies the two knobs that set the LLM budget: the fraction ρ of anchors queried per early round, and the number M of candidates verified per queried anchor. On BANKING (left), ACC, ARI, and NMI rise as ρ grows from 0.05 to 0.30, with most of the gain secured by our default ρ=0.20 and smaller returns beyond it. We therefore keep ρ=0.20 on BANKING, and use ρ=0.10 on CLINC and StackOverflow. On CLINC (right), M has only a mild effect: scores are already strong at M=10 and improve gradually through M=50. We fix M=30 across datasets as a practical balance of cost and accuracy.

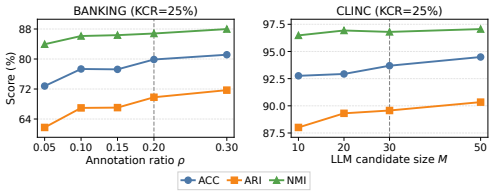


Figure 7: **Sensitivity to LLM-budget knobs.** Left: annotation ratio ρ on BANKING; right: candidate size M on CLINC. Dashed lines mark defaults (0.20 / M=30).

LLM annotation quality. Table 5 reports pairwise same-or-different predictions scored against ground-truth category equality with same-class as the positive label. CLINC and StackOverflow show high precision and recall; BANKING is harder, with precision 0.74 and recall 0.81. The annotator is imperfect on this fine-grained benchmark, yet verify+expanded still yields the largest gains on BANKING in Table 2. This supports using the LLM for relational verification over an expanded pool even when its pairwise labels are imperfect. Full KCR breakdowns are in Appendix F.

Table 5: LLM relation-label quality (Qwen3.5-Flash) at KCR=25%.

Dataset	Prec.	Rec.	F1	Acc.
BANKING	0.74	0.81	0.77	0.78
CLINC	0.96	0.81	0.88	0.85
StackOverflow	0.95	0.90	0.92	0.90

5 Related Work

Generalized Category Discovery. GCD (Vaze et al., 2022; Cao et al., 2022) asks a model to recognize known categories and discover novel ones from partially labeled data. Supervision for unlabeled instances is typically derived from clustering, pairwise relations, or the learned representation (Han et al., 2019; Fini et al., 2021; Lin et al., 2020; Zhang et al., 2021; Wen et al., 2023; Zhang et al., 2024). Text GCD often adopts neighborhood contrastive learning, in which neighbors in the current embedding serve as positives (Zhong et al., 2021; Zhang et al., 2022). DiPo follows this NCL formulation: it treats the retrieved neighborhood as a candidate pool and changes how the positive is selected.

LLMs for Category Discovery. Recent methods use LLMs (Brown et al., 2020) to supply semantic supervision beyond what the current embedding can reliably infer. The supervision is typically a category cue, such as a pseudo-label or an alignment to a cluster description (Liang et al., 2024b; Fan et al., 2025; Zou et al., 2026), or a relational cue that selects a contrastive neighbor (An et al., 2024; Zou et al., 2026; Zhang et al., 2023; Fan et al., 2024), usually collected on uncertain or locally inconsistent anchors. The aim is higher reliability of pseudo-supervision, but candidates stay nearby or in locally consistent clusters, so even correct judgments rarely recover less-local same-class pairs. DiPo keeps relational supervision, but targets diversity as well as reliability: the neighborhood is treated as a candidate pool, and selective verification is used to recover reliable same-class positives beyond the local neighborhood rather than only to correct nearby relations. Enrichment-based variants and concrete query formats are compared in Appendix C.

6 Conclusion

Positive supervision in text GCD requires both semantic reliability and diversity. False-positive neighbors harm discovery, yet even ground-truth-consistent positives remain weak when they are drawn only from a narrow neighborhood. DiPo treats the neighborhood as a candidate pool, verifies a fraction of anchors with an LLM over an expanded neighborhood, and uses cluster-consistent mutual neighbors for the rest. On BANKING, CLINC, and StackOverflow, it attains the best average performance among strong LLM-assisted baselines, with larger gains on novel categories that narrow the gap between known and novel categories.

AI use statement

In this work, we used generative AI tools to interpret experimental results, to give feedback on experimental comparisons, to draft and edit the manuscript, and to write figure-plotting code. Parts of the evaluation code and code comments were drafted with Cursor in Auto mode. We have not used generative AI tools to generate synthetic data, develop theoretical models, formulate or prove mathematical claims, translate text, or clean and reformat datasets; qualitative or thematic analysis is not applicable. We have reviewed all AI-assisted work and checked numerical claims and figure values against experimental outputs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility statement

Dataset statistics, metric definitions, and splits are in Appendix A; core hyperparameters are in Appendix B; seed-wise standard deviations for Table 1 are in Appendix E.9; and LLM prompts are in Appendix F.6. Code is available at DiPo.

References

Wenbin An, Wenkai Shi, Feng Tian, Haonan Lin, QianYing Wang, Yaqiang Wu, Mingxiang Cai, Luyan Wang, Yan Chen, Haiping Zhu, and Ping Chen. Generalized category discovery with large language models in the loop. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8653–8665, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.512. URL https://aclanthology.org/2024.findings-acl.512/.

M. R. Brito, E. L. Chávez, A. J. Quiroz, and J. E. Yukich. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, 35(1):33–42, 1997. doi: 10.1016/S0167-7152(96)00213-1.

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. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, Virtual, 2020. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=O-r8LOR-CCA.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Computer Vision – ECCV 2018*, pp. 139–156, Munich, Germany, 2018. Springer. doi: 10.1007/978-3-030-01264-9_9.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5. URL https://aclanthology.org/2020.nlp4convai-1.5/.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1597–1607, Virtual, 2020. PMLR.

Maarten De Raedt, Fréderic Godin, Thomas Demeester, and Chris Develder. IDAS: Intent discovery with abstractive summarization. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pp. 71–88, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlp4convai-1.7. URL https://aclanthology.org/2023.nlp4convai-1.7/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Kangjia Fan, Yilong Zhao, Daifeng Li, Changze Lin, Weijun Zhang, and Zhiwen Zhong. LLM-enhanced generalized category discovery via iterative graph diffusion. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 645–654, Seoul, Republic of Korea, 2025. ACM. doi: 10.1145/3746252.3760997.

Lu Fan, Jiashu Pu, Rongsheng Zhang, and Xiao-Ming Wu. LANID: LLM-assisted new intent discovery. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 10110–10116, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.883/.

Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9284–9292, 2021.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552. URL https://aclanthology.org/2021.emnlp-main.552/.

Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8401–8409, Seoul, Republic of Korea, 2019. IEEE.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. doi: 10.1007/BF01908075.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, Virtual, 2020. Curran Associates, Inc.

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. 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)*, pp. 1311–1316, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1131. URL https://aclanthology.org/D19-1131/.

Jinggui Liang and Lizi Liao. ClusterPrompt: Cluster semantic enhanced prompt learning for new intent discovery. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10468–10481, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.702. URL https://aclanthology.org/2023.findings-emnlp.702/.

Jinggui Liang, Lizi Liao, Hao Fei, and Jing Jiang. Synergizing large language models and pre-trained smaller models for conversational intent discovery. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14133–14147, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.840. URL https://aclanthology.org/2024.findings-acl.840/.

Jinggui Liang, Lizi Liao, Hao Fei, Bobo Li, and Jing Jiang. 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)*, pp. 7845–7858, Mexico City, Mexico, June 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.434. URL https://aclanthology.org/2024.naacl-long.434/.

Ting-En Lin, Hua Xu, and Hanlei Zhang. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8360–8367, 2020. doi: 10.1609/aaai.v34i05.6353.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. SCAN: Learning to classify images without labels. In *Computer Vision – ECCV 2020*, pp. 268–285, Glasgow, UK, 2020. Springer. doi: 10.1007/978-3-030-58607-2_16.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7492–7501, New Orleans, LA, USA, 2022. IEEE. doi: 10.1109/CVPR52688.2022.00734.

Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16590–16600, October 2023.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. Short text clustering via convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 62–69, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-1509. URL https://aclanthology.org/W15-1509/.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14365–14373, 2021. doi: 10.1609/aaai.v35i16.17689.

Hanlei Zhang, Hua Xu, Xin Wang, Fei Long, and Kai Gao. A clustering framework for unsupervised and semi-supervised new intent discovery. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):5468–5481, 2024. doi: 10.1109/TKDE.2023.3340732.

Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. New intent discovery with pre-training and contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 256–269, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.21. URL https://aclanthology.org/2022.acl-long.21/.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. ClusterLLM: Large language models as a guide for text clustering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13903–13920, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.858. URL https://aclanthology.org/2023.emnlp-main.858/.

Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10867–10875, 2021.

Yunhua Zhou, Guofeng Quan, and Xipeng Qiu. A probabilistic framework for discovering new intents. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3771–3784, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.209. URL https://aclanthology.org/2023.acl-long.209/.

Henry Peng Zou, Siffi Singh, Yi Nian, Jianfeng He, Jason Cai, Saab Mansour, and Hang Su. Active generalized category discovery with diverse LLM feedback. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7637–7658, Rabat, Morocco, March 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-long.358. URL https://aclanthology.org/2026.eacl-long.358/.

A Datasets and Evaluation Metrics

A.1 Datasets and Splits

We follow the standard text GCD protocol on three benchmarks (Table 6). **BANKING** (Casanueva et al., 2020) is a fine-grained banking customer-service corpus with 77 intents and about 13K utterances. Queries are short, domain-specific, and often lexically similar across intents (e.g., card, transfer, and balance-related requests), so category boundaries are subtle and early neighborhood structure is especially noisy. **CLINC** (Larson et al., 2019) is a multi-domain personal-assistant intent set with 150 categories and 22.5K utterances spanning everyday domains such as banking, travel, and weather. It is comparatively balanced and less vertically specialized than BANKING, but the large category inventory still stresses open-set discovery under scarce known-class labels. **StackOverflow** (Xu et al., 2015) uses 20 topic-tag categories over 20K short question titles from a technical QA forum. Unlike the dialogue-intent datasets, categories are programming topics rather than user intents; titles are terse and jargon-heavy, and we report it under the standard text GCD protocol for comparability with prior work. For each dataset the main experiments use KCR=25% and 50%; we additionally report KCR=5% and 10% in Appendix E.3. Within each known category, 10% of the training instances are labeled; all other training instances are unlabeled and may belong to known or novel categories. We use the same train/validation/test splits as prior work (Vaze et al., 2022; Zhang et al., 2022; Zou et al., 2026). At test time, the learned embeddings are clustered with k-means into C groups, using the ground-truth number of categories.

Table 6: Dataset statistics. Known/Novel denote the numbers of known and novel categories under each KCR setting.

Dataset	Domain	#Categories	#Samples	KCR=25%	KCR=50%
BANKING	Banking	77	13,083	19 / 58	38 / 39
CLINC	Multi-domain	150	22,500	38 / 112	75 / 75
StackOverflow	Technical QA	20	20,000	5 / 15	10 / 10

A.2 Evaluation Metrics

Let $\{y_i\}$ denote ground-truth category ids and $\{\hat{y}_i\}$ the cluster assignments. Clustering accuracy (ACC) is defined after optimal permutation alignment between clusters and categories (Hungarian matching):

$$\mathrm{ACC} = \max_{p \in \mathcal{P}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i = p(\hat{y}_i)) \,, \tag{1}$$

where $\mathcal{P}$ is the set of all permutations of cluster ids. **ARI** (adjusted Rand index) (Hubert & Arabie, 1985) measures agreement between partitions, corrected for chance. **NMI** (normalized mutual information) (Strehl & Ghosh, 2002) measures shared information between assignments, normalized to $[0, 1]$. Higher is better for all three metrics. For known/novel breakdowns, we restrict evaluation to instances whose ground-truth labels fall in $\mathcal{C}_k$ or $\mathcal{C}_u$ after the same global clustering and Hungarian alignment used for the overall score.

Let ACC_k and ACC_u denote these subset accuracies. Their harmonic mean is the **H-score**,

$$\text{H-score} = \frac{2\,\mathrm{ACC}_k\,\mathrm{ACC}_u}{\mathrm{ACC}_k + \mathrm{ACC}_u}. \tag{2}$$

H-score is high only when both subset accuracies are high.

B Core Hyperparameters

Table 7 lists the main hyperparameters used in our experiments. Shared settings are shown once and centered across datasets; dataset-specific entries (neighbor size and annotation ratio ρ) are listed per column. Neighbor graphs and cluster assignments are refreshed every 5 epochs; LLM annotation runs for 3 rounds synchronized with early refreshes (epochs 0, 5, 10 under the default schedule). The default LLM annotator is Qwen3.5-Flash.

Table 7: Core hyperparameters. Shared values are centered across datasets.

Setting	BANKING	CLINC	StackOverflow
Top-K neighbors	100	120	500
Annotation ratio ρ	0.2	0.1	0.1
LLM candidate size M		30	
LLM neighbor pick		random	
Unqueried positive strategy		mutual	
Pretrain LR		5×10^{-5}	
Discovery LR		1×10^{-5}	
Discovery batch size		32	
Discovery epochs		120	
LLM annotation rounds		3	
Graph refresh interval		5	
Token-replacement prob. p_{rtr}		0.25	
Labeled ratio (within known)		0.1	

C LLM Supervision in Category Discovery

The main text focuses on what supervision LLM-based GCD methods inject into training and where that leaves positive diversity. This appendix compares methods by the answer they request and by which instances enter the query, including enrichment of utterances or category text. Some methods rewrite or enrich the input text. Others ask for an absolute category assignment. A third line asks for a relation between instances. DiPo belongs to the relational line. The differences lie in which pairs are eligible and how the answer becomes a contrastive positive.

Semantic enrichment. IDAS (De Raedt et al., 2023) prompts an LLM to write an abstractive description of each utterance and clusters the descriptions. SynCID (Liang et al., 2024a) enriches utterances and known intent labels with an LLM, then aligns the enriched text with a smaller encoder. Neither method assigns a category to an unlabeled instance or forms a training pair from an LLM judgment; both aim to make the text representation itself more reliable before clustering or alignment.

Category assignment. ALUP (Liang et al., 2024b) selects high-uncertainty instances, matches each against representatives of the current clusters, and treats the match as a pseudo-category label that soft-propagates to neighbors. LIGD (Fan et al., 2025) selects hard negatives and central unlabeled instances on a semantic-correlation graph, assigns each selected instance to a known or novel category with a two-stage prompt, and propagates the assignment along the graph. In both methods the current clustering defines the categories on offer, and propagation extends one assignment beyond the queried instance.

DeLFGCD (Zou et al., 2026) also uses this form of feedback: near each centroid the LLM writes a cluster name and description, and a later query aligns an uncertain instance with one of those descriptions. That alignment is a category decision. Its instance-level neighbor query is separate and is described below.

Relational judgments. ClusterLLM (Zhang et al., 2023) studies text clustering. Given an anchor and two candidates, the LLM chooses the closer match; given a pair, it decides whether the two instances share a category, and that decision helps set the number of clusters. The pairwise answer is a same-category judgment, not a contrastive positive under a GCD objective.

LANID (Fan et al., 2024) targets new intent discovery. It forms pairs from a local kNN graph and from DBSCAN dense regions, asks whether the two utterances share an intent, and fine-tunes a lightweight encoder with a contrastive loss on the labeled pairs. The LLM acts as a teacher to be distilled, and the setting does not use the known-class labeled split of GCD.

Loop (An et al., 2024) is a neighborhood-contrastive GCD method. Local Inconsistent Sampling keeps anchors whose neighbors disagree with them and whose cluster assignment has high entropy. Candidates for such an anchor come from the clusters that already contain most of its neighbors; the LLM selects the candidate that better matches the anchor, and Refined Neighborhood Contrastive

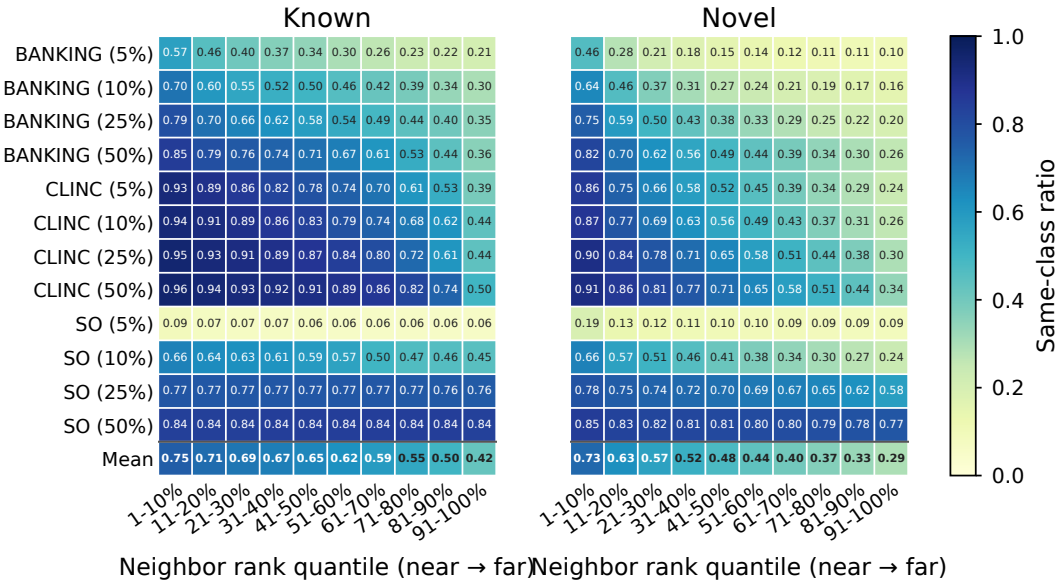


Figure 8: **Neighborhood purity of known and novel anchors.** Same-class neighbor ratio in successive Top-K rank bins on BANKING, CLINC, and StackOverflow (KCR=5%/10%/25%/50%). Left: known anchors. Right: novel anchors. Averaged over the twelve settings, mean purity drops from 0.75 to 0.42 for known anchors and from 0.73 to 0.29 for novel anchors (nearest 10% to farthest 10%).

Learning pulls toward that neighbor. The query corrects a neighborhood that the current clustering marks as inconsistent. A separate query asks the LLM to name novel clusters.

On high-entropy instances, DeLFGCD asks the LLM to select a similar neighbor and uses that neighbor as the positive in neighborhood contrastive learning. Instances outside this set keep a representation-chosen neighbor without an LLM judgment. The queried positive is a neighbor of an anchor the model already treats as ambiguous.

Where DiPo differs. Relative to Loop and DeLFGCD, DiPo does not restrict LLM queries to uncertain anchors, and it does not draw candidates only from locally agreeing clusters or from the nearest neighbors of ambiguous points. Anchors are a random fraction of the training set. Candidates are M draws from an expanded Top-K pool; each same-or-different judgment retains or discards that candidate, so a verified positive can sit well beyond the nearest ranks. Unqueried anchors receive no category label and no propagated soft label; their positive is the nearest cluster-consistent mutual neighbor (Sec. 3.3). Unlike ALUP, LIGD, and the category-alignment path in DeLFGCD, DiPo never asks the LLM to name a class or to match an instance to a cluster description. The judgment is only a filter that keeps same-class pairs that a narrow neighborhood would miss.

D Additional Diagnostics for Positive Construction

D.1 Known versus Novel Neighborhood Purity

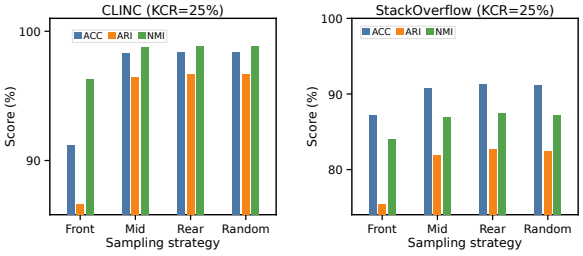
Fig. 2 pools all anchors. Novel categories have no labeled exemplars, so we repeat the same rank-bin measurement separately for known and novel anchors (Fig. 8). The pretrained encoder is evaluated on BANKING, CLINC, and StackOverflow at KCR=5%, 10%, 25%, and 50%.

As in Sec. 2.2, purity falls with neighbor rank on both panels, and the drop is steeper for novel anchors than for known ones. On BANKING and CLINC the two subsets stay close in the nearest bin and separate farther out; absolute levels are lower at KCR=5% and 10% than at higher ratios, most visibly on BANKING. StackOverflow at KCR=5% is an exception: under a one-category known split, both panels stay low across ranks. Neighborhood purity is therefore uneven over the category space, and novel-class neighborhoods degrade more sharply beyond the nearest ranks.

D.2 Oracle Candidate Sampling on CLINC and StackOverflow

Sec. 2.3 compares Front, Mid, Rear, and Random sampling on BANKING under an oracle that keeps only same-class pairs among the M sampled neighbors. We repeat that protocol on CLINC and StackOverflow at KCR=25% with M=30 and the dataset Top-K used in the main experiments (Fig. 9).

Front is the weakest strategy on both datasets. Mid, Rear, and Random stay close to each other and above Front. Every retained pair is ground-truth consistent, and each strategy still contributes one neighborhood positive per anchor, so the gap is not due to purity or to positive count. Thus, as on BANKING, useful same-class relations are not concentrated among the nearest neighbors.



(a) CLINC. (b) StackOverflow.

Figure 9: **Oracle candidate sampling on CLINC and StackOverflow** (KCR=25%, M=30). (a) CLINC (K=120). (b) StackOverflow (K=500). Best ACC/ARI/NMI when only the sampled neighbors are ground-truth filtered. Front is weaker than Mid, Rear, and Random on both datasets.

D.3 t-SNE Training Dynamics

Figs. 1a and 3 show that removing false-positive neighbors improves accuracy. Fig. 10 shows the corresponding embedding geometry on StackOverflow at KCR=25%. Rows are ordered as NCL known, DiPo known, NCL novel, and DiPo novel. Columns are the start of discovery and 25%, 75%, and 100% of each run, using the saved checkpoint nearest to that fraction of discovery-stage training. Discovery begins from the encoder pretrained with cross-entropy on labeled known-class samples and masked language modeling on all training instances.

Under NCL, most of the reorganization occurs early. Later columns change little for known and novel categories alike, so the geometry induced by neighborhood positives is consolidated and left in place. DiPo reaches a clearer class separation on both subsets by mid and late discovery. The figure compares the resulting embeddings; it does not by itself identify which design choice produces the gap.

E Additional Results

E.1 Known versus Novel Discovery

Table 8 reports DiPo known- and novel-class ACC, ARI, and NMI, together with H-score, on all three datasets under KCR=5%/10%/25%/50%. Numbers at KCR=25% and 50% use default DiPo; the lower-KCR rows use the DiPo runs reported in Appendix E.3. H-score is defined in Appendix A. The compact BANKING comparison against baselines at KCR=25% and 50% is in Table 4. Across settings, novel-class scores stay close to known-class scores on CLINC and BANKING. StackOverflow is more sensitive at low KCR, consistent with its coarser category structure and smaller labeled known-class pool; H-score drops most clearly there at KCR=5%. Under that split the known subset is a single class, so known-class ARI/NMI are undefined and reported as 0.00.

E.2 Comparison with Prior Methods

Fig. 11 evaluates known-class and novel-class discovery on BANKING at KCR=25% and 50%, in the same three-panel layout as in Loop. Numbers for the earlier methods are taken from Loop's known category ratio comparison (An et al., 2024); DiPo and ALUP follow the protocol of Table 8. Table 4 reports the same setting in tabular form.

DiPo is highest on every metric at both ratios, and its novel-class ACC stays close to its known-class ACC. The other methods drop more on novel categories; Loop remains the closest baseline with full coverage at both ratios.

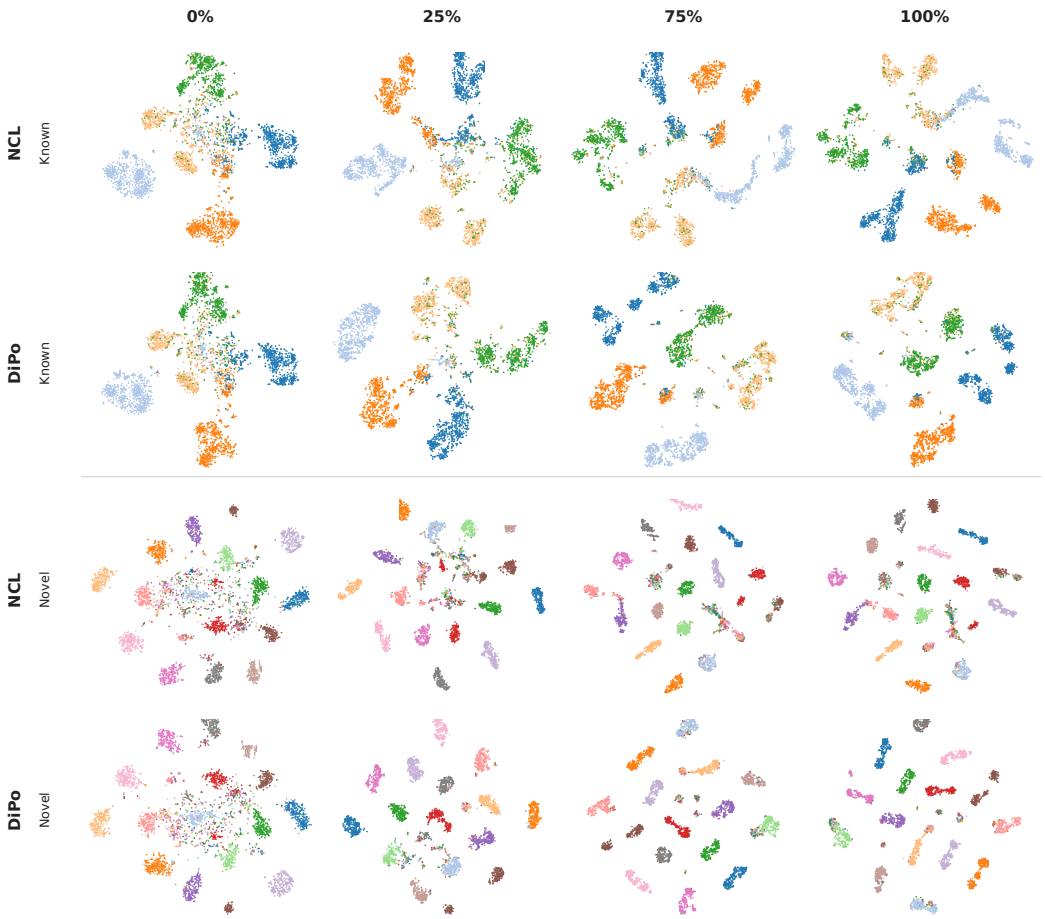


Figure 10: **Discovery embeddings on StackOverflow** (KCR=25%). Rows: NCL known, DiPo known, NCL novel, DiPo novel. Columns: start of discovery and 25%, 75%, and 100% of discovery-stage training. Under NCL the layout settles early and stays nearly fixed; DiPo yields a clearer separation by the later stages.

Table 8: Known vs. novel discovery of DiPo (%). H-score is the harmonic mean of known- and novel-class ACC. Subset ACC, ARI, and NMI are evaluated after Hungarian alignment on the full test set.

Dataset	**KCR**	**H-score**	**Known**			**Novel**		
			ACC	ARI	NMI	ACC	ARI	NMI
CLINC	5%	94.69	98.33	98.08	98.55	91.31	86.85	96.23
	10%	93.25	95.11	93.99	97.08	91.46	87.21	96.26
	25%	94.22	95.26	94.05	97.36	93.21	89.75	96.79
	50%	96.52	97.24	95.41	98.17	95.82	94.27	97.99
BANKING	5%	67.41	71.88	67.51	75.57	63.46	55.64	80.75
	10%	74.45	74.38	73.79	82.98	74.53	65.87	85.72
	25%	81.35	83.03	79.69	87.56	79.74	73.14	87.78
	50%	84.61	85.13	79.47	89.50	84.10	80.55	90.91
StackOverflow	5%	32.65	40.00	0.00	0.00	27.58	11.53	26.66
	10%	75.26	74.00	54.65	47.95	76.56	63.20	75.88
	25%	79.50	90.00	85.78	83.87	71.20	62.89	76.86
	50%	87.20	90.20	83.42	85.46	84.40	76.11	81.05

E.3 Broader KCR Coverage

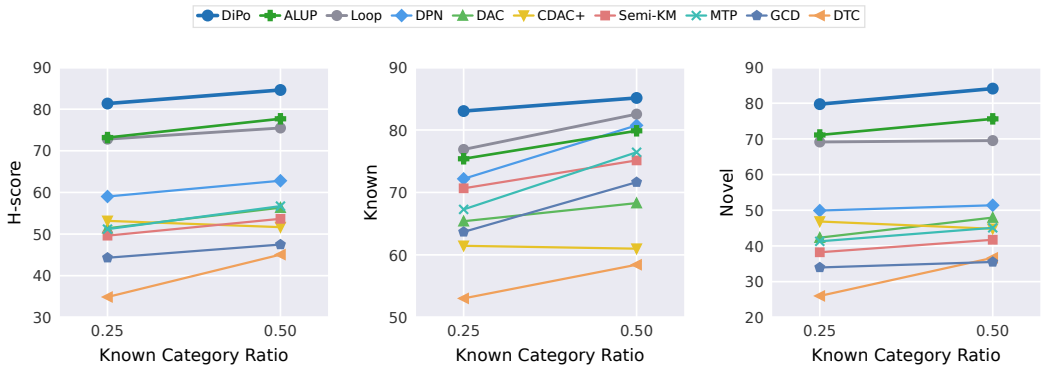


Figure 11: **Known versus novel discovery on BANKING.** Same layout as Loop's known category ratio comparison: one panel per metric, and one line per method. DiPo lies above the other methods at both ratios. Its novel-class ACC stays close to its known-class ACC, while the other methods are lower on novel categories.

Table 9: DiPo vs. NCL across KCR settings (ACC/ARI/NMI, %). ΔACC is DiPo−NCL. NCL: Top-K=50 (No+local) on BANKING/CLINC; Top-K=500 without LLM on StackOverflow. DiPo uses each dataset's default configuration.

Dataset	KCR	NCL			DiPo			ΔACC
		ACC	ARI	NMI	ACC	ARI	NMI	
BANKING	5%	40.29	27.54	62.26	68.70	59.69	82.58	+28.41
	10%	62.79	49.44	77.69	74.25	64.72	85.20	+11.46
	25%	68.93	58.63	82.35	79.38	69.53	87.29	+10.45
	50%	77.53	66.86	85.75	83.02	73.42	88.78	+5.49
CLINC	5%	84.62	77.46	93.49	92.93	88.34	96.48	+8.31
	10%	84.80	78.24	93.76	92.62	88.05	96.45	+7.82
	25%	87.16	81.35	94.51	93.69	89.57	96.80	+6.53
	50%	90.04	85.12	95.69	95.73	92.76	97.74	+5.69
StackOverflow	5%	16.20	3.09	14.96	26.70	12.31	28.60	+10.50
	10%	75.50	58.17	72.75	79.10	64.74	76.47	+3.60
	25%	81.60	63.87	75.93	87.60	75.58	82.30	+6.00
	50%	86.20	72.93	81.14	88.90	78.27	83.98	+2.70

The main tables use KCR=25% and 50%. Here we additionally report KCR=5% and 10%, covering four known-category ratios on BANKING, CLINC, and StackOverflow under one matched protocol. We compare against the NCL baseline of Sec. 2.1 with the same splits, KCR definition, labeled ratio, backbone, and test-time k-means with ground-truth C. On BANKING and CLINC, NCL follows the No+local setting of Table 2 (Top-K=50, no LLM); on StackOverflow, NCL uses Top-K=500 without LLM (the dataset default). DiPo uses the defaults in Appendix B.

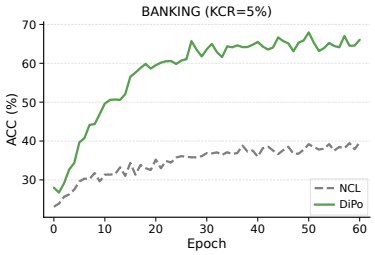


Figure 12: **BANKING at KCR=5%.** NCL vs. DiPo discovery ACC over epochs under the matched protocol.

Table 9 covers all three datasets. DiPo outperforms matched NCL at every KCR. On BANKING and CLINC, ΔACC increases as KCR decreases, in line with noisier early neighborhoods when fewer categories are known (Sec. 2.2; Appendix D.1). On StackOverflow the gains are smaller and less monotone, consistent with more reliable local neighborhoods on this coarser topic-tag set. Fig. 12 shows the discovery curves on BANKING at KCR=5%.

E.4 LLM Anchor Selection

Sec. 3.2 samples LLM-path anchors uniformly. Prior LLM-assisted GCD methods often prefer uncertain instances instead (An et al., 2024; Zou et al., 2026). We compare three rules under a matched budget $|\mathcal{A}| = \lfloor \rho N \rfloor$, with all other DiPo settings fixed (expanded Top-K, random Top-M, mutual unqueried positives, three early rounds). *Random* draws anchors uniformly. The two entropy rules score each training instance and retain the top-$|\mathcal{A}|$ scores (ties broken by distance to the assigned cluster centroid).

Neighbor-label entropy. Let $\hat{y}_j$ be the current cluster pseudo-label of instance j, and let $\mathcal{N}_{k_{\text{ent}}}(x_i)$ be the k_{ent} nearest neighbors of x_i under the current embedding (excluding itself). Define the empirical neighbor-label distribution

$$\hat{p}_{ic} = \frac{1}{k_{\text{ent}}} \sum_{j \in \mathcal{N}_{k_{\text{ent}}}(x_i)} \mathbb{I}[\hat{y}_j = c],$$

and the neighbor-label entropy

$$H_i^{\text{nbr}} = -\sum_{c=1}^{C} \hat{p}_{ic} \log \hat{p}_{ic}.$$

Higher H_i^{nbr} indicates that the local neighborhood spans more clusters.

Soft-assignment entropy. Let $\mathbf{z}_i$ be an L2-normalized embedding and $\boldsymbol{\mu}_c$ the L2-normalized k-means centroids. The soft assignment and its entropy are

$$p_{ic} = \text{softmax}_c\left(\mathbf{z}_i^\top \boldsymbol{\mu}_c / \tau\right), \qquad H_i^{\text{soft}} = -\sum_{c=1}^{C} p_{ic} \log p_{ic},$$

with temperature τ=0.1. Higher H_i^{soft} indicates a less peaked assignment over centroids.

Table 10: LLM anchor selection at KCR=25% (ACC/ARI/NMI, %). †Default. Best per column in **bold**.

Anchor rule	CLINC			BANKING		
	ACC	ARI	NMI	ACC	ARI	NMI
Random†	**93.69**	**89.57**	96.80	**79.38**	**69.53**	**87.29**
Soft-assignment entropy	93.38	89.03	96.66	79.25	68.89	86.75
Neighbor-label entropy	93.56	89.40	**96.81**	78.12	67.87	86.42

Table 10 shows that entropy-based rules do not outperform uniform sampling. On BANKING, soft-assignment entropy stays close to random, while neighbor-label entropy falls behind. On CLINC, all three rules remain within a narrow band; random leads on ACC and ARI, while neighbor-label entropy is marginally highest on NMI. Neighbor-label entropy also saturates as clusters sharpen later in training, so the ranking increasingly falls back on secondary heuristics. Uniform sampling avoids that dependence and remains competitive under a fixed ρ, which is why DiPo uses it by default.

E.5 Unqueried Positive Mining at Low KCR

Table 3 compares unqueried positive mining at KCR=25%. Table 11 repeats that comparison at KCR=10% on BANKING and CLINC. Only the positive for unqueried anchors changes. Mutual is the nearest cluster-consistent mutual neighbor (Sec. 3.3). Random draws from the same cluster-consistent pool without requiring the neighbor to retrieve the anchor in return. The LLM path stays at its default.

On BANKING, mutual again outperforms random, and the margin is wider at KCR=10% than at 25% (Tables 11 and 3). Fewer known categories leave both the early representation and the cluster assignments on this path less reliable. A one-sided neighbor is then more often a false positive. Mutual retrieval keeps the pair only when both instances retrieve each other, so this constraint matters more when known-class supervision is scarce.

Table 11: Ablation of unqueried positive mining at KCR=10% on BANKING and CLINC. ACC, ARI, and NMI (%). Only the positive for unqueried anchors changes. The LLM path is fixed at its default. Mutual is the nearest cluster-consistent mutual neighbor (Sec. 3.3); Random draws from the same cluster-consistent pool without mutual retrieval. Best in bold.

Unqueried positive	CLINC			BANKING		
	ACC	ARI	NMI	ACC	ARI	NMI
Random	89.87	83.72	95.05	69.29	57.97	82.20
Mutual	**92.62**	**88.05**	**96.45**	**74.25**	**64.72**	**85.20**

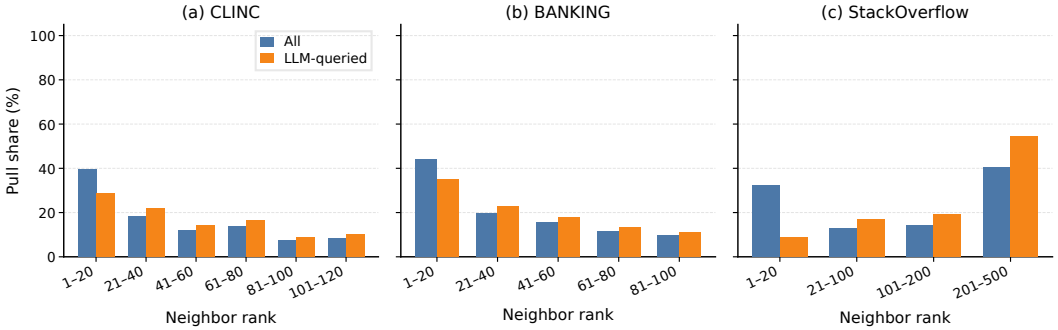


Figure 13: **InfoNCE pull share by neighbor rank** (KCR=25%). (a) CLINC. (b) BANKING. (c) StackOverflow. Blue: all training anchors; orange: LLM-path anchors. StackOverflow bins are merged for readability under K=500.

E.6 Encoded-Neighbor Pull by Rank

The contrastive objective encodes one selected neighbor x_i^+ as a second view (Sec. 3.4). For projected embeddings, let $\pi_{i,+}$ be the temperature-scaled softmax mass on the encoded pair $(\mathbf{z}_i, \mathbf{z}_i^+)$, and let $\mathcal{P}(i)$ be the SupCon positive set. The InfoNCE pull weight of that pair is

$$w_{i,+} = \left(1/|\mathcal{P}(i)| - \pi_{i,+}\right)_+,$$

summed over both directions. Near, already-aligned pairs have large $\pi_{i,+}$ and therefore small $w_{i,+}$. Pull share is the fraction of $\sum_i w_{i,+}$ that falls in a neighbor-rank bin.

We evaluate under default DiPo (random Top-M sampling on the LLM path; cluster-consistent mutual neighbors on the unqueried path; accumulated caches from epochs 0/5/10). Fig. 13 shows pull share by rank at KCR=25% on CLINC, BANKING, and StackOverflow. Table 12 collapses ranks 1–20 versus >20 at KCR=25% and 50%, and splits the training set into All, the LLM path, and the unqueried path.

On the unqueried path, selected positives fall almost entirely in ranks 1–20 at both KCR settings, so far-rank pull comes from the LLM path. On the full training set, neighbors beyond rank 20 are a minority of selected pairs, yet they carry most of the pull (Table 12; Fig. 13). Within the LLM path the imbalance is sharper, especially on StackOverflow where K=500 exposes a long tail. The same pattern holds at KCR=50% on CLINC and BANKING: unqueried positives remain local, while LLM-path neighbors beyond rank 20 again account for most of that subset's pull. Less-local verified positives are therefore uncommon under DiPo, but they dominate the attractive term.

E.7 Selected-Positive Accuracy

Table 13 reports the ground-truth accuracy of the neighbor x_i^+ that enters the contrastive loss (Sec. 3.4): the fraction of selected positives that share the anchor's category. We use the same All / LLM path / Unqueried path split as Table 12 under default DiPo at KCR=25% (BANKING and CLINC with seed 2; StackOverflow with seed 0).

Unqueried-path positives attain the highest accuracy on every dataset, consistent with mutual neighbors remaining local. LLM-path accuracy is lower on BANKING than on CLINC or StackOverflow, as verified candidates span a wider range of ranks (Sec. 2.3; Fig. 6). Sec. 2.2 measured neighborhood purity on the pretrained encoder (Fig. 2): at KCR=25%, the nearest rank band already admits

Table 12: Share of encoded neighbors (x_i^+) versus InfoNCE pull. All: full training set. LLM path / Unqueried path: anchors that do / do not receive LLM verification (Sec. 3.2, Sec. 3.3). Ranks 1–20 vs. $>$ 20 use the current kNN ranking under each dataset's default K.

KCR	Dataset	Split	Count 1–20	Count $>$ 20	Pull 1–20	Pull $>$ 20
25%	CLINC	All	81.7	18.3	39.6	60.4
		LLM path	32.1	67.9	28.7	71.3
		Unqueried path	100.0	0.0	100.0	0.0
	BANKING	All	72.6	27.4	44.0	56.0
		LLM path	43.8	56.2	35.2	64.8
		Unqueried path	100.0	0.0	99.8	0.2
	StackOverflow	All	76.0	24.0	32.5	67.5
		LLM path	10.8	89.2	9.0	91.0
		Unqueried path	100.0	0.0	99.9	0.1
50%	CLINC	All	81.1	18.9	36.9	63.1
		LLM path	29.3	70.7	25.5	74.5
		Unqueried path	100.0	0.0	100.0	0.0
	BANKING	All	71.3	28.7	42.0	58.0
		LLM path	40.4	59.6	32.8	67.2
		Unqueried path	100.0	0.0	100.0	0.0

Table 13: Selected-positive accuracy (%, KCR=25%). All / LLM path / Unqueried path follow Table 12: full training set, anchors with LLM verification, and anchors without. Accuracy is the fraction of selected neighbors that match the anchor's ground-truth category.

Dataset	All	LLM path	Unqueried path
BANKING	86.0	81.4	90.3
CLINC	96.9	96.4	97.1
StackOverflow	87.8	86.9	88.1

cross-class neighbors (same-class ratios 0.76/0.92/0.78 on BANKING/CLINC/StackOverflow), and purity falls further in farther bins (mean 0.85 nearest to 0.45 farthest across settings). Against that baseline, the selected positives in Table 13 are more reliable on every dataset, including on the LLM path where candidates are drawn beyond the local neighborhood. The two-path construction therefore supplies contrastive pairs that are more reliable than those obtained from representation-derived neighborhoods alone. Pairwise LLM label quality on the candidate pool is reported separately in Appendix F.1.

E.8 Number of LLM Annotation Rounds

DiPo queries the LLM only in early graph-refresh rounds and reuses the accumulated cache thereafter. Table 14 varies the number of annotation rounds from 1 to 4 on BANKING at KCR=25%, keeping all other hyperparameters at their defaults (K=100, M=30, ρ=0.2, random Top-M, mutual unqueried positives). Rounds are synchronized with the refresh schedule: one round annotates at epoch 0 only; two rounds at epochs 0/5; the default three rounds at epochs 0/5/10; four rounds extend to epoch 15.

Table 14: Effect of the number of LLM annotation rounds on BANKING (KCR=25%). ACC, ARI, and NMI (%). †Default.

#Rounds	ACC	ARI	NMI
1	74.12	63.50	84.43
2	76.14	66.13	85.07
3†	79.38	69.53	87.29
4	79.68	69.51	86.83

A single early round already improves over neighborhood contrastive learning without LLM verification (Table 2), and adding a second and third round continues to raise ACC, ARI, and NMI as the cache accumulates verified same-class pairs. A fourth round brings little further gain over the

three-round default. We therefore keep three annotation rounds as the default budget: most of the benefit is secured by epoch 10, and later refreshes reuse the cache without additional LLM queries.

E.9 Result Stability

Table 1 reports DiPo as the mean over three seeds. Table 15 adds the corresponding standard deviations under the default configuration (random Top-M sampling; mutual unqueried positives; Qwen3.5-Flash). The standard deviations are small on all three datasets.

Table 15: DiPo results (mean±std, %) at KCR=25% and 50%.

Dataset	KCR	ACC	ARI	NMI
BANKING	25%	79.02±1.07	69.41±0.47	86.92±0.33
	50%	83.00±0.13	73.36±0.56	88.68±0.37
CLINC	25%	93.57±0.11	89.31±0.31	96.75±0.10
	50%	95.58±0.22	92.26±0.71	97.60±0.20
StackOverflow	25%	87.00±0.85	74.34±1.75	81.90±0.57
	50%	88.85±0.07	78.11±0.23	83.92±0.08

F LLM Annotation

F.1 Annotation Quality

Table 16 expands LLM relation metrics across KCR settings under the default random Top-M policy (Qwen3.5-Flash). BANKING has lower precision than CLINC and StackOverflow at every KCR, consistent with its fine-grained vertical domain. Changing KCR moves the scores within each dataset, but it does not reverse this cross-dataset ranking. Despite the weaker BANKING labels, verify+expanded still yields the largest gains on that dataset in Table 2, so relational verification over an expanded pool remains useful even when the annotator is imperfect.

Table 16: LLM pairwise label quality from the main-result annotation caches (Qwen3.5-Flash).

Dataset	KCR	Precision	Recall	F1	Accuracy
BANKING	5%	0.61	0.85	0.71	0.84
	10%	0.69	0.86	0.76	0.81
	25%	0.74	0.81	0.77	0.78
	50%	0.79	0.80	0.79	0.75
CLINC	5%	0.84	0.84	0.84	0.92
	10%	0.95	0.85	0.90	0.89
	25%	0.96	0.81	0.88	0.85
	50%	0.97	0.79	0.87	0.83
StackOverflow	5%	0.87	0.81	0.84	0.97
	10%	0.86	0.83	0.85	0.90
	25%	0.95	0.90	0.92	0.90
	50%	0.95	0.84	0.89	0.84

F.2 Annotation Quality under Different Top-M Sampling

Complementing Table 16, we hold the annotation budget fixed at KCR=25% (M=30, same ρ and Top-K as the main experiments) and vary only how the M candidates are drawn from each anchor's Top-K neighborhood: Front (nearest third), Random (uniform), and Rear (farthest third). Metrics are scored against ground-truth same-class equality (Table 17).

Sampling changes the difficulty of the queried pairs rather than the LLM itself. Front concentrates verification on the nearest neighbors and therefore yields the highest precision on every dataset, most sharply on BANKING. Recall is comparatively stable across policies, so the F1 gap largely tracks precision. On CLINC and StackOverflow, absolute quality remains high under all three strategies,

Table 17: LLM pairwise label quality under Front / Random / Rear Top-M sampling (KCR=25%, Qwen3.5-Flash).

Dataset	Sampling	Precision	Recall	F1	Accuracy
BANKING	Front	0.84	0.85	0.84	0.79
	Random	0.74	0.81	0.77	0.78
	Rear	0.59	0.80	0.68	0.76
CLINC	Front	0.97	0.81	0.88	0.82
	Random	0.96	0.81	0.88	0.85
	Rear	0.91	0.81	0.85	0.88
StackOverflow	Front	0.96	0.87	0.91	0.87
	Random	0.95	0.90	0.92	0.90
	Rear	0.94	0.91	0.93	0.90

and pairwise accuracy even increases from Front to Rear as the sampled set includes more true negatives among far neighbors. Thus far-neighbor verification is hardest on BANKING, while CLINC and StackOverflow remain comparatively reliable away from the local band.

These quality trends should not be read as a preference for Front sampling in discovery. The candidate-sampling block in Table 3 shows Front to be the weakest Top-M rule despite its higher pairwise precision, consistent with the oracle sampling study in Sec. 2.3: useful same-class relations are not concentrated among the nearest neighbors. Random sits between Front and Rear in annotation difficulty and is the default DiPo policy under a fixed budget M.

F.3 Different LLM Annotators

Table 18 compares three compact annotators on BANKING at KCR=25% under the default DiPo configuration (K=100, M=30, ρ=0.2, random Top-M, mutual unqueried positives). We report pairwise label quality and discovery ACC/ARI/NMI. Qwen3.5-Flash is the default used in the main experiments.

Table 18: LLM annotator comparison on BANKING (KCR=25%). Pairwise metrics and discovery ACC/ARI/NMI (%). †Default annotator.

LLM	Pairwise quality				Discovery		
	Prec.	Rec.	F1	Acc.	ACC	ARI	NMI
Qwen3.5-Flash†	0.74	0.81	0.77	0.78	79.38	69.53	87.29
Qwen3.8-Flash	0.79	0.67	0.73	0.76	78.31	68.88	86.70
Doubao-Seed-2.0-mini	0.65	0.58	0.62	0.83	78.18	67.83	86.51

Under the same query schedule, Qwen3.8-Flash raises precision and lowers recall relative to Qwen3.5-Flash, while Doubao-Seed-2.0-mini is weaker on precision, recall, and F1. Discovery ACC/ARI/NMI nevertheless stay close across all three annotators. The comparison covers two Qwen variants and one Doubao model; it does not exhaust the Flash-tier API space.

F.4 Cache Reuse under Early Annotation

DiPo queries the LLM only in the first three graph-refresh rounds (epochs 0, 5, and 10) and then reuses the accumulated cache (Sec. 3.2). Table 19 reports reuse diagnostics at KCR=25% under the default main-result configuration. *Cached anchors* counts training instances with at least one cached same-or-different label after that epoch. *Pair hit* is the fraction of the current Top-M query pairs that are already in the cache. *Survive-1* is the fraction of historically verified same-class pairs that still appear in the anchor's current Top-K; LLM-path positives are sampled from this intersection. *Empty* is the fraction of cached anchors for which that intersection is empty; those anchors use the Top-K fallback in Sec. 3.2.

After round 3 the cache size freezes on every dataset, and the unique-anchor count stays well below $3\times\rho N$ because later random draws overlap earlier ones. API calls therefore stop while training continues.

Table 19: LLM cache reuse under default DiPo (KCR=25%). Annotation runs at epochs 0/5/10; later epochs reuse the cache only. Cached anc.: cumulative anchors with cached labels. Pair hit: fraction of current Top-M query pairs already cached. Survive-1: fraction of cached same-class pairs still in the current Top-K. Empty: fraction of cached anchors with no surviving same-class neighbor in Top-K.

Dataset	Epoch	Cached anc.	Pair hit	Survive-1	Empty
CLINC	0 (round 1)	1,800	0.00	1.00	0.01
	5 (round 2)	3,407	0.03	0.98	0.01
	10 (round 3)	4,860	0.05	0.98	0.01
	35 (reuse)	4,860	0.06	0.97	0.01
BANKING	0 (round 1)	1,800	0.00	1.00	0.01
	5 (round 2)	3,241	0.05	0.93	0.01
	10 (round 3)	4,403	0.09	0.92	0.01
	35 (reuse)	4,403	0.11	0.84	0.01
StackOverflow	0 (round 1)	1,798	0.00	1.00	0.08
	5 (round 2)	3,397	0.00	0.95	0.08
	10 (round 3)	4,844	0.01	0.93	0.07
	35 (reuse)	4,844	0.01	0.82	0.07

Survive-1 and Pair hit answer different questions (Table 19). On CLINC, Survive-1 stays high through mid-training, so verified same-class pairs remain available inside the refreshed Top-K without new queries. BANKING shows a larger Survive-1 drop by epoch 35 (0.84), while Pair hit remains modest (0.11), consistent with greater neighborhood churn on this fine-grained domain. Pair hit at the same snapshots remains low on all three datasets: few of the *current* Top-M candidates were labeled in the early rounds. The neighborhood is therefore moving, yet early same-class labels remain usable. StackOverflow shows the same pattern under a larger pool (K=500), with a larger Survive-1 drop by mid-training, consistent with greater rank churn. Even there, most verified same-class pairs survive, and Pair hit stays near zero. Empty pools remain rare on CLINC and BANKING (about 1% of cached anchors). On StackOverflow the empty rate is higher (7–8%), in line with coarser categories and a larger Top-K, but the majority of cached anchors still retain at least one verified same-class neighbor. Thus early annotation with cache reuse continues to supply LLM-path positives after the neighborhood refreshes; it does not imply that a live Top-M query would mostly repeat already labeled pairs.

F.5 Annotation Cost

Table 20: LLM annotation budget under Qwen3.5-Flash (three early rounds). Input/output tokens are reconstructed from the saved prompts and responses with a Qwen tokenizer. Cost is the billed USD total for those rounds.

Dataset	Input tok.	Output tok.	Cost ($)
BANKING	5,428,714	405,775	0.30
CLINC	5,019,416	447,846	0.30
StackOverflow	5,454,988	540,939	0.55

Table 20 reports absolute token usage and billed USD cost for the three early annotation rounds under the default DiPo schedule.

F.6 Prompt Templates

Each queried anchor is sent in one LLM call with M numbered neighbors. The model returns JSON of the form

```
{"anchor_index": <id>, "same": [<neighbor_ids>...]}
```

listing 1-based indices of neighbors judged as the same category; unlisted neighbors are treated as different. No category names are provided. The three datasets share this protocol and differ mainly in the same-category criterion:

Dataset	Criterion	Same iff
BANKING	banking *action*	same requested system action
CLINC	user *goal*	same assistant goal
StackOverflow	primary *platform*	same tech./product ecosystem

The BANKING system prompt is given below. CLINC and StackOverflow keep the same output format and decision rules, substituting the criterion in the table (with dataset-specific examples) for the BANKING action rule.

```
You are a banking intent classification assistant. Your task is to determine
whether the anchor query and its candidate neighbors belong to the same
intent category.

Before judging each anchor-neighbor pair, first read all candidate neighbors
as a whole to understand the surrounding context, rather than evaluating each
neighbor in isolation.

Judge by the requested action, not by surface form.
- SAME intent: the two utterances ask the system to perform the same banking
  action, even if wording, entities, amounts, or account details differ
  (e.g., "what's my balance" vs "how much money is in my checking account").
- DIFFERENT intent: the requested actions differ, even when keywords overlap
  (e.g., "transfer money" vs "check transfer status"; "activate card" vs
  "report lost card").

Surface similarity is evidence, not proof: shared wording does not make two
utterances the same intent, and different wording does not make them different.

**Output Format**: Output a valid JSON object with the following structure:
{"anchor_index": <index>, "same": [<neighbor_number>, ...]}
- List ONLY the neighbor numbers (1-based, as shown in the neighbor list)
  that share the SAME intent as the anchor.
- Neighbors not listed are treated as DIFFERENT intent.
- If none share the same intent, output an empty list: "same": []

Example: if neighbors 2 and 5 match the anchor, output
{"anchor_index": 123, "same": [2, 5]}

IMPORTANT:
1. Neighbor numbers MUST be integers from 1..num_neighbors.
2. Include a neighbor only when confident; omit borderline pairs.
   A wrong match costs more than a missed one.
3. Clear paraphrases are confident matches.
4. Keyword overlap alone is never enough.
5. Judge each neighbor independently; the number of matches is not fixed.
```

F.7 Case Study of LLM Pairwise Intent Labeling on BANKING

BANKING is a fine-grained vertical-domain benchmark: many of its 77 intents share near-identical lexical templates and differ only by a subtle functional cue (e.g., use vs. hold a currency, fee vs. limit, order vs. link a card). We therefore use it to inspect LLM pairwise same-or-different judgments qualitatively. False positives typically arise when surface-similar queries belong to distinct intents; false negatives arise when same-intent paraphrases omit a cue the LLM overweights (e.g., "specific date", mention of "card"); correct hard negatives usually hinge on catching that single cue. Notation: A denotes the anchor. FP, FN, and TN denote cases where a gold label of different is predicted as same ($0\to1$), a gold label of same is predicted as different ($1\to0$), and a gold label of different is predicted as different ($0\to0$), where 1 means same-class.

Failure modes. *Currency / payment-method.*
A "what currencies can i use?" (`supported_cards_and_currencies`)
FP "which flat currencies can i use from you?" (`fiat_currency_support`)
FP "what currencies can i use to top up my account?" (`transfer_into_account`)
FN "what card or currency can i use to pay?" (`supported_cards_and_currencies`)

Card delivery schedule vs. delayed arrival.
A "can cards be dispatched on a specific date?" (`card_delivery_estimate`)
FN "how soon do cards arrive after i order them?" (`card_delivery_estimate`)
FN "when will i receive the card?" (`card_delivery_estimate`)
A "i am still waiting for my card." (`card_arrival`)
FP "how long until the card arrives?" (`card_delivery_estimate`)

Declined transfer vs. declined card payment.
A "... my credit keeps being declined ... what is the status of my card?" (`declined_transfer`)
FP "today i was out eating and my card was declined. why's that?" (`card_not_working`)
FP "why has my card been declined? whats the deal?" (`declined_card_payment`)

Top-up fee channel / status.
A "what are the top-up fees for my card?" (`top_up_by_card_charge`)
FP "what is the fee to top-up my account" (`top_up_by_bank_transfer_charge`)
FN "will i be charged a fee for topping up using a european card?" (`top_up_by_card_charge`)
A "i am unable to use my top-up on the app. could you explain this to me?" (`top_up_reverted`)
FP "i tried to top-up, why didn't it work?" (`top_up_failed`)
FP "is my top up not working?" (`pending_top_up`)

Hard negatives labeled correctly. *Use vs. hold / exchange.*
A "what currencies can i use?" (`supported_cards_and_currencies`)
TN "what currencies can i hold?" (`fiat_currency_support`)
TN "what different currencies can i get?" (`exchange_via_app`)

Fee vs. limit / payment fee.
A "what are the top-up fees for my card?" (`top_up_by_card_charge`)
TN "what is the top-up limit for my card?" (`top_up_limits`)
TN "is there a fee for paying by card?" (`card_payment_fee_charged`)

Order vs. link a card.
A "where do i order a card?" (`order_physical_card`)
TN "how do i add a new card?" (`card_linking`)
A "how do i link a new payment card?" (`card_linking`)
TN "how do i order a card?" (`order_physical_card`)

ATM vs. acceptance; delivery vs. related card intents.
A "where can i find mastercard atms?" (`atm_support`)
TN "where can i use my mastercard?" (`card_acceptance`)
A "can cards be dispatched on a specific date?" (`card_delivery_estimate`)
TN "when will my new card be delivered?" (`card_arrival`)
TN "are cards available to those outside of the uk?" (`country_support`)

Overall, LLM pairwise judgments are informative but brittle on BANKING: the model often catches the key functional cue, yet still confuses near-template neighbors when that cue is weak or omitted.

G LIMITATIONS

LLM labeling errors. DiPo uses the LLM for same-or-different verification only. Pairwise quality is high on CLINC and StackOverflow, but lower on BANKING under the default random Top-M policy (Table 5), where near-template intents are often confused (Appendix F.7). False positives that pass verification can still enter the contrastive loss. A conservative prompt and a partial query budget reduce this exposure, but they do not remove annotator error on fine-grained vertical domains.

API-based annotation. All main results use Qwen3.5-Flash under a fixed early-round budget. A matched Qwen3.8-Flash run and a Doubao-Seed-2.0-mini run on BANKING remain close in discovery scores (Appendix F.3), but the comparison is still limited in provider coverage. The method

therefore assumes API access in the first few discovery refreshes. We have not tested a distilled local judge or fully offline annotation. Caching keeps the cost modest on our benchmarks (Appendix F.5), but larger or repeatedly refreshed corpora would still need a budgeted schedule.

Static early cache. LLM queries run only in the first three graph-refresh rounds; later epochs reuse the accumulated cache. Under this schedule, early verified pairs largely remain inside the refreshed Top-K, while few of the current Top-M candidates were labeled in those rounds (Appendix F.4). The cache is not recomputed as the neighborhood continues to move, so stale pairs may linger and newly informative far neighbors that appear late are never verified. Selective late refresh under a fixed budget is not studied here.

Unqueried path and initialization. Most anchors never receive an LLM judgment. Their positives are nearest cluster-consistent mutual neighbors, so errors in early clustering or in the pretrained encoder can still propagate on that path, especially at low known-category ratios (Appendix E.5). DiPo also follows the standard two-stage text GCD pipeline with a fixed BERT backbone. Other encoders and end-to-end training without that initialization are untested.

Evaluation scope. Experiments cover English short-utterance and title GCD on BANKING, CLINC, and StackOverflow, with a fixed BERT encoder and with the ground-truth category count C given to test-time k-means. Transfer to non-English data, long documents, unknown cardinality, or other modalities is not evaluated. The prompts further encode dataset-specific same-category criteria (Appendix F.6), which are task priors rather than universal semantic boundaries. Main tables use single-split reporting for comparability with prior work; mean$\pm$std appears in Appendix E.9.