

Lifelong Intent Discovery: A Continual Learning Protocol and Method

Peng Chen^{1*}, Yu Xia^{2*}, Bang Wang^{2,†},

¹School of Software Engineering,

Huazhong University of Science and Technology, Wuhan, China

²School of Electronic Information and Communications,

Huazhong University of Science and Technology, Wuhan, China

{hustchenpeng, hustxiayu, wangbang}@hust.edu.cn

Abstract

Discovering new intents is crucial for many systems to adapt to evolving users' needs, making it inherently a continual process. However, most related studies have not explored such continual learning scenarios for new intent discovery. To bridge this research gap, we formalize the *Continual Open-Stream Intent Discovery* (COSID) task, which requires simultaneous detection of historical intents and discovery of new ones from hybrid streaming data. Our central contribution is a configurable *open-stream dataset construction protocol* that transforms a static intent dataset into a stream of temporally partitioned datasets, as well as a set of evaluation metrics for the COSID task. We further propose the *Continual Intent Discovery Agent* (CIDA) for the task, a memory-augmented agent that screens the incoming stream, clusters new-intents' samples, and employs an LLM for intent abstraction and memory expansion. Experiments on the ClinCL and BankCL data streams constructed according to our protocol validate both the protocol expressiveness and agent superiority over existing baselines. Our code is available at: <https://github.com/hustchenpeng/COSID>

1 Introduction

Task-oriented conversation systems rely on accurately understanding the intents of users' queries. The *intent detection* (ID) task has achieved remarkable success, where all intents are pre-defined and each query is annotated with a corresponding intent label (Singhal et al., 2023; Zhang et al., 2023). However, real-world user queries are inherently dynamic: Users frequently express needs that fall outside the predefined intent scope, resulting in new intents continuously emerging (Chen and Wang, 2025). To address this, *out-of-scope* (OOS) detection (Larson et al., 2019) was introduced to flag

samples as unknown intents. With the goal of moving beyond mere detection to discover previously unseen intents, Zhang et al. (2021) proposed the task of *new intent discovery* (NID), which aims to not only detect queries with unknown intents but also group them into distinct clusters.

However, NID frames intent discovery as a one-pass task. This fails to characterize real-world applications, as new intents emerge incrementally throughout the timeline. To emulate this lifelong learning process, Song et al. (2023b) introduced *Continual Generalized Intent Discovery* (CGID), which requires the system to discover new intents without retraining from scratch. Nevertheless, the CGID adopts an idealized premise wherein the streaming data of each stage exclusively comprises samples only from new intents. In realistic applications, incoming data streams normally contain not only queries with new intents but also queries belonging to historical intents that the system has already discovered.

In this paper, we define a more realistic intent discovery task: *Continual Open-Stream Intent Discovery* (COSID), where the data stream contains a mixture of samples from both old intents and new intents in each stage. The objective is not only to detect old intents but also to conceptualize new intents for expanding the system intent knowledge base. Failing to discriminate the samples would severely degrade the detection of old intents and interfere with the discovery of new intents. Furthermore, another challenge for the COSID task is that there exists no such an open-stream dataset for system evaluation.

Although several intent detection datasets including CLINC (Larson et al., 2019) and BANKING (Casanueva et al., 2020) are publicly available, they cannot be directly adapted to the COSID task. They only contain static labeled samples and lack temporal partitioning, which is indispensable for depicting real-world evolving intent dynamics.

*These authors contributed equally.

†Corresponding author.

This constraint greatly impedes COSID research, thereby necessitating a standardized protocol to construct continuous streaming datasets from existing resources. In addition, conventional evaluation metrics for intent discovery require revision and expansion to accommodate the continual learning nature of the COSID task.

In this paper, we propose an *open-stream dataset construction protocol* and an evaluation framework for COSID. Differing from simple sample partitioning, the protocol emulates diverse practical scenarios in continual learning by designing multi-dimensional configurations for stream construction, thereby posing different levels of difficulties for the COSID task. The protocol also specifies the construction of test datasets at each stage. Furthermore, for comprehensive COSID assessment, we devise a set of performance metrics to quantify three core abilities: mixed stream discrimination, new intent discovery and historical knowledge retention.

In this paper, we also design a *Continual Intent Discovery Agent (CIDA)* for the COSID task. The agent comprises a *long-term memory* and three functional modules. Driven by the dynamically growing memory, the agent employs a *discrimination module* to first perform per-sample screening on the incoming mixed data stream, discriminating samples with old intents from new intents. The filtered samples as with new intents are then processed by a *discovery module*, which are first grouped into distinct clusters. For each cluster, this module leverages a Large Language Model (LLM) to conceptualize the intent label and description of the cluster. The *inference module* leverages the updated memory for the intent discovery.

To evaluate the COSID task, we construct continual learning data streams based on our proposed open-stream dataset construction protocol. Extensive experiments on these constructed data streams not only validate the effectiveness of the protocol in emulating complex, real-world intent emergence dynamics, but also demonstrate the substantial superiority of our proposed CIDA method. Our approach consistently exhibits outstanding performance and robustness across mixed stream screening, new intent discovery, and historical knowledge retention. A comprehensive discussion of related work is deferred to Appendix A.

2 Task Definition and Dataset Construction Protocol

Task Definition: We define the task of *Continual Open-Stream Intent Discovery* (COSID) as a continual learning process to discover the intents of arrival samples across $T + 1$ stages. In the initial stage ($t = 0$), a labeled sample set comprising some *base intents* is provided to initialize an intent discovery model. In each subsequent stage $t \geq 1$, the input of the model arrives as an unlabeled data stream containing a mixture of previously discovered intents and emerging new intents. The goal at each stage $t \geq 1$ is to first identify samples with previously discovered intents and next discover new intents from those unidentified samples.

Dataset Construction Principle: Existing intent discovery datasets, such as CLINC (Larson et al., 2019) and BANKING (Casanueva et al., 2020), contain labeled samples but cannot be directly applied to the COSID task, as they lack temporal divisions to capture intent evolution dynamics. Consider a typical dataset $\mathcal{D}$ with K samples across N distinct intents. For the COSID task, we need to construct a dataset stream $\{\mathcal{D}^{(t)}\}$ from $\mathcal{D}$.

We design a **protocol** to construct $\{\mathcal{D}^{(t)}\}$ from $\mathcal{D}$ following three dimensions. (1) Number of per-stage new intents: In the real-world, the number of new intents fluctuates dynamically over time rather than arriving at a constant rate. (2) Distribution of per-stage new intents: Intents emerging concurrently may exhibit distinct semantic relationships. They may be tightly clustered around a specific trending topic or dispersed across disparate topics. We need to define the semantic cohesion among concurrently arriving intents. (3) Distribution of per-stage samples: Natural data streams are inherently heterogeneous, with samples of recurring historical intents constantly mixing with samples of sparse and imbalanced new intents, where their distributions should be regulated by the protocol.

Figure 1 illustrates the operational pipeline of the protocol. It first regulates how to construct the new-intent sets in each stage, including the number and the distribution of new-intents. In each stage, an intent is called a *new-intent* if it is the first time appearing in the system; Otherwise, it is called an *old-intent*. The protocol then regulates how to construct the sample sets in each stage.

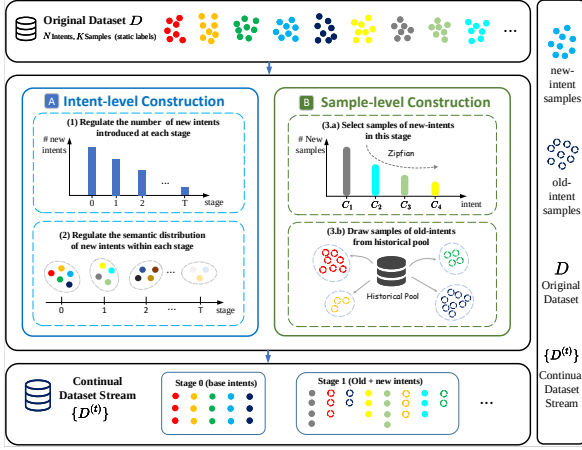


Figure 1: Illustration of the operational pipeline of the proposed dataset construction protocol.

2.1 Number of Per-stage New Intents

The protocol first regulates the number of new-intents arriving in each stage. A straightforward approach is to allocate new-intents equally in each stage. Yet to fit realistic uneven emergence that new-intents surge first then level off with intent space expansion, we propose a decreasing strategy.

Let n_t denote the number of new-intents in the t -th stages, and $\sum_t n_t = N$. At the initial stage t_0 , we set $n_0 = \rho N$ with ρ a predefined parameter. The remaining $N - n_0$ intents are partitioned into T disjoint subsets $\mathcal{C}_1, \dots, \mathcal{C}_T$ scheduled over the incremental stages, ensuring that no intents are repeated across different stages (i.e., $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for all $i \neq j$). Under the decreasing strategy, the number of new-intents $n_t = |\mathcal{C}_t|$ at the t -th stage is computed by

$$n_t = \left\lfloor \frac{2(N - n_0)}{T(T + 1)} \cdot (T - t + 1) \right\rfloor. \quad (1)$$

Any rounding residual $r = N - \sum_{t=0}^T n_t$ is absorbed by incrementing $n_1, \dots, n_r$ each by one.

2.2 Distribution of Per-stage New Intents

The protocol next controls the semantic distribution of new-intents in different stages. A straightforward approach is a **Random** distribution strategy, where all intents are randomly allocated into each stage, regardless of their semantic similarities. In the real-world, the semantics of new-intents arriving at one stage could also be either with high similarities or low similarities. To emulate this phenomenon, we propose two distribution strategies, namely **Cohesion** and **Expansion**.

We first conduct a *semantic correlation analysis* to quantify the semantic relationships for all pairs of intents. Specifically, we represent each intent c by an intent prototype $\mathbf{v}_c$, computed as the ℓ_2 -normalized mean of the unit-norm embeddings of all samples belonging to this intent c in the original dataset $\mathcal{D}$, extracted via a frozen sentence encoder. The semantic similarity between two intents c and c' is computed by $\cos(\mathbf{v}_c, \mathbf{v}_{c'})$. Let $\mathbf{M}$ denote the semantic similarity matrix for all intents.

We employ a greedy procedure to select intents into each stage. Let $\mathcal{C}$ denote the set of all intents. At first, we randomly select n_0 intents from $\mathcal{P}_0 = \mathcal{C}$ to construct the base intent set $\mathcal{C}_0$ in the initial stage t_0 . In the 1st stage t_1 , we select n_1 intents from the candidate pool $\mathcal{P}_1 = \mathcal{C} \setminus \mathcal{C}_0$ to construct $\mathcal{C}_1$. This procedure continues until all intents are distributed in each stage.

Cohesion distribution strategy: It selects the most semantically similar intents in each stage. Take the t -th stage for example. We first select a *central seed* $c_1 = \arg \max_{c \in \mathcal{P}_t} \sum_{c' \in \mathcal{P}_t} \mathbf{M}_{cc'}$ and $\mathcal{C}_t = \mathcal{C}_t \cup \{c_1\}$. This central seed acts as the semantic core of the candidate pool $\mathcal{P}_t$, representing the intent with the highest cumulative similarity to all other available intents. Next, we iteratively select an intent that maximizes the average similarity to $\mathcal{C}_t$ as follows:

$$c^* = \arg \max_{c \in \mathcal{P}_t \setminus \mathcal{C}_t} \frac{1}{|\mathcal{C}_t|} \sum_{c' \in \mathcal{C}_t} \mathbf{M}_{cc'}. \quad (2)$$

This objective ensures that each newly selected intent maintains maximum semantic proximity to those already selected intents in $\mathcal{C}_t$, iteratively growing a tightly semantically cohesive subset.

Expansion distribution strategy: It selects the most semantically dissimilar intents in each stage. A *peripheral seed* $c_1 = \arg \min_{c \in \mathcal{P}_t} \sum_{c' \in \mathcal{P}_t} \mathbf{M}_{cc'}$ is first selected, and $\mathcal{C}_t = \mathcal{C}_t \cup \{c_1\}$. This peripheral seed acts as the semantic outlier of the candidate pool $\mathcal{P}_t$, representing the intent with the lowest cumulative similarity to all other available intents. Next, we iteratively select an intent that minimizes its maximum similarity to $\mathcal{C}_t$ as follows:

$$c^* = \arg \min_{c \in \mathcal{P}_t \setminus \mathcal{C}_t} \left(\max_{c' \in \mathcal{C}_t} \mathbf{M}_{cc'} \right). \quad (3)$$

This objective ensures that each newly selected intent maintains maximum semantic distance from those already selected intents in $\mathcal{C}_t$, iteratively growing a set of semantically dispersed intents and

also scattering semantically similar intents across different stages.

2.3 Distribution of Per-stage Samples

The first two steps determine the sets of new-intents for each stage, that is, $\{\mathcal{C}_t\}$ with $|\mathcal{C}_t| = n_t$. In the third step, the protocol further regulates the samples in each stage, including the new-intents' samples and the mixing proportion of recurring old-intents' samples, i.e., determining $\{\mathcal{D}^{(t)}\}$.

Take the t -th stage for example. For each intent c_i in $\mathcal{C}_t$, let $\mathcal{K}_i$ denote the set of samples with this intent label c_i in the original dataset $\mathcal{D}$. We divide $\mathcal{K}_i$ into two disjoint subsets: $\mathcal{K}_i^{\text{io}}$ containing the samples in $\mathcal{K}_i$ as the initial occurrence in the t -th stage, and $\mathcal{K}_i^{\text{so}} = \mathcal{K}_i \setminus \mathcal{K}_i^{\text{io}}$ containing the samples as subsequent recurrences in the later stages. Furthermore, we maintain a global reservoir $\mathcal{G}$ to include $\mathcal{K}_i^{\text{so}}$ for all intents in all stages, which serves to supply samples of recurring old-intents.

Long-tailed sample distribution Note that $\mathcal{C}_t$ normally contains more than one intent. We can set $|\mathcal{K}_i^{\text{io}}| = K_{\text{init}}$ for all intents in $\mathcal{C}_t$, where K_{init} is a predefined parameter, denoting the number of per-new-intent samples. However, the samples across different intents could not be guaranteed as uniform distributed in real-world, and often follows a long-tailed distribution where a few intents with much more samples. To emulate this, the protocol adopts a Zipfian schedule governed by a skewness parameter s , where the sample count $K_i^{\text{io}} = |\mathcal{K}_i^{\text{io}}|$ for a rank index $i \in \{1, \dots, n_t\}$ follows:

$$K_i^{\text{io}} \propto i^{-s}, \quad \text{s.t.} \quad \frac{1}{n_t} \sum_{i=1}^{n_t} K_i^{\text{io}} = K_{\text{init}}, \quad (4)$$

By solving for the normalization constant, the count for each rank is computed by

$$K_i^{\text{io}} = \max \left(1, \left\lfloor \frac{n_t \cdot K_{\text{init}}}{\sum_{j=1}^{n_t} j^{-s}} \cdot i^{-s} \right\rfloor \right), \quad (5)$$

with any rounding residual distributed in a head-first round-robin fashion across the rank-ordered intents until the budget $\sum_{i=1}^{n_t} K_i^{\text{io}} = n_t K_{\text{init}}$ is matched exactly. Next, in total K_i^{io} samples are randomly drawn from $\mathcal{K}_i$ to form $\mathcal{K}_i^{\text{io}}$ for each c_i in $\mathcal{C}_t$. Let $K_t^{\text{new}} = \sum_{i=1}^{n_t} K_i^{\text{io}}$ denote the number of new-intents' samples in the t -th stage.

Old-Intent Recurrence In real-world data streams, samples of new-intents are continuously

intertwined with those samples with recurring old-intents, posing a challenge for a model to distinguish new-intents from already discovered ones. To emulate this, the protocol mixes new-intents' samples with old-intents' samples in each stage by a *mixing ratio* $\alpha \in [0, 1)$. The number of old-intents' samples to be mixed is computed as:

$$K_t^{\text{old}} = \left\lfloor \frac{\alpha}{1-\alpha} K_t^{\text{new}} \right\rfloor. \quad (6)$$

In each stage, the protocol randomly draws K_t^{old} samples from the global reservoir $\mathcal{G}$, without considering their intent labels and without replacement, to mix with new-intents' samples. The smaller the α , the fewer the old-intents' samples been mixed. Notice that $\alpha = 0$ corresponds to only new-intents' samples in each stage.

2.4 Test Dataset Construction Protocol

Following the evaluation framework in the CGID (Song et al., 2023b) and continual learning in the computer vision domain (Dai and Chauhan, 2025), we adopt a cumulative testing protocol. Let $\mathcal{D}_{\text{test}}$ denote the test dataset corresponding to $\mathcal{D}$. We construct a sequence of cumulative test datasets $\{\mathcal{D}_{\text{test}}^{(0)}, \dots, \mathcal{D}_{\text{test}}^{(T)}\}$ from $\mathcal{D}_{\text{test}}$ as follows. For each stage t , all samples in $\mathcal{D}_{\text{test}}$ whose labels belong to the intent set $\mathcal{C}_t$ are included into the current test set $\mathcal{D}_{\text{test}}^{(t)}$ and $\mathcal{D}_{\text{test}}^{(t)} = \mathcal{D}_{\text{test}}^{(t-1)} \cup \mathcal{D}_{\text{test}}^{(t)}$. So $\mathcal{D}_{\text{test}}^{(t-1)} \subset \mathcal{D}_{\text{test}}^{(t)}$. This setup is designed to evaluate the system ability in the continual learning process: robustness of identifying old-intents and effectiveness of discovering new-intents in each stage.

3 Evaluation Metrics

We propose to evaluate the COSID task from three aspects. More details on the calculation of these performance metrics are provided in Appendix B.

Discrimination metrics assess the model capability to distinguish new-intents from old-intents.

- **D-score:** We propose D-score to evaluate the model ability to distinguish between new-intents and old-intents in each stage.

- **OSR (Over-screening rate).** We propose OSR to evaluate the degree to which the model incorrectly flags old-intent samples as new-intents.

- **USR (Under-screening rate).** We propose USR to evaluate the degree to which the model incorrectly flags new-intent samples as old-intents.

Discovery metrics evaluate the capability of clustering new-intents' samples.

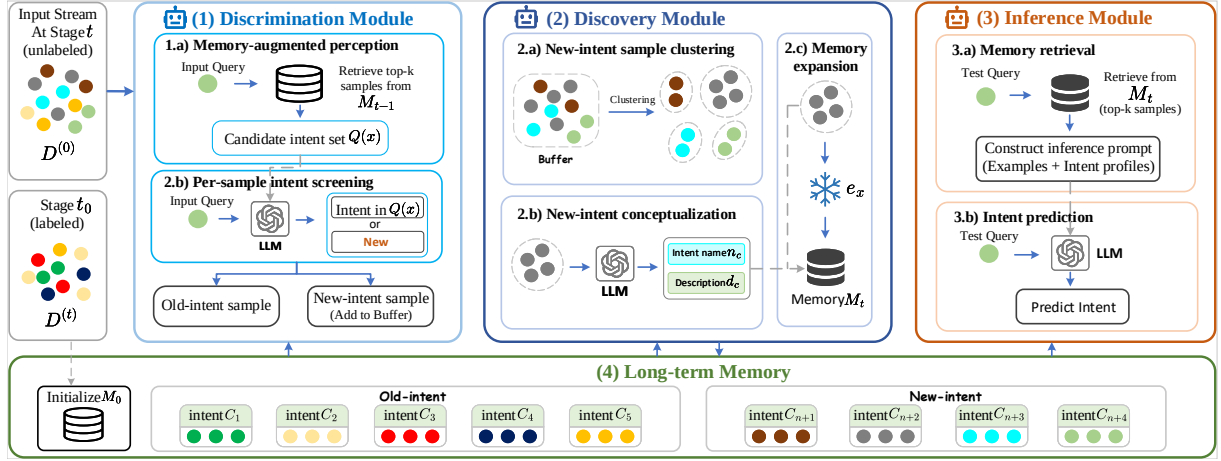


Figure 2: Overview of the proposed CIDA, containing one long-term memory and three functional modules.

- **NMI** (Normalised Mutual Information). We adopt NMI (Vinh et al., 2009) to assess the overall structural sanity of the generated clusters.
 - **ARI** (Adjusted Rand Index). We adopt ARI (Hubert and Arabie, 1985) to evaluate the pairwise clustering quality.
 - **ACC_{all}** (All-intent clustering accuracy). We adopt ACC_{all} (Cai et al., 2005) to measure the global partition accuracy across all intents.
 - **ACC_{new}** (New-intent clustering accuracy). We propose ACC_{new} to evaluate the model accuracy in discovering new-intents.
 - **B-score**. We propose B-score to evaluate the model robustness against the long-tailed distribution of new-intents’ samples.
- Retention metrics** measure the retention capability of identifying old-intents’ samples:
- **ACC_{old}** (Old-intent clustering accuracy). We propose ACC_{old} to measure historical knowledge retention for monitoring catastrophic forgetting.
 - **H-score**. We propose H-score to evaluate the balance between old-intent retention and new-intent discovery.
 - **AIA** (Average incremental accuracy). We propose AIA to evaluate the long-term performance stability across the entire data stream.

4 Continual Intent Discovery Agent

We design a **Continual Intent Discovery Agent** (CIDA) for the COSID task, which comprises a shared *long-term memory* and three functional modules, including the *discrimination*, *discovery* and *inference* module. The first two modules are used as the continual learning of the agent; While the inference module tests the learning outcome of the agent. Figure 2 illustrates the CIDA framework.

4.1 Long-term Memory

The global *Long-term Memory* (*LM*), denoted as $\mathcal{M}$, serves as the foundational knowledge base for the agent. Rather than a static storage, it evolves through a continuous expansion process: As the agent progresses through successive stages, newly discovered intents are formalized and their associated data are appended to the memory. The snapshot of LM at the end of stage t is formulated as a set $\mathcal{M}_t = \{(n_c, d_c, \mathbf{v}_c, \mathcal{E}_c)\}$, where n_c is the name of an intent c , d_c the intent description. $\mathcal{E}_c = \{(x, \mathbf{e}_x)\}$ is the exemplar pool, where x is a sample with the intent c and $\mathbf{e}_x$ the sample encoding encoded by a frozen sentence encoder. $\mathbf{v}_c = \frac{1}{|\mathcal{E}_c|} \sum_{\mathcal{E}_c} \mathbf{e}_x$ is the prototype vector of $\mathcal{E}_c$.

In the initial stage t_0 , the memory $\mathcal{M}_0$ is initialized using the labeled dataset $\mathcal{D}^{(0)}$. For each intent c in the base intent set $\mathcal{C}_0$, its n_c is derived from the intent label. Let $\mathcal{D}_c^{(0)}$ denote the set of samples in $\mathcal{D}^{(0)}$ with the intent label c . The d_c is derived from $\mathcal{D}_c^{(0)}$ via an LLM (The prompt is provided in Appendix F.1). The $\mathcal{E}_c$ contains the samples in $\mathcal{D}_c^{(0)}$ as well as the samples’ encodings.

4.2 Discrimination Module

This module processes the incoming data stream $\mathcal{D}^{(t)}$ by conducting per-sample screening against the previous-stage memory snapshot. By retrieving relevant historical knowledge, the agent determines the identity of each sample. That is, if it is an old-intent sample, the agent classifies it into an existing intent; Otherwise, the sample is included into a *temporary buffer* $\mathcal{B}$.

Memory-augmented perception. For each sample x in $\mathcal{D}^{(t)}$, the agent uses its encoding $\mathbf{e}_x$ to first

retrieve the top- k most similar examples from the pool $\{\mathcal{E}_c\}$ in the memory $\mathcal{M}$:

$$\mathcal{R}_k(x) = \text{top-}k (x', \mathbf{e}_{x'}) \in \{\mathcal{E}_c\} \cos(\mathbf{e}_x, \mathbf{e}_{x'}).$$

Collapsing the example set $\mathcal{R}_k(x)$ over the intent dimension yields a candidate intent set $\mathcal{Q}(x)$.

Per-sample intent screening. The agent employs an LLM to discriminate the sample x . The LLM takes x and the profiles of retrieved candidates $\{(n_c, d_c)\}_{c \in \mathcal{Q}(x)}$ as its input:

$$\hat{z}_x = \text{LLM}(x, \{(n_c, d_c)\}_{c \in \mathcal{Q}(x)}),$$

where the valid output space is $\mathcal{Q}(x) \cup \{\text{NEW}\}$. The LLM is instructed to output NEW if x does not match any candidate intent in $\mathcal{Q}(x)$, and the agent updates $\mathcal{B} = \mathcal{B} \cup \{x\}$; Otherwise, it outputs the name of the matched intent. The prompt is provided in Appendix F.2. To handle potential format hallucinations, if the LLM output cannot be exactly parsed into the predefined candidates, it is mapped to the valid candidate intent with the highest semantic similarity.

4.3 Discovery Module

This module processes the new-intent samples in the temporary buffer $\mathcal{B}$. The agent first groups these samples into distinct clusters. Next for each formed cluster, it generates a semantic profile (consisting of an intent name n_c and an intent description d_c) and expands the long-term memory with the samples and profile of the cluster.

New-intent sample clustering. Since the exact number of emerging new-intents is unavailable beforehand in real-world, we follow the method in NID tasks (Zhang et al., 2021) to first estimate the number of clusters. Using this estimation, the agent applies K-means clustering to group the new-intent samples in $\mathcal{B}$ into distinct clusters.

New-intent conceptualization. For each cluster, the agent then employs an LLM to conceptualize an intent name of the cluster. To avoid semantic collisions, all samples in a cluster are fed into the LLM, alongside the list of historical intents. The LLM generates a new-intent name that is semantically distinct from all existing intent names. Based on this new name and the cluster’s samples, the LLM then generates an intent description. Appendix F.3 provides the conceptualization prompt.

Memory expansion. For each cluster, the agent appends the profile (n_c, d_c) , the exemplar pool $\mathcal{E}_c$

(comprising the samples in the cluster and their corresponding encodings), and the prototype $\mathbf{v}_c$ of the cluster into the long-term memory. This expands the memory $\mathcal{M}_t$ with the updated knowledge for the next stage.

4.4 Inference Module

This module leverages the updated memory for the intent discovery application.

Memory retrieval. At inference time, the agent operates on the fully updated memory $\mathcal{M}$. For a given test query y , it retrieves the top- k most similar examples across all intents stored in the memory. The retrieved examples and their corresponding intent profiles are used to construct the *inference prompt* (Appendix F.4).

Intent prediction. The agent employs an LLM to predict the intent name for y based on the inference prompt. To alleviate LLM hallucination, its output is parsed into a specific valid intent via case-insensitive matching against all known intent names in the memory. If the output cannot match any intent name, the example y is classified as the intent of the top-1 retrieved example.

5 Experiment Results

We construct two continual learning data streams, the ClinCL from the CLINC150 dataset (Larson et al., 2019) and BankCL from the BANKING77 dataset (Casanueva et al., 2020), based on our protocol. Figure 14 in Appendix E provides an example of constructed data stream. This section presents the main results on ClinCL, and more experiment results are provided in Appendix D. Besides our CIDA, two competitors are evaluated: the CGID (Song et al., 2023b) and VBCGCD (Dai and Chauhan, 2025). Results are reported as the average over three runs. All experiment settings are detailed in Appendix C.

5.1 Impact of Sample Mixing Ratio

Table 1 compares the performance of the three methods against the sample mixing ratio α on ClinCL. The parameter α controls the mixing ratio of old-intents’ samples within the incoming data stream. As α increases, introducing a higher proportion of old-intents’ samples, all methods experience performance degradation. This is because old-intents’ samples generally interfere with the clustering and discovery of new-intents’ samples.

Among the three methods, our CIDA consistently outperforms the other two and exhibits a

Setting	Method	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
$\alpha=0$	CGID	—	—	—	86.15	59.53	75.79	88.37	89.42	74.34	80.64	72.10
	VBCGCD	88.87	0.19	18.63	93.08	79.14	85.90	72.52	85.01	88.58	79.32	92.21
	Ours	96.97	0.38	3.83	96.58	88.73	92.42	89.59	92.87	93.42	91.32	94.89
$\alpha=0.5$	CGID	—	—	—	74.15	24.47	50.16	60.95	72.69	48.81	54.11	50.47
	VBCGCD	85.83	0.47	21.82	92.07	76.00	82.71	65.67	77.20	85.97	74.14	90.41
	Ours	94.72	0.73	6.22	96.18	87.41	90.94	86.08	91.96	92.13	88.77	93.90
$\alpha=0.8$	CGID	—	—	—	63.79	15.10	26.46	29.36	60.28	25.07	24.34	27.80
	VBCGCD	65.64	3.67	36.61	89.08	67.20	74.85	41.24	66.40	80.77	53.89	87.96
	Ours	88.69	2.47	5.66	94.70	83.62	88.83	83.81	89.87	89.94	86.54	92.00

Table 1: Results on ClinCL across α (%). Each row is the mean over five incremental stages; full per-stage breakdown is reported in Table 9. $\alpha \in [0, 1]$ is the old-intent recurrence ratio. ↓: lower-is-better.

Strategy	Method	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
EXPANSION	CGID	—	—	—	75.83	26.48	52.17	67.46	71.14	49.50	56.65	50.18
	VBCGCD	78.40	0.60	29.98	91.73	74.67	83.29	62.19	78.21	86.55	72.23	91.03
	Ours	95.33	0.56	5.33	97.35	92.10	95.01	90.52	94.21	95.72	92.92	97.05
RANDOM	CGID	—	—	—	74.15	24.47	50.16	60.95	72.69	48.81	54.11	50.47
	VBCGCD	85.83	0.47	21.82	92.07	76.00	82.71	65.67	77.20	85.97	74.14	90.41
	Ours	94.72	0.73	6.22	96.18	87.41	90.94	86.08	91.96	92.13	88.77	93.90
COHESION	CGID	—	—	—	72.37	21.17	47.46	53.32	54.19	47.32	49.67	48.38
	VBCGCD	93.02	0.66	7.12	90.15	66.55	76.44	56.55	57.53	80.67	65.00	86.28
	Ours	98.05	0.66	0.72	95.32	85.17	89.66	88.14	88.11	90.80	89.13	92.72

Table 2: Results on ClinCL across intent distribution strategies (%). Each row is the mean over five incremental stages; full per-stage breakdown is reported in Table 12. Strategies follow Section 2.2. ↓: lower-is-better.

significantly slower degradation rate with the increase of α . This can be attributed to our design of decoupled discrimination and discovery. By performing the per-sample screening within an independent discrimination module backed by the long-term memory, our agent can effectively filter out the heavy load of old-intents’ samples. This ensures a purer pool of new-intent samples for the subsequent discovery module.

In contrast, with increasing α , the performance degradations of CGID and VBCGCD are more prominent (c.f. Appendix D.1, Figure 5). Lacking an explicit sample screening mechanism (where screening metrics are thus inapplicable and denoted as ‘-’), the CGID forces the mixed old-intents’ samples to participate directly in the discovery of new intents, thereby introducing severe noise into the intents’ clustering and prototyping. Although the VBCGCD incorporates a screening mechanism, its robustness degrades sharply when confronted with abundant samples of old-intents, resulting in performance deterioration.

We note that similar observations can also be

found on the BankCL data stream, and more detailed results are reported in Appendix D.1.

5.2 Impact of Intent Distribution Strategy

Table 2 presents performance comparisons across the three intent distribution strategies. The *cohesion* strategy groups semantically similar intents into the same stage; While the *expansion* strategy does the opposite, scattering similar intents across different stages. The *random* strategy lies between the two. The difference in intent distribution reveals a structural trade-off between the discrimination and discovery phase for the COSID task (c.f. Appendix D.2, Figure 8 and Figure 9). Under cohesion, the ability to separate mixed old-intent and new-intent samples improves, because the new-intents are semantically far from the old-intents in memory; However, the subsequent clustering phase becomes more challenging due to the tight pack of semantic-similar new-intents’ samples. Conversely, under expansion, screening becomes the primary bottleneck: Because new intents are highly similar to old ones, it is inherently difficult to distinguish

Setting	Method	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
$s=0.5$	CGID	—	—	—	73.38	23.35	48.85	61.49	74.41	47.24	53.32	48.94
	VBCGCD	82.54	0.62	24.29	90.87	72.06	78.93	59.67	73.99	83.08	68.34	88.80
	Ours	95.52	0.70	4.62	95.96	87.36	91.13	85.03	88.32	92.43	88.51	94.31
$s=0.8$	CGID	—	—	—	72.75	21.74	47.29	46.12	62.68	46.76	45.16	48.80
	VBCGCD	79.53	0.94	27.05	90.44	71.07	78.24	54.43	69.96	82.71	65.09	88.58
	Ours	93.92	1.06	5.36	94.45	81.31	87.36	75.81	80.06	89.50	81.72	92.42
$s=1.0$	CGID	—	—	—	73.63	23.57	49.58	55.18	62.35	49.27	51.36	54.07
	VBCGCD	73.18	1.99	30.67	89.53	68.63	76.61	46.60	65.30	81.66	59.05	88.07
	Ours	90.36	1.31	8.52	94.34	81.35	86.07	72.32	79.55	88.43	79.49	91.76

Table 3: Results on ClineCL across the long-tail exponent s (%). Each row is the mean over five incremental stages; full per-stage breakdown is in Table 15. ↓: lower-is-better.

Setting	Method	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
$K_{\text{init}}=5$	CGID	—	—	—	65.79	18.53	40.96	29.12	53.50	41.52	32.02	31.92
	VBCGCD	66.56	2.57	39.70	81.94	54.93	67.99	46.55	65.04	71.95	56.10	78.06
	Ours	76.87	4.46	11.39	91.90	69.79	83.12	75.19	84.85	85.07	79.49	88.42
$K_{\text{init}}=10$	CGID	—	—	—	72.38	24.10	49.29	44.36	60.66	49.68	45.76	45.49
	VBCGCD	78.12	1.22	27.84	88.93	67.69	76.55	55.58	69.82	80.60	65.10	86.09
	Ours	84.82	2.91	8.46	93.27	77.83	86.37	80.01	87.88	88.18	83.45	91.03
$K_{\text{init}}=30$	CGID	—	—	—	74.15	24.47	50.16	60.95	72.69	48.81	54.11	50.47
	VBCGCD	85.83	0.47	21.82	92.07	76.00	82.71	65.67	77.20	85.97	74.14	90.41
	Ours	94.72	0.73	6.22	96.18	87.41	90.94	86.08	91.96	92.13	88.77	93.90
$K_{\text{init}}=50$	CGID	—	—	—	69.70	18.73	42.72	59.72	72.34	40.20	47.69	43.46
	VBCGCD	86.91	0.30	21.11	92.72	77.47	83.38	62.80	73.89	86.93	72.54	91.63
	Ours	95.81	0.72	4.60	96.37	89.19	92.90	90.52	93.59	93.68	91.97	94.98

Table 4: Results on ClineCL across the base sample size K_{init} (%). Each row is the mean over five incremental stages; full per-stage breakdown is in Table 18. ↓: lower-is-better.

between them, leading to more discrimination failures. Even facing these challenges, our CIDA still outperforms the other two in almost all settings.

5.3 Impact of Sample Distribution

Table 3 compares the performance across different long-tail distribution settings, controlled by the Zipf exponent s . As s increases from 0.5 to 1.0, the sample distribution becomes increasingly skewed: A few head new-intents account for most data volume in each stage, whereas tail new-intents face severe data sparsity. Consequently, with increasing sample distribution skewness (larger s), all methods experience a noticeable performance drop, particularly in discovery metrics such as ACC_{new} and B -score. The methods struggle to form robust clusters for tail intents due to severe sample scarcity, making them prone to being overshadowed by head intents. Our CIDA outperforms the competitors across all s settings in almost all performance met-

rics, demonstrating its superior robustness in handling long-tail distributed data streams.

5.4 Impact of Number of New-Intent Samples

Table 4 compares the performance against the per-new-intent sample budget K_{init} in each stage. Generally, more samples of a particular intent (i.e., a larger K_{init}) can help clustering boundary-clearer clusters, thereby improving new-intent discovery. As K_{init} decreases, all methods experience performance degradation, because data sparsity hinders the clustering of new-intents samples. Despite this, our CIDA outperforms the baselines in almost all metrics. This robustness is attributed to two factors: the long-term memory enables accurate sample screening with minimal evidence, and the LLM-driven discovery module effectively conceptualizes sparse clusters by leveraging external knowledge rather than relying purely on limited data.

top- k	D -score	ARI	ACC _{all}	ACC _{new}	ACC _{old}	H -score	AIA
$k=3$	94.43	86.75	91.11	87.36	92.16	89.58	94.03
$k=5$	95.19	88.32	92.59	91.36	93.32	92.19	94.67
$k=10$	95.44	88.56	92.92	92.26	93.62	92.77	95.02
$k=20$ (default)	94.72	87.41	90.94	86.08	92.13	88.77	93.90

Table 5: Sensitivity of CIDA to the retrieval depth k on ClinCL (%), reported as the mean over the five incremental stages t_1 – t_5 under the default configuration. The best result in each column is in **bold**.

Backbone LLM	D -score	ARI	ACC _{new}	ACC _{old}	H -score
Qwen3.5-9B (default)	94.72	87.41	86.08	92.13	88.77
Qwen3.5-4B	94.83	88.77	86.62	93.76	90.02
Qwen3.5-27B	94.93	89.60	86.59	95.07	90.38

Table 6: Impact of the backbone LLM scale on ClinCL (%), reported as the mean over the five incremental stages t_1 – t_5 under the default configuration.

5.5 Sensitivity to Retrieval Depth

Table 5 compares the performance of CIDA across different retrieval depths $k \in \{3, 5, 10, 20\}$ on ClinCL under the default configuration, where k controls how many memory exemplars are provided to the LLM during the per-sample screening and the inference. The performance improves as k grows from 3 to 10 and then saturates, with only marginal fluctuations between $k=10$ and $k=20$. Overall, CIDA is insensitive to the choice of k , and we adopt $k=20$ as the default.

5.6 Impact of the Backbone LLM

Table 6 compares the performance of CIDA using backbone LLMs of different scales, where we replace the LLMs, i.e., Qwen3.5-4B, Qwen3.5-9B, and Qwen3.5-27B, while keeping all other settings fixed. This examines whether the effectiveness of CIDA merely stems from the capacity of a large backbone LLM. The performance of CIDA is largely insensitive to the number of backbone parameters; even the smallest 4B model attains performance comparable to, and on some metrics slightly better than, the 9B and 27B models. This demonstrates that the advantage of CIDA does not primarily come from the model scale, nor does it rely on a larger-parameter LLM. We attribute this robustness to the fact that CIDA decomposes intent discovery into fine-grained, memory-grounded per-sample judgments that are relatively easy for LLMs to reason about, thereby imposing only mild requirements on the backbone capacity.

6 Conclusion

This paper has tried to fill the research gap in life-long intent discovery for real-world systems. We have formulated the COSID task, designed a protocol with flexible configurations to construct data streams of different difficulty levels for the task, and established a set of performance evaluation metrics in continual learning scenarios. Furthermore, we have proposed the CIDA for the task. Experiments on the constructed data streams following our protocol have validated the protocol expressiveness and method superiority. We expect that the presented protocol and method can facilitate subsequent advancements in this field.

Limitations

A fundamental challenge in any sequential lifelong learning system is the compounding effect of historical errors. As new intents are continuously discovered and added to the system knowledge base, any earlier errors, such as sample intent boundaries or misclassified samples, are inevitably retained. Over subsequent stages, these errors can propagate and accumulate, gradually polluting the system knowledge base and interfering with the discovery of future intents.

Although our proposed method significantly mitigates this issue and exhibits superior robustness compared to existing baselines, completely eliminating error cascades in endless data streams remains an unresolved challenge. We hope our benchmark can serve as a stepping stone, and more efforts from the community are deeply needed to push the boundaries of more robust lifelong intent discovery.

References

- D. Cai, X. He, and J. Han. 2005. [Document clustering using locality preserving indexing](#). *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637.
- Qian Cao, Deyi Xiong, Qinlong Wang, and Xia Peng. 2022. [CoCoID: Learning contrastive representations and compact clusters for semi-supervised intent discovery](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 226–236, Abu Dhabi, UAE. Association for Computational Linguistics.
- Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. [Efficient intent detection with dual sentence encoders](#). In *Proceedings of the 2nd Workshop on Natural Language*

- Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.
- Peng Chen and Bang Wang. 2025. [On collaborating small and large models for few-shot intent detection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13937–13953, Suzhou, China. Association for Computational Linguistics.
- Hao Dai and Jagmohan Chauhan. 2025. Continual generalized category discovery: learning and forgetting from a bayesian perspective. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML’25. JMLR.org.
- Lu Fan, Jiashu Pu, Rongsheng Zhang, and Xiao-Ming Wu. 2024. [LANID: LLM-assisted new intent discovery](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10110–10116, Torino, Italia. ELRA and ICCL.
- Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. [Continual learning for text classification with information disentanglement based regularization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2736–2746, Online. Association for Computational Linguistics.
- Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.
- Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. [An evaluation dataset for intent classification and out-of-scope prediction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.
- Ting-En Lin and Hua Xu. 2019. [Deep unknown intent detection with margin loss](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5491–5496, Florence, Italy. Association for Computational Linguistics.
- Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. [Discovering new intents via constrained deep adaptive clustering with cluster refinement](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8360–8367.
- Han Liu, Junjie Sun, Xiaotong Zhang, and Hongyang Chen. 2024. [New intent discovery with multi-view clustering](#). In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12381–12385.
- Shengkun Ma, Jiale Han, Yi Liang, and Bo Cheng. 2024. [Making pre-trained language models better continual](#)
- [few-shot relation extractors](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10970–10983, Torino, Italia. ELRA and ICCL.
- Sahisnu Mazumder and Bing Liu. 2024. Continual learning for task-oriented dialogue systems. In *Lifelong and Continual Learning Dialogue Systems*, pages 127–151. Springer.
- Yutao Mou, Keqing He, Yanan Wu, Pei Wang, Jingang Wang, Wei Wu, Yi Huang, Junlan Feng, and Weiran Xu. 2022. [Generalized intent discovery: Learning from open world dialogue system](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 707–720, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Yutao Mou, Xiaoshuai Song, Keqing He, Chen Zeng, Pei Wang, Jingang Wang, Yunsen Xian, and Weiran Xu. 2023. [Decoupling pseudo label disambiguation and representation learning for generalized intent discovery](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9661–9675, Toronto, Canada. Association for Computational Linguistics.
- Mrinal Rawat, Hithesh Sankararaman, and Victor Barres. 2023. Controllable discovery of intents: Incremental deep clustering using semi-supervised contrastive learning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–803.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Grzegorz Rypeść, Daniel Marczak, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. 2025. Category adaptation meets projected distillation in generalized continual category discovery. In *Computer Vision – ECCV 2024*, pages 320–337, Cham. Springer Nature Switzerland.
- Bhavuk Singhal, Ashim Gupta, V P Shivasankaran, and Amrith Krishna. 2023. [IntenDD: A unified contrastive learning approach for intent detection and discovery](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14204–14216, Singapore. Association for Computational Linguistics.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. [Mpnet: Masked and permuted pre-training for language understanding](#). *Preprint*, arXiv:2004.09297.

- Xiaoshuai Song, Keqing He, Pei Wang, Guanting Dong, Yutao Mou, Jingang Wang, Yunsen Xian, Xunliang Cai, and Weiran Xu. 2023a. [Large language models meet open-world intent discovery and recognition: An evaluation of ChatGPT](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10291–10304, Singapore. Association for Computational Linguistics.
- Xiaoshuai Song, Yutao Mou, Keqing He, Yueyan Qiu, Jinxu Zhao, Pei Wang, and Weiran Xu. 2023b. [Continual generalized intent discovery: Marching towards dynamic and open-world intent recognition](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4370–4382, Singapore. Association for Computational Linguistics.
- Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. 2022. Generalized category discovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7492–7501.
- Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080.
- Hossam Zawbaa, Wael Rashwan, Sourav Dutta, and Haytham Assem. 2024. [Improved out-of-scope intent classification with dual encoding and threshold-based re-classification](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8708–8718, Torino, Italia. ELRA and ICCL.
- Li-Ming Zhan, Haowen Liang, Bo Liu, Lu Fan, Albert Y.S. Lam, and Xiao-Ming Wu. 2021. [Out-of-distribution intent detection with self-supervision and discriminative training](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3521–3532, Online. Association for Computational Linguistics.
- Feng Zhang, Wei Chen, Fei Ding, Meng Gao, Tengjiao Wang, Jiahui Yao, and Jiabin Zheng. 2024a. [From discrimination to generation: Low-resource intent detection with language model instruction tuning](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10167–10183, Bangkok, Thailand. Association for Computational Linguistics.
- Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021. [Discovering new intents with deep aligned clustering](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14365–14373.
- Haode Zhang, Haowen Liang, Liming Zhan, Albert Y.S. Lam, and Xiao-Ming Wu. 2023. [Revisit few-shot intent classification with PLMs: Direct fine-tuning vs. continual pre-training](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11105–11121, Toronto, Canada. Association for Computational Linguistics.
- Shun Zhang, Yan Chaoran, Jian Yang, Jiaheng Liu, Ying Mo, Jiaqi Bai, Tongliang Li, and Zhoujun Li. 2024b. [Towards real-world scenario: Imbalanced new intent discovery](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3949–3963, Bangkok, Thailand. Association for Computational Linguistics.
- Yuwei Zhang, Haode Zhang, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2022. [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)*, pages 256–269, Dublin, Ireland. Association for Computational Linguistics.

A Related Work

Intent Detection. Intent detection aims to classify user queries into predefined intent categories (Lin and Xu, 2019; Zhang et al., 2024a). However, real-world dialogue systems inevitably encounter queries beyond the predefined intent scope. To address this issue, early studies mainly focused on *out-of-scope* (OOS) detection (Larson et al., 2019; Zhan et al., 2021; Zawbaa et al., 2024), which detects whether a query belongs to unknown intents. To further conceptualize unknown intents, *new intent discovery* (NID) was proposed to group unlabeled queries into latent intent categories (Zhang et al., 2021; Fan et al., 2024; Zhang et al., 2024b). Existing methods typically leverage clustering-based representation learning and semantic structure modeling to improve intent discovery performance (Lin et al., 2020; Zhang et al., 2022; Liu et al., 2024).

Subsequent works further explore semi-supervised intent discovery (Cao et al., 2022), contrastive representation learning (Rawat et al., 2023), and open-world generalized intent discovery (Mou et al., 2022, 2023; Song et al., 2023a). Although these methods achieve promising results, they generally rely on static dataset partitions and assume that all emerging intents are simultaneously accessible, overlooking the evolving and streaming nature of real-world systems.

Continual Learning. In the continual learning field, the research focus is mainly on how to mitigate catastrophic forgetting through replay mechanisms, prototype learning, and representation regularization (Mazumder and Liu, 2024; Ma et al., 2024; Huang et al., 2021). These studies have established a good foundation for some task-oriented new category/intent discovery. In computer vision, Generalized Category Discovery (GCD) (Vaze et al., 2022) and Continual Generalized Category Discovery (CGCD) (Rypešć et al., 2025; Dai and Chauhan, 2025) extend category discovery toward continual open-world settings, where novel categories emerge incrementally over time. Inspired by these studies, *Continual Generalized Intent Discovery* (CGID) (Song et al., 2023b) introduces a continual intent discovery task, requiring systems to continuously discover new intents. However, the CGID adopts an idealized premise, wherein the streaming data of each stage exclusively comprises samples only from new intents.

We note that in real-world systems, intent evo-

lution is a highly dynamic process. Streaming queries naturally comprise a complex mixture of both historical intents and newly emerging ones, often arriving in non-uniform distributions. There is a critical need for a more realistic task definition and a data stream construction protocol to emulate diverse real-world scenarios. This paper aims to bridge this research gap in the field of new intent discovery.

Compared with related work, **our contributions** are summarized as follows.

- Propose a new Continual Open-Stream Intent Discovery (COSID) task with a more general setting for real-world systems.
- Design an *open-stream dataset construction protocol* with flexible configurations to set different difficulty levels for COSID.
- Devise a set of COSID performance evaluation metrics to evaluate system ability from three aspects.
- Construct two sets of data streams (CLincCL and BankCL) for COSID from existing static intent datasets.
- Design a *Continual Intent Discover Agent* (CIDA) for the COSID task, with superior performance compared with two baselines.
- Conduct extensive experiments to evaluate not only the performance of the methods, but also the expressiveness of our protocol.

In addition, we publish our protocol implementation, constructed data streams and agent implementation ¹.

B Evaluation Metrics

We propose to evaluate the COSID task from three aspects: discrimination capability of dividing new intents from old ones, discovery capability of clustering new intents, and retention capability of identifying old intents. To formally define these metrics, let $\mathcal{D}_{\text{test}}^{(t)}$ denote the cumulative test set at stage t . For each test sample $(x_i, y_i) \in \mathcal{D}_{\text{test}}^{(t)}$, $y_i \in \mathcal{C}_{\leq t}$ represents its ground-truth intent label. We partition the test set into a novelty subset $\mathcal{N}^{(t)} = \{(x_i, y_i) \in \mathcal{D}_{\text{test}}^{(t)} \mid y_i \in \mathcal{C}_t\}$ and a historical subset $\mathcal{O}^{(t)} = \{(x_i, y_i) \in \mathcal{D}_{\text{test}}^{(t)} \mid y_i \in \mathcal{C}_{< t}\}$. Let

¹<https://github.com/hustchenpeng/COSID>

$z_i \in \{\text{NEW}, \text{OLD}\}$ denote the ground-truth novelty label of sample x_i , where $z_i = \text{NEW}$ if $y_i \in \mathcal{C}_t$, and $z_i = \text{OLD}$ otherwise. Correspondingly, let $\hat{z}_i \in \{\text{NEW}, \text{OLD}\}$ denote the model’s predicted novelty label for sample x_i , where $\hat{z}_i = \text{NEW}$ if the model assigns x_i to an intent outside $\mathcal{C}_{<t}$, and $\hat{z}_i = \text{OLD}$ otherwise.

B.1 Discrimination Metrics

D-score. This metric evaluates the model’s ability to distinguish between new and old intents, a critical capability that directly impacts the subsequent discovery of new intents. Treating NEW as the positive class, we first define True Positives (TP), False Positives (FP), and False Negatives (FN) over the test set $\mathcal{D}_{\text{test}}^{(t)} = \mathcal{N}^{(t)} \cup \mathcal{O}^{(t)}$:

$$\text{TP} = \sum_{x_i \in \mathcal{D}_{\text{test}}^{(t)}} \mathbb{I}(z_i = \text{NEW} \wedge \hat{z}_i = \text{NEW}), \quad (7)$$

$$\text{FP} = \sum_{x_i \in \mathcal{D}_{\text{test}}^{(t)}} \mathbb{I}(z_i = \text{OLD} \wedge \hat{z}_i = \text{NEW}), \quad (8)$$

$$\text{FN} = \sum_{x_i \in \mathcal{D}_{\text{test}}^{(t)}} \mathbb{I}(z_i = \text{NEW} \wedge \hat{z}_i = \text{OLD}), \quad (9)$$

where $\mathbb{I}(\cdot)$ is the indicator function. The precision (P) and recall (R) are computed as $P = \text{TP}/(\text{TP} + \text{FP})$ and $R = \text{TP}/(\text{TP} + \text{FN})$. The D-score is calculated as the binary $F1$ score:

$$\text{D-score} = \frac{2 \cdot P \cdot R}{P + R}. \quad (10)$$

OSR (Over-screening rate). This metric calculates the proportion of old samples in $\mathcal{O}^{(t)}$ that are mislabelled as new, serving to evaluate the degree to which the model excessively flags old intent samples as new intents. When the screening mechanism is overly aggressive, it mistakenly treats old-intent samples as new-intent ones, thereby contaminating the downstream clustering and naming of genuinely new intents. Formally, it is computed as:

$$\text{OSR} = \frac{1}{|\mathcal{O}^{(t)}|} \sum_{x_i \in \mathcal{O}^{(t)}} \mathbb{I}(\hat{z}_i = \text{NEW}). \quad (11)$$

USR (Under-screening rate). This metric calculates the proportion of new-intent samples in $\mathcal{N}^{(t)}$ that are mislabelled as old, serving to evaluate the degree to which the model excessively flags new-intent samples as old intents. When the screening mechanism is overly conservative, it mistakenly

treats new-intent samples as old ones, thereby suppressing the discovery and identification of new intents. Formally, it is computed as:

$$\text{USR} = \frac{1}{|\mathcal{N}^{(t)}|} \sum_{x_i \in \mathcal{N}^{(t)}} \mathbb{I}(\hat{z}_i = \text{OLD}). \quad (12)$$

B.2 Discovery Metrics

For the following discovery metrics, let $\hat{y}_i$ denote the model cluster assignment for sample x_i . Since cluster IDs may not strictly match the ground-truth intent labels, we apply the Hungarian algorithm to find the optimal one-to-one mapping π between the predicted clusters and the ground-truth intents that maximizes the overall accuracy.

NMI (Normalised Mutual Information). This metric evaluates the overall structural sanity of the generated intent spaces. Let Y represent the true label distribution and $\hat{Y}$ represent the predicted cluster distribution. NMI measures the shared information between the predicted partition and the ground-truth partition, normalized by their entropies $H(Y)$ and $H(\hat{Y})$:

$$\text{NMI}(Y, \hat{Y}) = \frac{2 \cdot I(Y; \hat{Y})}{H(Y) + H(\hat{Y})}, \quad (13)$$

where $I(Y; \hat{Y})$ is the mutual information. It effectively penalizes the *over-screening* structural anomaly—where a single intent is dispersed across multiple clusters—which ACC alone might mask.

ARI (Adjusted Rand Index). This metric evaluates the clustering quality from a pair-wise alignment perspective. Let RI be the Rand Index, defined as the ratio of sample pairs that are either grouped together correctly or separated correctly in both Y and $\hat{Y}$. ARI corrects for chance grouping by incorporating the expected RI:

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}) - \mathbb{E}[\text{RI}]}. \quad (14)$$

While NMI is sensitive to over-screening, ARI explicitly penalizes the *over-merging* structural anomaly, where multiple distinct intents are collapsed into a single large cluster.

ACC_{all} (All-intent clustering accuracy). This metric evaluates the system capacity to reconstruct the overall intent space, serving to reflect the global partition performance across both old and new intents. It reflects the system recognition capability

over the cumulative intent space, calculated via the global Hungarian alignment π over the full test dataset $\mathcal{D}_{\text{test}}^{(t)}$:

$$\text{ACC}_{\text{all}} = \frac{1}{|\mathcal{D}_{\text{test}}^{(t)}|} \sum_{x_i \in \mathcal{D}_{\text{test}}^{(t)}} \mathbb{I}(\pi(\hat{y}_i) = y_i). \quad (15)$$

ACC_{new} (New-intent clustering accuracy). This metric evaluates the system discovery capability for new intents, serving to assess its new-intent recognition capability when facing unseen intents. It reflects the clustering quality for genuinely new-intent samples, defined as the Hungarian-aligned accuracy evaluated exclusively on the novelty subset $\mathcal{N}^{(t)}$:

$$\text{ACC}_{\text{new}} = \frac{1}{|\mathcal{N}^{(t)}|} \sum_{x_i \in \mathcal{N}^{(t)}} \mathbb{I}(\pi(\hat{y}_i) = y_i). \quad (16)$$

B-score. Standard accuracy metrics are easily dominated by high-frequency head intents, which often obscure a system failure in discovering sparse tail categories. To address this, B-score is devised to evaluate the model robustness against the long-tailed distribution of new intents. We partition the new intent set $\mathcal{C}_t$ into a high-support head subset $\mathcal{C}_{\text{head}}$ and a low-support tail subset $\mathcal{C}_{\text{tail}}$. We run a separate Hungarian assignment $\pi_{\mathcal{N}}$ on the new-intent subset $\mathcal{N}^{(t)}$ to pair each $c \in \mathcal{C}_t$ with at most one predicted cluster, and compute the micro- $F1$ score within each half, denoted as $F1_{\text{head}}$ and $F1_{\text{tail}}$, with unmatched intents contributing zero to strictly penalize the collapse of tail intents. The B-score is defined as their harmonic mean:

$$\text{B-score} = \frac{2 \cdot F1_{\text{head}} \cdot F1_{\text{tail}}}{F1_{\text{head}} + F1_{\text{tail}}}. \quad (17)$$

Because this B-score collapses to zero if either head or tail subset is entirely abandoned, it can effectively track the performance of head-intent and tail-intent discovery.

B.3 Retention Metrics

ACC_{old} (Old-intent clustering accuracy). This metric measures the system retention capability for historical knowledge, serving to monitor whether the system suffers from catastrophic forgetting as the data stream evolves. It reflects the recognition accuracy for test queries with old intents, defined as the Hungarian-aligned accuracy evaluated exclusively on the historical subset $\mathcal{O}^{(t)}$:

$$\text{ACC}_{\text{old}} = \frac{1}{|\mathcal{O}^{(t)}|} \sum_{x_i \in \mathcal{O}^{(t)}} \mathbb{I}(\pi(\hat{y}_i) = y_i). \quad (18)$$

H-score. This metric evaluates the balance between old intent retention and new intent discovery. It is introduced to evaluate the overall performance of the model across both old-intent detection and new-intent discovery. Formally, it is expressed as the harmonic mean of ACC_{old} and ACC_{new} :

$$\text{H-score} = \frac{2 \cdot \text{ACC}_{\text{old}} \cdot \text{ACC}_{\text{new}}}{\text{ACC}_{\text{old}} + \text{ACC}_{\text{new}}}. \quad (19)$$

Because the harmonic mean is strictly dominated by the smaller of its two components, a model cannot cheat a high H -score by trading performance in one aspect to mask deficiencies in the other.

AIA (Average incremental accuracy). This metric evaluates the long-term stability and performance trajectory of the system across the entire data stream, serving to capture the holistic historical trend of knowledge retention rather than a single static snapshot. A snapshot at any single stage cannot tell how a system ages with the evolving stream. Writing $M_t = \text{ACC}_{\text{old}}^{(t)}$ for $t \geq 1$ (the retention ability at stage t , read under the same π described above), AIA reports the running mean:

$$\text{AIA}_t = \frac{1}{t} \sum_{s=1}^t M_s, \quad (20)$$

which integrates the entire trajectory rather than only its endpoint. It is therefore a measure robust to a late-stage rebound that would inflate an endpoint measurement.

C Detailed Experiment Settings

Source datasets. We construct two data streams from two widely used intent-discovery datasets: **CLINC150** (Larson et al., 2019) and **BANKING77** (Casanueva et al., 2020). Their basic statistics are summarised in Table 7.

Dataset	CLINC150	BANKING77
# intents	150	77
# domains	10	1
# train samples	18,000	10,003
# test samples	4,500	3,080

Table 7: Statistics of the two source datasets.

Stream construction protocol. Each stream is constructed by the protocol described in Section 2 with $T=5$ incremental stages and a base-intent ratio $\rho=0.5$, yielding $N_{\text{base}}=75$ base intents and 75

residual intents for ClinCL, and $N_{\text{base}}=38$ base intents and 39 residual intents for BankCL. According to our decreasing schedule, the per-stage new-intent counts are $\{25, 20, 15, 10, 5\}$ on ClinCL and $\{14, 11, 7, 5, 2\}$ on BankCL. In each dimensional sweep, only one parameter is adjusted while the rest remain set to default values.

- Intent mixing ratio $\alpha \in \{0, 0.5, 0.8\}$ (default $\alpha=0.5$).
- Partitioning strategy including random, cohesion and expansion (default *random*).
- Long-tail exponent $s \in \{0, 0.5, 0.8, 1.0\}$ (default $s=0$, i.e. uniform).
- Base sample size $K_{\text{init}} \in \{5, 10, 30, 50\}$ (default $K_{\text{init}}=30$).

Backbone model. We use the open-source Qwen3.5-9B as the backbone LLM for the agent. All semantic encoders used in this work (for retrieval, clustering, and similarity-based label mapping) are instantiated with all-mpnet-base-v2 (Reimers and Gurevych, 2019; Song et al., 2020).

Competitors and evaluation protocol. We compare CIDA with two representative baselines. CGID (Song et al., 2023b) is the most recent continual generalized intent discovery method. VBCGCD (Dai and Chauhan, 2025) is proposed in computer vision for continual generalized category discovery; we adapt it to the COSID task with its officially released implementation and hyperparameters. All the three methods are evaluated under the same configuration of protocol parameters, so that the generated ClinCL and BankCL streams are identical across methods. Discovery metrics are computed via a single global Hungarian assignment between predicted clusters and ground-truth intents, following the definitions in Appendix B.

D More Experiment Results

The main body reports the comparison results on ClinCL. Here we provide the main results on BankCL, together with the full per-stage results of CIDA on both datasets, so that each dimension can be inspected at finer granularity.

D.1 Impact of Intent Mixing Ratio

This subsection presents experiment results on the parameter α , which controls the mixing ratio of old-intents’ samples within the incoming data stream (Section 5.1).

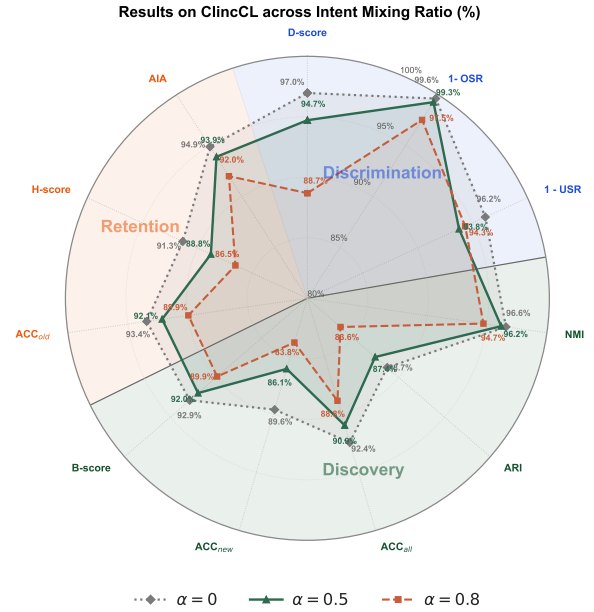


Figure 3: Radar comparison across all metrics on ClinCL under varying α . Click [Section D.1](#) to jump back.

- Table 8: Main results on BankCL. Comparison among the three methods at each α , paralleling the ClinCL main results, i.e., each averaged over all stages.
- Table 9: Per-Stage results on ClinCL. Reports CIDA performance at the base stage t_0 and at each incremental stage t_1-t_5 on ClinCL under each α .
- Table 10: Per-Stage results on BankCL. Reports CIDA performance at the base stage t_0 and at every incremental stage t_1-t_5 on BankCL under each α .
- Figure 3: Radar plot of CIDA on ClinCL. It plots the main results of all performance metrics of CIDA under different choices of α on ClinCL. Note that we change the USR and OSR as 1-USR and 1-OSR, respectively, in this radar plot. The result closer to the circumference of the circle indicates better performance. It can be observed that the smaller the value of α , the better the performance of the CIDA, indicating that the mixing ratio can control the difficulty level of the COSID task (c.f., Discussion in Section 5.1).
- Figure 4: Radar plot on BankCL. It plots the main results of all performance metrics

Setting	Method	Discrimination			Discovery					Retention		
		D -score	OSR $\downarrow$	USR $\downarrow$	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
$\alpha=0$	CGID	—	—	—	66.95	31.34	54.99	65.49	67.56	53.61	58.20	50.80
	VBCGCD	82.30	0.79	24.41	83.08	63.69	76.66	68.15	80.71	79.38	72.34	83.72
	Ours	88.09	0.60	17.55	89.70	75.18	83.57	70.84	81.32	86.03	77.17	89.51
$\alpha=0.5$	CGID	—	—	—	52.97	11.66	34.82	49.09	61.86	34.19	37.98	34.97
	VBCGCD	77.37	1.04	30.07	81.15	59.22	71.47	49.89	56.85	75.66	59.09	81.58
	Ours	80.19	1.25	25.64	88.62	73.86	82.17	67.01	81.24	84.57	74.09	88.23
$\alpha=0.8$	CGID	—	—	—	41.48	7.66	19.76	13.12	52.92	19.85	13.28	21.61
	VBCGCD	61.06	3.20	43.45	78.40	53.34	65.02	35.58	52.30	70.41	46.90	78.23
	Ours	76.87	3.23	23.60	86.85	69.57	80.12	62.30	75.86	82.80	70.37	85.83

Table 8: Results on BankCL across α (%). Each row is the mean over incremental stages t_1 – t_5 ; full per-stage breakdown is reported in Table 10. $\alpha \in [0, 1)$ is the old-intent recurrence ratio. $\downarrow$: lower-is-better. Click [Section D.1](#) to jump back.

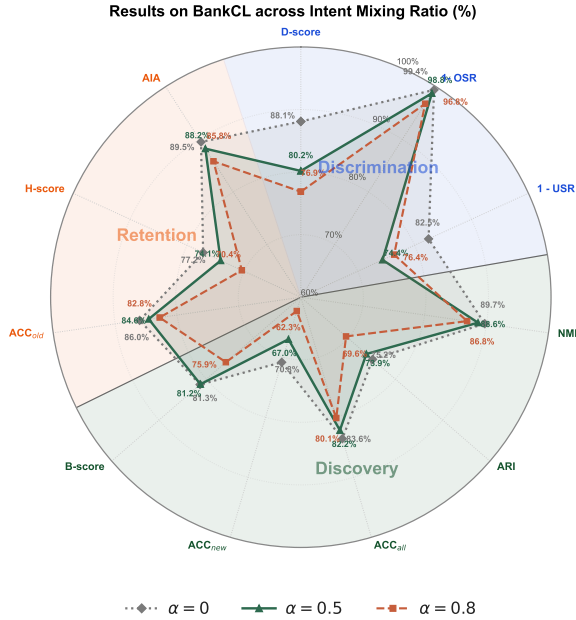


Figure 4: Radar comparison across all metrics on BankCL under varying α . Click [Section D.1](#) to jump back.

of CIDA under difference choices of α on BankCL.

- Figure 5: Performance degradation rates of the main results on ClinicCL. It plots the performance degradation rates of each method on choosing $\alpha = 0.5, 0.8$ over choosing $\alpha = 0.0$. It can be observed that our CIDA is more robust than the other two methods, i.e., with much smaller performance degradation rate when choosing larger value of α

D.2 Impact of Intent Distribution Strategy

This subsection presents experiment results on the partitioning strategy (*random*, *cohesion*, *expan-*

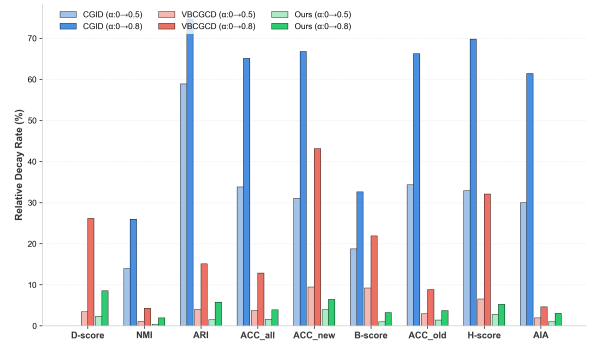


Figure 5: Relative decay rate (%) of various evaluation metrics on the ClinicCL dataset for three methods. Click [Section D.1](#) to jump back.

sion), which decides how semantically related intents are scattered across stages (Section 5.2).

- Table 11: Main results on BankCL. Comparison among the three methods under each new-intent distribution strategy, paralleling the ClinicCL main results.
- Table 12: Per-Stage results on ClinicCL. Reports CIDA performance at the base stage t_0 and at each incremental stage t_1 – t_5 on ClinicCL under each new-intent distribution strategy.
- Table 13: Per-Stage results on BankCL. Reports CIDA’s performance at the base stage t_0 and at every incremental stage t_1 – t_5 on BankCL under each strategy.
- Figure 6: Radar plot on ClinicCL. It plots the main results of all performance metrics of CIDA under the three new-intent distribution strategies on ClinicCL. It can be observed

Setting	Stage	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
init (shared)	t_0	—	—	—	—	—	97.91	—	—	97.91	—	—
$\alpha=0$	t_1	96.23	0.89	4.80	96.65	90.29	93.93	84.40	87.54	97.11	90.31	97.11
	t_2	98.25	0.30	2.00	96.65	88.84	92.39	83.33	86.70	94.20	88.43	95.66
	t_3	93.88	0.50	8.00	96.42	87.99	91.75	88.22	95.71	92.19	90.16	94.50
	t_4	98.52	0.20	0.33	96.52	88.12	91.91	96.00	96.16	91.60	93.75	93.78
	t_5	97.96	0.00	4.00	96.63	88.39	92.11	96.00	98.25	91.98	93.95	93.42
	<i>avg</i>	96.97	0.38	3.83	96.58	88.73	92.42	89.59	92.87	93.42	91.32	94.89
$\alpha=0.5$	t_1	95.36	1.78	4.00	96.45	88.62	92.50	79.47	84.82	96.84	87.30	96.84
	t_2	94.90	0.97	5.33	96.31	87.83	90.89	82.17	88.21	92.63	87.09	94.74
	t_3	95.39	0.42	5.78	96.01	86.83	90.27	89.11	94.87	90.42	89.76	93.30
	t_4	96.91	0.42	0.67	96.08	87.13	90.64	99.00	99.33	90.02	94.30	92.48
	t_5	91.04	0.05	15.33	96.08	86.64	90.40	80.67	92.56	90.74	85.41	92.13
	<i>avg</i>	94.72	0.73	6.22	96.18	87.41	90.94	86.08	91.96	92.13	88.77	93.90
$\alpha=0.8$	t_1	91.33	4.84	3.73	95.60	86.95	92.17	83.07	85.97	95.20	88.72	95.20
	t_2	91.26	2.80	4.33	94.71	84.07	88.67	77.67	80.64	90.87	83.75	93.03
	t_3	87.70	2.08	8.89	94.41	82.46	87.56	81.33	89.22	88.33	84.69	91.47
	t_4	87.96	1.90	1.33	94.36	82.28	87.91	98.33	98.99	87.14	92.40	90.38
	t_5	85.17	0.74	10.00	94.40	82.34	87.84	78.67	94.56	88.16	83.14	89.94
	<i>avg</i>	88.69	2.47	5.66	94.70	83.62	88.83	83.81	89.87	89.94	86.54	92.00

Table 9: Per-stage results on ClincCL across α (%). $\alpha \in [0, 1]$ is the old-intent recurrence ratio. ↓: lower-is-better.

that different strategies have different impacts. Generally, the *expansion* strategy results in better discovery and retention performance; While the *cohesion* strategy results in better discrimination performance (c.f., Discussion in Section 5.2)

- Figure 7: Radar plot on BankCL. It plots the main results of all performance metrics of CIDA under the three new-intent distribution strategies on BankCL.
- Figure 8: t-SNE projections of the three strategies on ClincCL. Visualizes how *cohesion*, *expansion*, and *random* scatter intents across stages. The closeness of two points (intents) in the figure indicates that they are more semantically similar. It can be observed that the *cohesion* strategy tends to scatter semantically similar intents in one stage; While the *expansion* tends to scatter semantically dissimilar intents in one stage.
- Figure 9: t-SNE projections of the three strategies on BankCL. Visualizes how *cohesion*, *expansion*, and *random* scatter intents across stages.

D.3 Impact of Sample Distribution

This subsection presents experiment results on the Zipf exponent s , which controls the long-tail skewness of per-new-intent sample counts in each stage (Section 5.3).

- Table 14: Main results on BankCL. Comparison among the three methods at each s , paralleling the ClincCL main results.
- Table 15: Per-Stage results on ClincCL. Reports CIDA performance at the base stage t_0 and at each incremental stage t_1 – t_5 on ClincCL under each s .
- Table 16: Per-Stage results on BankCL. Reports CIDA performance at the base stage t_0 and at every incremental stage t_1 – t_5 on BankCL under each s .
- Figure 10: Radar plot on ClincCL. It plots the main results of all performance metrics of CIDA under difference choices of s on ClincCL. The larger the value of the Zipf exponent s , the more unbalanced distribution of sample counts for new-intents in one stage. It can be observed from this figure that the smaller value of s , the better the performance of the CIDA, indicating that the per-new-intent sample distribution can control the

Setting	Stage	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
init (shared)	t_0	—	—	—	—	—	94.28	—	—	94.28	—	—
$\alpha=0$	t_1	89.92	1.25	15.54	91.12	81.33	88.94	75.54	82.45	93.88	83.72	93.88
	t_2	87.36	0.48	20.68	89.86	76.28	84.44	63.64	77.07	88.85	74.16	91.36
	t_3	94.70	0.71	4.29	89.46	74.31	82.61	69.29	70.17	84.09	75.97	88.94
	t_4	74.42	0.57	36.00	88.98	71.93	80.80	57.00	83.34	82.50	67.42	87.33
	t_5	94.04	0.00	11.25	89.09	72.08	81.04	88.75	93.55	80.83	84.61	86.03
	<i>avg</i>	88.09	0.60	17.55	89.70	75.18	83.57	70.84	81.32	86.03	77.17	89.51
$\alpha=0.5$	t_1	89.76	2.11	13.93	90.38	79.79	87.40	71.61	78.59	93.22	81.00	93.22
	t_2	84.26	2.36	19.09	89.23	75.97	84.33	74.77	86.66	86.35	80.14	89.78
	t_3	80.40	0.63	28.93	88.52	73.28	81.46	66.43	79.76	83.13	73.85	87.57
	t_4	67.28	0.61	45.00	87.56	70.31	78.83	43.50	73.68	81.36	56.69	86.02
	t_5	79.25	0.53	21.25	87.42	69.98	78.80	78.75	87.50	78.80	78.77	84.57
	<i>avg</i>	80.19	1.25	25.64	88.62	73.86	82.17	67.01	81.24	84.57	74.09	88.23
$\alpha=0.8$	t_1	82.08	7.11	16.96	88.43	74.78	85.05	72.32	80.06	89.74	80.09	89.74
	t_2	78.47	4.09	22.95	87.30	70.81	81.11	64.09	76.73	84.71	72.97	87.22
	t_3	86.34	2.26	8.57	86.60	68.76	78.96	56.07	64.71	81.51	66.44	85.32
	t_4	57.45	2.25	47.00	86.03	66.78	77.67	41.50	71.31	80.25	54.71	84.05
	t_5	80.00	0.43	22.50	85.86	66.71	77.79	77.50	86.50	77.80	77.65	82.80
	<i>avg</i>	76.87	3.23	23.60	86.85	69.57	80.12	62.30	75.86	82.80	70.37	85.83

Table 10: Per-stage results on BankCL across α (%). $\alpha \in [0, 1)$ is the old-intent recurrence ratio. ↓: lower-is-better. Click [Section D.1](#) to jump back.

difficulty level of the COSID task (c.f., Discussion in Section 5.3).

- Figure 11: Radar plot on BankCL. It plots the main results of all performance metrics of CIDA under different choices of s on BankCL.

D.4 Impact of Number of New-Intent Samples

This subsection presents experiment results on the per-new-intent sample budget K_{init} in each stage, which determines how many labeled samples each base intent provides at t_0 and, equivalently, the average sample budget of each newly arriving intent in t_1 – t_5 (Section 5.4).

- Table 17: Main results on BankCL. Comparison among the three methods at each K_{init} , paralleling the ClinCL main results.
- Table 18: Per-Stage results on ClinCL. Reports CIDA performance at the base stage t_0 and at each incremental stage t_1 – t_5 on ClinCL under each K_{init} .
- Table 19: Per-Stage results on BankCL. Reports CIDA performance at the base stage t_0 and at each incremental stage t_1 – t_5 on BankCL under each K_{init} .

- Figure 12: Radar plot on ClinCL. It plots the main results of all performance metrics of CIDA under different choices of K_{init} on ClinCL. The larger value of K_{init} , the more samples of a new-intent (with fixed new-intent sample distribution skewness and old-intent sample mixing ratio). Generally, more samples of a particular intent can help clustering boundary-clear clusters, thereby improving new-intent discovery. And as with more samples per new-intent have been appended into the long-term memory, the prototype of an intent is more differentiable to other intents, thereby increasing the system discrimination and retention abilities. This also indicates that the parameter K_{init} can control the difficulty level of the COSID task.

- Figure 13: Radar plot on BankCL. It plots the main results of all performance metrics of CIDA under different choices of K_{init} on BankCL.

E Illustration of Stream Construction

Figure 14 provides an example on partial results of the ClinCL constructed from the original dataset CLINC150. Pannel (a) presents a few samples each in the form of (*intent-name*, *query*), and panel (b)

Strategy	Method	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
EXPANSION	CGID	—	—	—	55.57	12.85	36.12	33.39	57.17	35.69	30.92	36.68
	VBCGCD	62.69	0.78	49.75	82.66	59.92	72.92	46.18	71.98	78.03	57.07	85.11
	Ours	73.71	1.23	34.65	89.63	74.86	82.71	57.41	77.83	85.80	68.09	90.11
RANDOM	CGID	—	—	—	52.97	11.66	34.82	49.09	61.86	34.19	37.98	34.97
	VBCGCD	77.37	1.04	30.07	81.15	59.22	71.47	49.89	56.85	75.66	59.09	81.58
	Ours	80.19	1.25	25.64	88.62	73.86	82.17	67.01	81.24	84.57	74.09	88.23
COHESION	CGID	—	—	—	46.23	7.50	29.78	38.14	45.62	28.89	32.43	28.67
	VBCGCD	85.70	0.98	15.95	81.58	59.16	71.11	51.87	55.38	74.81	60.84	79.92
	Ours	84.11	1.42	18.45	86.72	69.06	79.52	68.85	81.23	81.75	73.54	84.88

Table 11: Results on BankCL across intent distribution strategies (%) at $\alpha=0.5$. Each row is the mean over incremental stages t_1 – t_5 ; full per-stage breakdown is reported in Table 13. Strategies follow §2.2. ↓: lower-is-better. Click [Section D.2](#) to jump back.

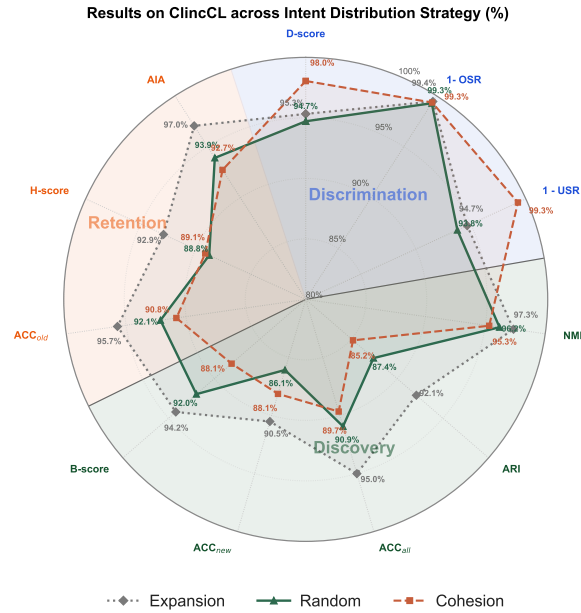


Figure 6: Radar comparison across all metrics on ClinCL under different intent distribution strategies. Click [Section D.2](#) to jump back.

shows how these samples are divided into different stages. Furthermore, an intent is labeled as **NEW** if it is the initial occurrence in a stage; Otherwise, it is labeled as **OLD** for its subsequent recurrences in later stages.

F Prompt design

This appendix lists the four prompt templates referenced in the Method section (Section 4). All prompts are issued to the same backbone LLM, Qwen3.5-9B in our experiments.

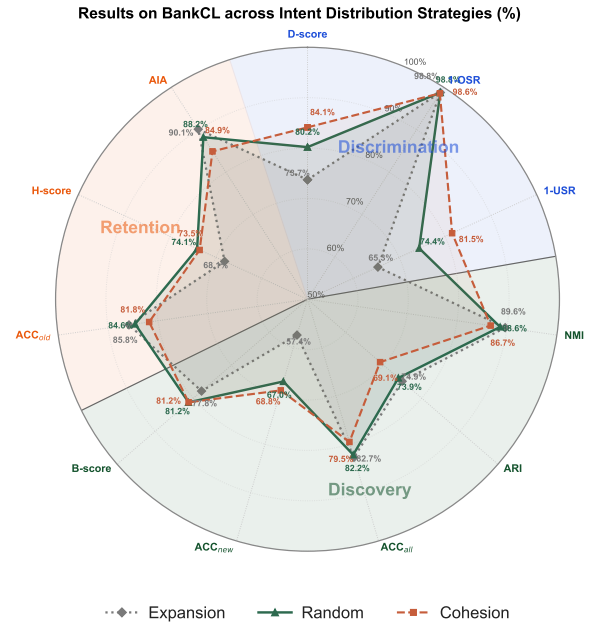


Figure 7: Radar comparison across all metrics on BankCL under different intent distribution strategies. Click [Section D.2](#) to jump back.

F.1 The Prompt in Long-term Memory

Figure 15 sketches the prompt used in the initialization stage to derive the intent description for each base intent from its labeled samples (Section 4.1). The same template is also used in the *discovery module* to write a description for every newly discovered intent.

F.2 The Prompt in Discrimination Module

Figure 16 sketches the prompt for the agent to decide whether a sample x matches one of the retrieved candidate intents $\mathcal{Q}(x)$ or should be included to the temporary buffer $\mathcal{B}$ as a new-intent

Strategy	Stage	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
EXPANSION	t_0	—	—	—	—	—	98.98	—	—	98.98	—	—
	t_1	97.80	0.76	2.13	98.42	96.20	98.10	97.60	98.66	98.27	97.93	98.27
	t_2	97.19	0.70	2.17	97.67	93.06	95.89	87.67	89.31	97.53	92.34	97.90
	t_3	90.95	0.89	10.67	97.07	90.83	93.80	79.33	89.10	95.61	86.71	97.14
	t_4	94.84	0.40	5.00	96.80	90.22	93.63	94.67	97.08	93.56	94.11	96.24
	t_5	95.89	0.05	6.67	96.80	90.18	93.64	93.33	96.90	93.66	93.49	95.72
	<i>avg</i>	95.33	0.56	5.33	97.35	92.10	95.01	90.52	94.21	95.72	92.92	97.05
RANDOM	t_0	—	—	—	—	—	97.91	—	—	97.91	—	—
	t_1	95.36	1.78	4.00	96.45	88.62	92.50	79.47	84.82	96.84	87.30	96.84
	t_2	94.90	0.97	5.33	96.31	87.83	90.89	82.17	88.21	92.63	87.09	94.74
	t_3	95.39	0.42	5.78	96.01	86.83	90.27	89.11	94.87	90.42	89.76	93.30
	t_4	96.91	0.42	0.67	96.08	87.13	90.64	99.00	99.33	90.02	94.30	92.48
	t_5	91.04	0.05	15.33	96.08	86.64	90.40	80.67	92.56	90.74	85.41	92.13
	<i>avg</i>	94.72	0.73	6.22	96.18	87.41	90.94	86.08	91.96	92.13	88.77	93.90
COHESION	t_0	—	—	—	—	—	97.82	—	—	97.82	—	—
	t_1	96.82	2.00	0.53	95.31	86.51	91.03	75.60	75.55	96.18	84.66	96.18
	t_2	98.51	0.50	0.50	95.20	85.28	89.78	85.33	84.42	90.67	87.92	93.42
	t_3	97.40	0.64	0.22	95.20	84.32	88.81	85.78	85.38	89.19	87.45	92.01
	t_4	98.50	0.10	1.67	95.37	84.66	89.17	94.67	95.45	88.77	91.62	91.20
	t_5	99.00	0.05	0.67	95.53	85.09	89.51	99.33	99.72	89.17	93.98	90.80
	<i>avg</i>	98.05	0.66	0.72	95.32	85.17	89.66	88.14	88.11	90.80	89.13	92.72

Table 12: Per-stage results on ClinCCL across intent distribution strategies (%) at $\alpha=0.5$. Strategies follow §2.2. ↓: lower-is-better. Click [Section D.2](#) to jump back.

(Section 4.2). When the LLM output cannot exactly match one of the intent labels in $\mathcal{Q}(x)$, it is mapped to its semantically closest intent label via cosine similarity. were AI-generated.

F.3 The Prompt in Discovery Module

Figure 17 sketches the prompt for the agent to name an intent for each cluster formed from the temporary buffer. It is required that the name of a new-intent should be distinct from those of historical intents in the long-term memory.

F.4 The Prompt in Inference Module

Figure 18 sketches the intent prediction prompt for the inference module. It is used to classify a test query to its best matching intent in the updated memory $\mathcal{M}$ using the top- k retrieved exemplars and their associated intent names (Section 4.4). If the output fails to match any name of the historical intents, the prompt instructs an LLM to classify the query to the intent of the top-1 retrieved exemplar.

G Use of AI Assistants

We used AI assistants only for coding support. Each line of codes was carefully reviewed by the authors. No experimental results or conclusions

Strategy	Stage	Discrimination			Discovery				Retention			
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
EXPANSION	t_0	—	—	—	—	—	96.71	—	—	96.71	—	—
	t_1	89.83	2.37	13.21	92.63	84.09	89.95	76.07	84.54	95.07	84.51	95.07
	t_2	83.50	1.25	24.09	91.08	78.51	85.79	68.18	85.55	89.52	77.41	92.29
	t_3	66.23	0.91	46.43	88.92	72.76	80.75	41.79	75.60	85.08	56.05	89.89
	t_4	73.82	1.46	29.50	88.05	70.31	79.07	61.00	81.98	80.36	69.35	87.51
	t_5	55.17	0.13	60.00	87.45	68.61	77.99	40.00	61.49	79.00	53.11	85.80
	avg	73.71	1.23	34.65	89.63	74.86	82.71	57.41	77.83	85.80	68.09	90.11
RANDOM	t_0	—	—	—	—	—	94.28	—	—	94.28	—	—
	t_1	89.76	2.11	13.93	90.38	79.79	87.40	71.61	78.59	93.22	81.00	93.22
	t_2	84.26	2.36	19.09	89.23	75.97	84.33	74.77	86.66	86.35	80.14	89.78
	t_3	80.40	0.63	28.93	88.52	73.28	81.46	66.43	79.76	83.13	73.85	87.57
	t_4	67.28	0.61	45.00	87.56	70.31	78.83	43.50	73.68	81.36	56.69	86.02
	t_5	79.25	0.53	21.25	87.42	69.98	78.80	78.75	87.50	78.80	78.77	84.57
	avg	80.19	1.25	25.64	88.62	73.86	82.17	67.01	81.24	84.57	74.09	88.23
COHESION	t_0	—	—	—	—	—	90.99	—	—	90.99	—	—
	t_1	91.54	1.12	13.04	87.56	72.76	83.12	64.29	68.35	90.07	75.02	90.07
	t_2	90.23	0.72	15.00	85.65	67.13	78.29	57.05	66.53	82.79	67.55	86.43
	t_3	76.57	4.88	10.71	86.31	67.72	78.29	76.43	82.54	78.49	77.45	83.78
	t_4	95.54	0.39	3.50	87.07	69.22	79.37	96.50	98.39	78.14	86.36	82.37
	t_5	66.67	0.00	50.00	87.00	68.47	78.51	50.00	90.32	79.27	61.32	81.75
	avg	84.11	1.42	18.45	86.72	69.06	79.52	68.85	81.23	81.75	73.54	84.88

Table 13: Per-stage results on BankCL across intent distribution strategies (%). ↓: lower-is-better. Click [Section D.2](#) to jump back.

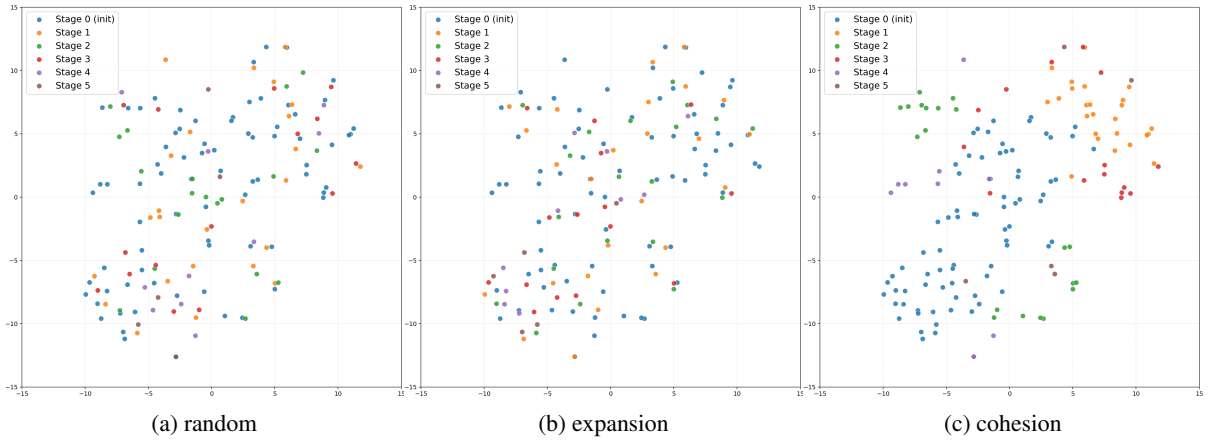


Figure 8: t-SNE projections of three intent distribution strategies on ClinCL. Different colors represent different stages. Click [Section D.2](#) to jump back.

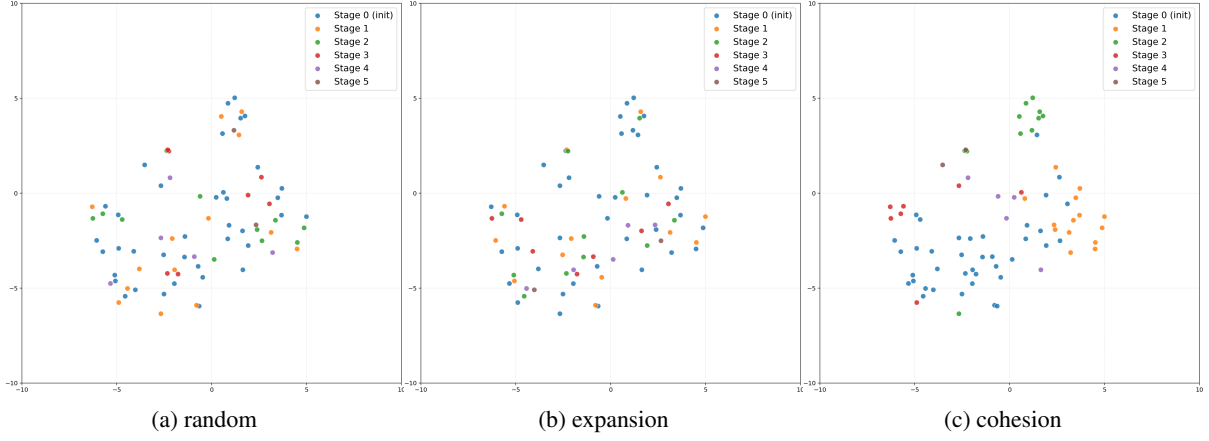


Figure 9: t-SNE projections of three intent distribution strategies on BankCL. Different colors represent different stages. Click [Section D.2](#) to jump back.

Setting	Method	Discrimination			Discovery				Retention			
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
$s=0.5$	CGID	—	—	—	52.85	11.43	33.29	46.03	59.17	33.29	35.61	34.54
	VBCGCD	76.65	0.97	30.80	80.61	57.67	70.52	54.99	68.18	74.75	61.26	81.02
	Ours	84.10	1.18	21.09	87.87	71.11	79.80	65.15	77.62	82.82	71.98	87.18
$s=0.8$	CGID	—	—	—	51.93	11.05	33.18	46.91	57.53	33.07	35.90	34.89
	VBCGCD	73.20	1.34	33.91	79.88	56.99	69.99	47.95	59.08	74.41	57.21	80.51
	Ours	82.69	1.20	23.70	87.45	70.19	78.44	63.57	76.09	81.65	70.75	86.54
$s=1.0$	CGID	—	—	—	50.28	9.53	31.76	46.39	57.36	31.60	34.28	33.44
	VBCGCD	71.60	1.43	35.28	79.49	55.59	67.93	50.30	66.78	72.50	57.12	79.52
	Ours	83.10	1.33	22.08	85.80	65.88	74.96	60.67	72.84	78.78	66.87	84.41

Table 14: Results on BankCL across the long-tail exponent s (%). Each row is the mean over incremental stages t_1 – t_5 ; full per-stage breakdown is in Table 16. ↓: lower-is-better. Click [Section D.3](#) to jump back.

Setting	Stage	Discrimination			Discovery				Retention			
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
init (shared)	t_0	—	—	—	—	—	97.91	—	—	97.91	—	—
$s=0.5$	t_1	95.52	1.38	4.80	96.47	89.58	93.20	81.33	85.95	97.16	88.54	97.16
	t_2	94.68	1.10	5.17	96.13	88.09	91.64	84.50	89.56	93.07	88.58	95.11
	t_3	95.42	0.50	5.11	95.83	86.71	90.59	84.67	89.05	91.33	87.87	93.85
	t_4	96.10	0.49	1.33	95.74	86.40	90.14	85.33	85.93	90.49	87.84	93.01
	t_5	95.89	0.05	6.67	95.65	86.03	90.07	89.33	91.09	90.09	89.71	92.43
	avg	95.52	0.70	4.62	95.96	87.36	91.13	85.03	88.32	92.43	88.51	94.31
$s=0.8$	t_1	93.96	2.18	5.60	95.28	83.48	90.90	73.73	78.09	96.62	83.64	96.62
	t_2	94.10	1.23	5.67	94.67	81.42	88.36	77.00	81.57	90.63	83.26	93.63
	t_3	93.57	0.83	6.22	94.15	81.09	86.94	78.67	84.12	87.97	83.06	91.74
	t_4	92.74	0.99	2.00	94.13	80.51	85.33	61.00	65.98	87.14	71.76	90.59
	t_5	95.21	0.07	7.33	94.03	80.05	85.24	88.67	90.53	85.13	86.86	89.50
	avg	93.92	1.06	5.36	94.45	81.31	87.36	75.81	80.06	89.50	81.72	92.42
$s=1.0$	t_1	94.69	1.47	6.13	95.43	85.72	90.43	71.47	77.28	96.76	82.21	96.76
	t_2	89.71	2.87	7.00	94.74	82.26	86.72	72.33	79.09	89.60	80.05	93.18
	t_3	91.13	0.97	9.78	93.97	80.49	85.33	76.44	84.24	86.44	81.14	90.93
	t_4	95.88	0.40	3.00	93.85	79.65	84.23	70.00	70.88	85.28	76.89	89.52
	t_5	80.39	0.83	16.67	93.73	78.65	83.64	71.33	86.23	84.07	77.18	88.43
	avg	90.36	1.31	8.52	94.34	81.35	86.07	72.32	79.55	88.43	79.49	91.76

Table 15: Per-stage results on ClinCL across the long-tail exponent s (%). ↓: lower-is-better. Click [Section D.3](#) to jump back.

Setting	Stage	Discrimination			Discovery				Retention			
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
init (shared)	t_0	—	—	—	—	—	94.28	—	—	94.28	—	—
$s=0.5$	t_1	86.39	2.70	18.39	89.96	78.86	86.68	70.18	83.84	92.76	79.91	92.76
	t_2	83.06	1.25	24.77	88.06	72.01	80.60	54.77	71.62	86.06	66.94	89.41
	t_3	90.07	1.19	9.29	87.63	69.90	78.46	59.29	63.72	80.60	68.32	86.47
	t_4	68.67	0.64	43.00	86.84	67.35	76.57	51.50	74.52	78.36	62.15	84.44
	t_5	92.31	0.13	10.00	86.86	67.43	76.69	90.00	94.39	76.33	82.61	82.82
	<i>avg</i>	84.10	1.18	21.09	87.87	71.11	79.80	65.15	77.62	82.82	71.98	87.18
$s=0.8$	t_1	86.54	1.97	19.64	90.16	78.79	85.82	65.54	79.66	93.29	76.99	93.29
	t_2	82.83	1.97	22.73	87.84	71.19	79.13	52.50	65.87	84.76	64.84	89.02
	t_3	86.65	1.19	15.36	86.87	68.47	76.68	61.07	65.90	78.41	68.66	85.49
	t_4	66.27	0.86	44.50	86.11	66.15	75.13	55.00	78.16	76.57	64.02	83.26
	t_5	91.16	0.00	16.25	86.28	66.37	75.45	83.75	90.88	75.23	79.26	81.65
	<i>avg</i>	82.69	1.20	23.70	87.45	70.19	78.44	63.57	76.09	81.65	70.75	86.54
$s=1.0$	t_1	84.25	2.57	22.14	88.07	73.11	81.59	51.61	65.33	92.63	66.29	92.63
	t_2	81.46	1.44	26.59	86.10	67.08	76.15	51.14	68.25	81.44	62.83	87.04
	t_3	85.01	1.83	13.93	85.30	64.24	73.36	57.86	63.80	75.08	65.35	83.05
	t_4	69.23	0.75	41.50	84.60	62.17	71.53	49.00	70.11	73.14	58.69	80.57
	t_5	95.54	0.07	6.25	84.95	62.82	72.18	93.75	96.69	71.60	81.19	78.78
	<i>avg</i>	83.10	1.33	22.08	85.80	65.88	74.96	60.67	72.84	78.78	66.87	84.41

Table 16: Per-stage results on BankCL across the long-tail exponent s (%). ↓: lower-is-better. Click [Section D.3](#) to jump back.

Setting	Method	Discrimination			Discovery				Retention			
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
$K_{\text{init}}=5$	CGID	—	—	—	56.76	22.79	40.42	35.22	45.04	41.81	37.50	40.89
	VBCGCD	41.02	1.95	69.75	56.81	26.41	43.65	21.77	41.97	47.98	28.89	53.77
	Ours	74.68	2.18	32.17	84.14	62.57	72.95	57.34	73.97	76.27	64.95	81.11
$K_{\text{init}}=10$	CGID	—	—	—	57.17	19.14	41.22	35.27	47.92	42.72	38.37	47.00
	VBCGCD	69.63	1.98	37.94	72.71	46.13	62.07	49.73	63.55	65.36	54.48	70.84
	Ours	77.35	2.07	26.44	85.17	65.26	76.01	62.63	78.82	79.01	68.97	83.05
$K_{\text{init}}=30$	CGID	—	—	—	52.97	11.66	34.82	49.09	61.86	34.19	37.98	34.97
	VBCGCD	77.37	1.04	30.07	81.15	59.22	71.47	49.89	56.85	75.66	59.09	81.58
	Ours	80.19	1.25	25.64	88.62	73.86	82.17	67.01	81.24	84.57	74.09	88.23
$K_{\text{init}}=50$	CGID	—	—	—	52.91	12.10	33.87	54.89	65.38	31.55	37.92	31.03
	VBCGCD	77.17	1.13	29.82	81.00	58.44	70.43	53.22	66.38	74.77	60.19	81.11
	Ours	85.13	1.07	20.66	88.38	72.83	81.09	65.74	76.60	83.85	72.84	87.82

Table 17: Results on BankCL across the base sample size K_{init} (%). Each row is the mean over incremental stages t_1 – t_5 . ↓: lower-is-better. Click [Section D.4](#) to jump back.

Setting	Stage	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
$K_{\text{init}}=5$	t_0	—	—	—	—	—	96.89	—	—	96.89	—	—
	t_1	88.58	3.51	12.13	93.39	79.96	87.37	67.33	75.88	94.04	78.48	94.04
	t_2	83.15	6.47	5.83	91.81	70.82	83.33	73.83	76.56	85.23	79.12	89.64
	t_3	71.06	6.25	17.33	91.28	64.66	81.43	70.44	90.24	82.81	76.13	87.36
	t_4	91.76	1.19	1.67	91.72	67.95	82.21	86.33	87.86	81.90	84.06	86.00
	t_5	49.79	4.87	20.00	91.28	65.54	81.24	78.00	93.70	81.36	79.64	85.07
	<i>avg</i>	76.87	4.46	11.39	91.90	69.79	83.12	75.19	84.85	85.07	79.49	88.42
$K_{\text{init}}=10$	t_0	—	—	—	—	—	97.51	—	—	97.51	—	—
	t_1	89.88	3.91	8.80	94.67	84.26	89.80	73.07	82.57	95.38	82.74	95.38
	t_2	89.49	2.43	9.17	93.27	78.58	87.11	78.33	83.71	88.87	83.27	92.12
	t_3	79.63	3.75	14.00	92.83	76.10	85.26	72.00	85.77	86.92	78.76	90.39
	t_4	75.41	3.73	9.00	92.65	74.47	84.64	78.00	87.92	85.14	81.41	89.07
	t_5	89.70	0.74	1.33	92.96	75.74	85.04	98.67	99.44	84.57	91.08	88.17
	<i>avg</i>	84.82	2.91	8.46	93.27	77.83	86.37	80.01	87.88	88.18	83.45	91.03
$K_{\text{init}}=30$	t_0	—	—	—	—	—	97.91	—	—	97.91	—	—
	t_1	95.36	1.78	4.00	96.45	88.62	92.50	79.47	84.82	96.84	87.30	96.84
	t_2	94.90	0.97	5.33	96.31	87.83	90.89	82.17	88.21	92.63	87.09	94.74
	t_3	95.39	0.42	5.78	96.01	86.83	90.27	89.11	94.87	90.42	89.76	93.30
	t_4	96.91	0.42	0.67	96.08	87.13	90.64	99.00	99.33	90.02	94.30	92.48
	t_5	91.04	0.05	15.33	96.08	86.64	90.40	80.67	92.56	90.74	85.41	92.13
	<i>avg</i>	94.72	0.73	6.22	96.18	87.41	90.94	86.08	91.96	92.13	88.77	93.90
$K_{\text{init}}=50$	t_0	—	—	—	—	—	98.22	—	—	98.22	—	—
	t_1	96.02	1.47	3.60	96.92	90.84	94.17	84.67	89.47	97.33	90.56	97.33
	t_2	95.86	1.40	1.50	96.37	89.29	92.78	88.83	88.96	93.57	91.14	95.45
	t_3	93.86	0.47	8.22	96.12	88.41	92.30	89.78	95.07	92.61	91.17	94.50
	t_4	97.86	0.25	1.00	96.25	88.90	92.69	98.67	99.15	92.25	95.35	93.94
	t_5	95.47	0.00	8.67	96.20	88.49	92.56	90.67	95.30	92.62	91.63	93.68
	<i>avg</i>	95.81	0.72	4.60	96.37	89.19	92.90	90.52	93.59	93.68	91.97	94.98

Table 18: Per-stage results on ClincCL across the base sample size K_{init} (%). ↓: lower-is-better. Click [Section D.4](#) to jump back.

Setting	Stage	Discrimination			Discovery					Retention		
		D -score	OSR↓	USR↓	NMI	ARI	ACC _{all}	ACC _{new}	B -score	ACC _{old}	H -score	AIA
$K_{\text{init}}=5$	t_0	—	—	—	—	—	88.68	—	—	88.68	—	—
	t_1	74.77	3.49	34.64	85.87	69.53	79.33	56.79	76.65	87.63	68.91	87.63
	t_2	77.76	4.09	24.09	85.22	64.65	74.13	47.50	63.14	79.76	59.54	83.70
	t_3	76.37	1.87	27.86	83.82	61.09	71.18	54.64	69.96	73.02	62.51	80.14
	t_4	64.77	1.36	43.00	83.01	59.09	70.10	59.00	80.61	70.89	64.40	77.82
	t_5	79.71	0.10	31.25	82.79	58.50	70.03	68.75	79.52	70.07	69.40	76.27
	avg	74.68	2.18	32.17	84.14	62.57	72.95	57.34	73.97	76.27	64.95	81.11
$K_{\text{init}}=10$	t_0	—	—	—	—	—	91.45	—	—	91.45	—	—
	t_1	74.92	4.01	33.57	86.76	70.73	79.52	52.50	74.15	89.47	66.17	89.47
	t_2	82.25	2.45	22.05	85.64	67.21	77.54	67.27	81.85	79.71	72.97	84.59
	t_3	83.79	1.55	17.86	84.67	64.14	75.18	64.64	74.31	76.35	70.01	81.84
	t_4	64.52	1.86	40.00	84.31	62.15	73.83	47.50	76.14	75.71	58.38	80.31
	t_5	81.25	0.50	18.75	84.44	62.06	73.99	81.25	87.64	73.80	77.35	79.01
	avg	77.35	2.07	26.44	85.17	65.26	76.01	62.63	78.82	79.01	68.97	83.05
$K_{\text{init}}=30$	t_0	—	—	—	—	—	94.28	—	—	94.28	—	—
	t_1	89.76	2.11	13.93	90.38	79.79	87.40	71.61	78.59	93.22	81.00	93.22
	t_2	84.26	2.36	19.09	89.23	75.97	84.33	74.77	86.66	86.35	80.14	89.78
	t_3	80.40	0.63	28.93	88.52	73.28	81.46	66.43	79.76	83.13	73.85	87.57
	t_4	67.28	0.61	45.00	87.56	70.31	78.83	43.50	73.68	81.36	56.69	86.02
	t_5	79.25	0.53	21.25	87.42	69.98	78.80	78.75	87.50	78.80	78.77	84.57
	avg	80.19	1.25	25.64	88.62	73.86	82.17	67.01	81.24	84.57	74.09	88.23
$K_{\text{init}}=50$	t_0	—	—	—	—	—	94.28	—	—	94.28	—	—
	t_1	87.64	2.63	16.43	90.45	79.99	87.60	73.57	82.25	92.76	82.06	92.76
	t_2	84.46	1.54	21.59	88.66	74.12	82.38	61.82	77.26	86.73	72.19	89.75
	t_3	93.55	0.67	6.79	88.27	72.02	80.11	59.29	62.56	82.42	68.96	87.30
	t_4	67.29	0.46	46.00	87.14	68.90	77.53	46.50	68.61	79.75	58.75	85.42
	t_5	92.72	0.03	12.50	87.37	69.15	77.86	87.50	92.31	77.60	82.25	83.85
	avg	85.13	1.07	20.66	88.38	72.83	81.09	65.74	76.60	83.85	72.84	87.82

Table 19: Per-stage results on BankCL across the base sample size K_{init} (%). ↓: lower-is-better. Click [Section D.4](#) to jump back.

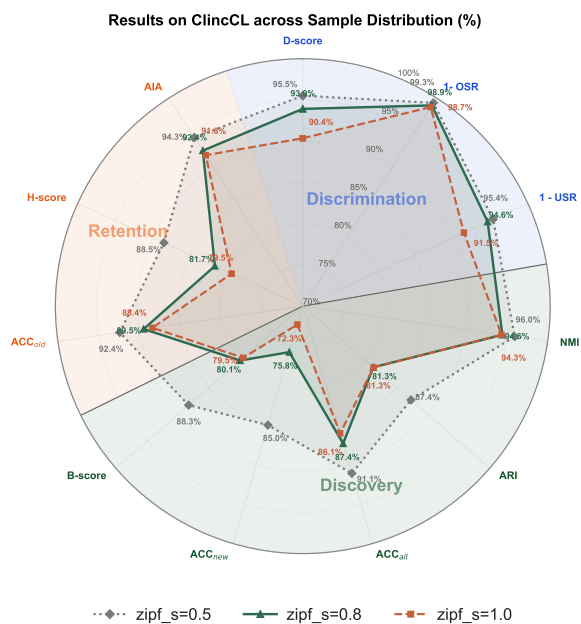


Figure 10: Radar comparison across all metrics on ClincCL under varying s . Click [Section D.3](#) to jump back.

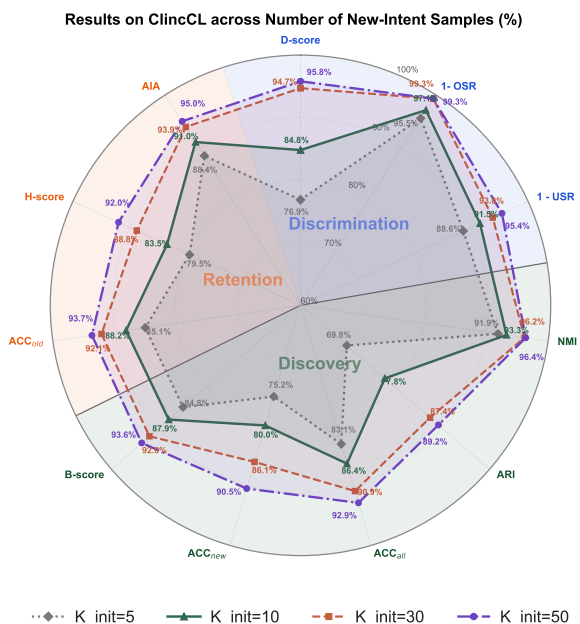


Figure 12: Radar comparison across all metrics on ClincCL under varying K_{init} . Click [Section D.4](#) to jump back.

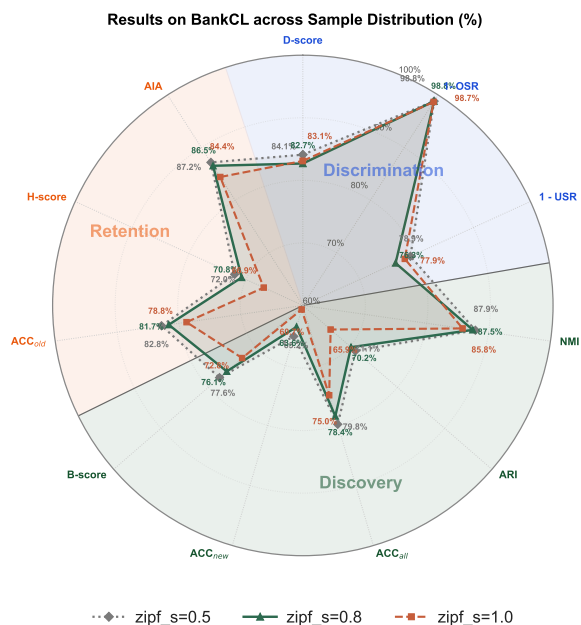


Figure 11: Radar comparison across all metrics on BankCL under varying s . Click [Section D.3](#) to jump back.

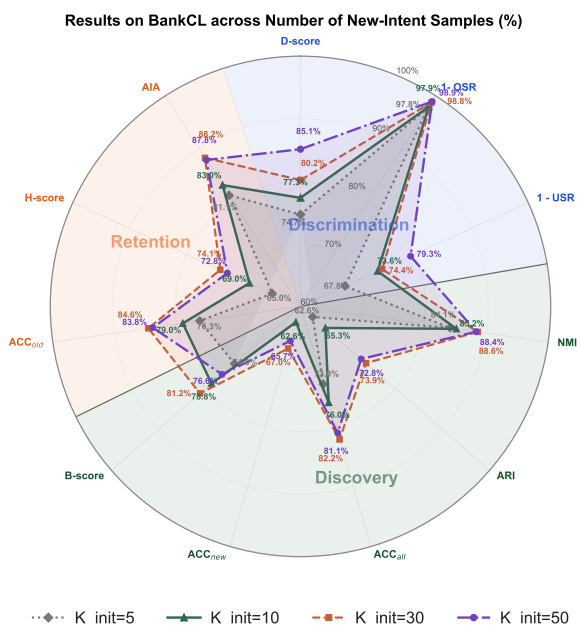


Figure 13: Radar comparison across all metrics on BankCL under varying K_{init} . Click [Section D.4](#) to jump back.

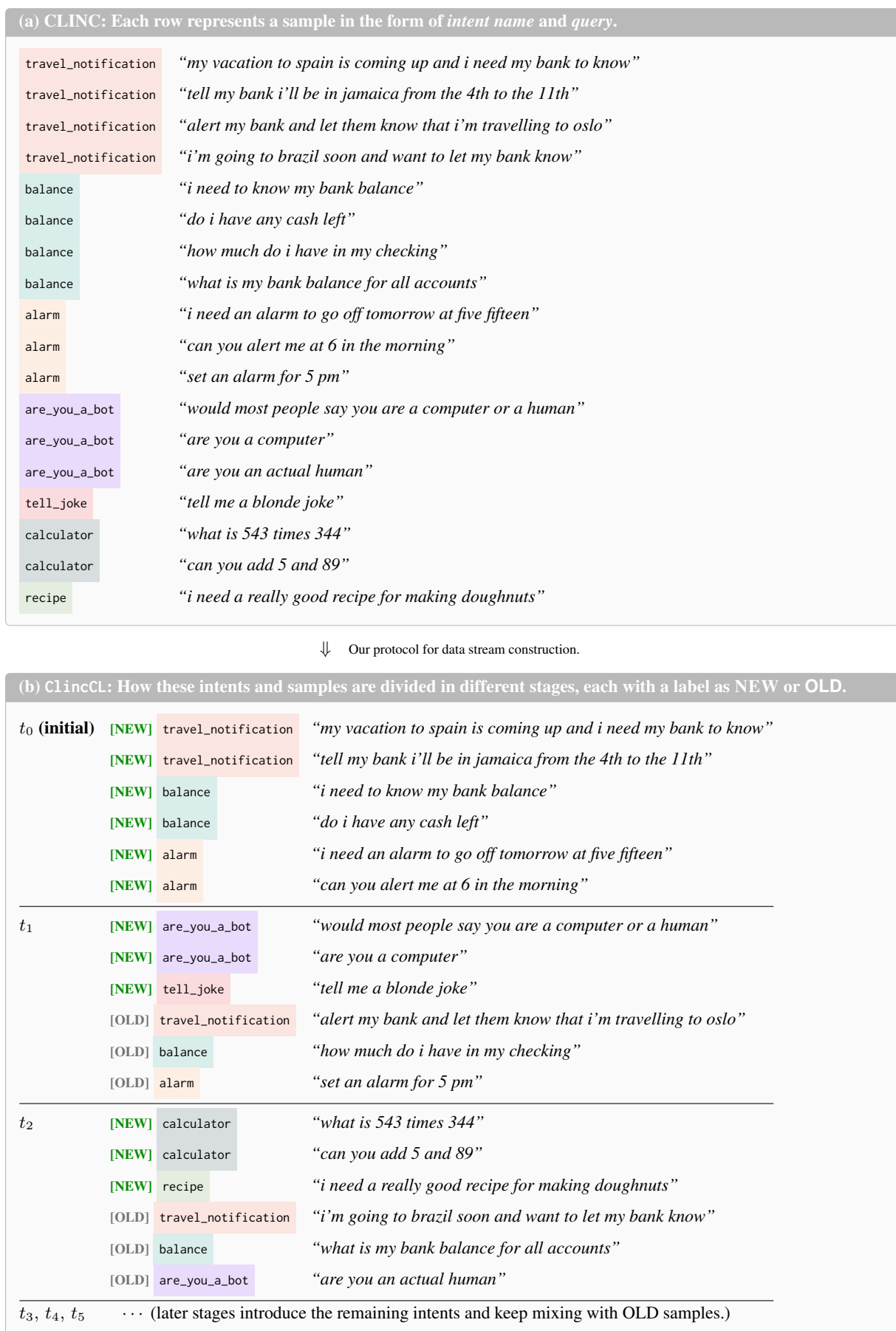


Figure 14: Sample-level view of how the ClincCL stream (b) is constructed from the CLINC dataset (a).

Prompt Template
Prompt for Intent Description Generation Role. You are an intent analysis expert. Given an intent name and a list of user queries that belong to this intent, write a concise description of what this intent covers. Input. • Intent name: {intent_name} • User queries belonging to this intent: {samples} Instructions. • Write a concise description in 1–2 sentences (English, no more than 40 words). • Focus on the user’s goal/action, not on specific entities. • Return only the description text, without any prefix or quotes.

Figure 15: Prompt for generating intent descriptions d_c . Used in the Long-term Memory and reused in the Discovery Module; click [Section 4.1](#) or [Section 4.3](#) to jump back.

Prompt Template
Prompt for Per-sample Intent Screening Role. You are assisting with new intent discovery in a continual learning system. Given a user query and a list of known intents (each with a description), decide whether the query belongs to one of the known intents, or whether it represents a new intent that should be added to the system. Input. • User query: {sample} • Candidate known intents (one of these may match, or none of them): {candidates} Instructions. • If the query clearly matches one of the known intents above, respond with exactly that intent’s name. • If the query does <i>not</i> match any of the above known intents, respond with exactly: NEW. • Be strict: only pick a known intent when the query is a genuine instance of it; otherwise prefer NEW so the system can discover the new intent. Output. Your answer (either one known intent name, or NEW):

Figure 16: Prompt for per-sample intent screening in the Discrimination Module; click [Section 4.2](#) to jump back. Slot {candidates} is a bullet list of name: description pairs for the top- k retrieved candidate intents.

Prompt Template
Prompt for New-intent Conceptualization Role. You are an intent analysis expert. Below is a group of highly similar customer query samples. Analyze their common intent. Input. • Sample list: {examples} • Existing intent names (must not be reused, including close synonyms): {existing_intents} Instructions. • Provide a <i>new and unique</i> intent name (English, concise, no more than 5 words) that is different from every existing intent name above. • Return only the intent name, without any additional explanation.

Figure 17: Prompt for naming each cluster in the Discovery Module; click [Section 4.3](#) to jump back. The slot {existing_intents} is populated with every intent already in the long-term memory plus every intent named earlier in the same stage, so that the LLM never produces collisions.

Prompt Template
Prompt for Intent Prediction at Inference Role. Based on the following similar samples, determine which intent the query sample most likely belongs to. Input. • Query sample: {sample} • Similar samples context (top-k retrieved exemplars with their intent names): {context} Instructions. • Return only the intent name, without any additional explanation.

Figure 18: Prompt used by the Inference Module to assign a test query to one of the intents stored in the memory $\mathcal{M}$; click [Section 4.4](#) to jump back.