Continual Learning Using Only Large Language Model Prompting *

Jiabao Qiu[1], **Zixuan Ke**[2], **Bing Liu**[1]
[1]Department of Computer Science, University of Illinois Chicago
[2]Salesforce AI Research
[1]{jqiu23, liub}@uic.edu
[2]zixuan.ke@salesforce.com

Abstract

We introduce *CLOB*, a novel *continual learning* (CL) *paradigm* wherein a large language model (LLM) is regarded as a black box. Learning is done incrementally via only verbal prompting. CLOB does not fine-tune any part of the LLM or add any trainable parameters to it. It is particularly suitable for LLMs that are accessible via APIs. We also propose a new CL technique, called CIS, based on *incremental summarization* that also overcomes the LLM's input length limit. Experiments show CIS outperforms baselines by a very large margin.

1 Introduction

Continual learning (CL) learns a sequence of tasks incrementally (Chen and Liu, 2018). Once a task is learned, its training data is discarded. Existing CL work in NLP mainly fine-tunes language model (LM) parameters or uses adapters or variants to adapt the LM (Ke and Liu, 2022). In learning a new task, the system must update the network parameters without seeing the old task data. This may corrupt the knowledge learned from old tasks, causing *catastrophic forgetting* (CF). Further, since a large number of training samples, expensive training, and access to the LM parameters are needed, it is limited to "small" LMs.

Recently, large language models (LLMs) such as GPT4 (OpenAI, 2023) and Gemini (Google, 2024) have revolutionized NLP, but they are typically accessed only through APIs. In these "black-boxes"[1] LLMs, the users typically use prompts that include few-shot in-context examples and instructions to ask the LLMs to perform their tasks.

CL with black-box LLMs is clearly important. However, to our knowledge, CL has not been explored in this context. Our **first contribution** is thus to propose a new CL paradigm for this context, called **CLOB** (**C**ontinual **L**earning **O**ver **B**lack-box LLMs). The user works with the LLM using only verb prompts with few-shot in-context examples and instructions. The traditional ***parameter-based CF*** caused by parameter updating *disappears*, but a new ***prompt-based CF*** *appears*.

This paper works in the *class-incremental learning* (CIL) setting of CL. In CIL, each task consists of one or more classes to be learned. In testing, no task identification information is given. To make CIL more realistic, we allow the arrivals of the training data from different tasks to intertwine, called ***blurry task boundaries***. That is, when a new task arrives, only a portion of its training data is available and the rest of the labeled samples of the task may come at any time later (Koh et al., 2022). This is referred to as ***online*** or ***streaming*** **CIL**, where the training data arrives as a continuous stream, and each training example is seen only once by the system (Guo et al., 2022).

Since an LLM has a **maximum input length** or **token limit**, it restricts the number of in-context examples that can be used in a prompt. This poses a major challenge for CIL because it needs to learn more and more tasks/classes. Thus, the ability to learn and store a minimum amount of knowledge for each class and to incrementally ***update the knowledge on the fly*** when additional data of old tasks becomes available is critical. This update also needs to ensure that the knowledge learned previously is not forgotten. Unlike existing approaches, in CLOB, we must do all these by using verbal prompts without touching any network parameters.

Our **second contribution** is to propose a simple and novel CIL method for CLOB, called CIS (*in-context **C**L via **I**ncremental **S**ummarization)*. CIS leverages the summarization capabilities of LLMs to encapsulate the knowledge about each class in a summary and incrementally learn and update the summary when some data from some old tasks

*To Appear in COLING-2025

[1]We use the term *black-box* because, as users, we can only interact with LLMs using verbal prompts or via an API.

come. CIS can also deal with the token limit issue for LLMs. Experimental results show significant accuracy gains compared to baselines.

Related Work. Overcoming CF is a key goal of CL. There are many existing approaches, e.g., *regularization-based approaches* (Kirkpatrick et al., 2016; Li et al., 2022; Liu et al., 2019), *replay-based approaches* (Rebuffi et al., 2017; Liu et al., 2021; Scialom et al., 2022; Qin et al., 2022; Huang et al., 2021) and *parameter isolation based approaches* (Ke et al., 2021, 2023; Serrà et al., 2018; Gururangan et al., 2022; Qin et al., 2022; Zhu et al., 2022; Geng et al., 2021; Madotto et al., 2021). Recent research in NLP also used parameter-tuning (Zhao et al., 2024) or a rehearsal-free modular (Wang et al., 2024b) to help knowledge transfer among tasks. A data-efficient CL paradigm for fine-tuning LLMs (Guo et al., 2024) and a prompt tuning-based CL method (Wang et al., 2024c) are also proposed. Reflexion (Shinn et al., 2023) and LLM as optimizer (Yang et al., 2024) update previous knowledge without knowledge retention. Voyager (Wang et al., 2024a) adds new skills to a library and retrieves it for each new task. We are different as we treat LLMs as black boxes and use only verb prompting for CL.

Many few-shot and instruction prompting techniques are proposed in (Wei et al., 2022; Zhou et al., 2023; Khot et al., 2022; Yao et al., 2023; Hao et al., 2023). However, they do not do prompting for CL.

2 Proposed CIS Method for CLOB

CIS learns a sequence of tasks 1, ..., T in CIL. Each task t has an input space $\mathcal{X}^{(t)}$, a class label space $\mathcal{Y}^{(t)}$ ($\mathcal{Y}^{(t)} \cap \mathcal{Y}^{(i)} = \emptyset$ for all $i \neq t$), and a training set $\mathcal{D}^{(t)} = \{(x_j^{(t)}, y_j^{(t)})\}_{j=1}^{n^{(t)}}$ drawn *i.i.d.* from $\mathcal{P}_{\mathcal{X}^{(t)}\mathcal{Y}^{(t)}}$. When task t arrives, only a subset $\mathcal{D}_{sub}^{(t)}$ of $\mathcal{D}^{(t)}$ is available for learning. The rest of the training samples $\mathcal{D}_{rest}^{(t)} = \mathcal{D}^{(t)} \backslash \mathcal{D}_{sub}^{(t)}$ may arrive at any time later. That is why we say that the arrivals of the data from different tasks intertwine, or the task boundaries are ***blurry***. Our goal is to learn a function $f : \cup_{t=1}^{T} \mathcal{X}^{(t)} \rightarrow \cup_{t=1}^{T} \mathcal{Y}^{(t)}$ to predict the class label of each test sample x. No task identifier is given for a test sample in testing.

CIS works in online CIL (Guo et al., 2022; de Masson d'Autume et al., 2019; Wang et al., 2020) with ***no training sample saved*** after being seen. Learning is done incrementally via verbal prompts over a black-box LLM whenever some training samples arrive.

2.1 CIS System

Due to the blurry task boundary and stream data, the following three scenarios may occur **during training** and the system needs to learn from any arrival data immediately and incrementally: **(1)** a new task t arrives with only a subset of the training data $\mathcal{D}_{sub}^{(t)}$. The system ***generates a summary*** of the data in each class and saves the summary. **(2)** Only a set S of new samples from the remaining samples of the old tasks arrives. CIS ***updates the summary*** of each class in S using samples of the class in S. **(3)** Both (1) and (2) occur. CIS performs the corresponding actions of (1) and (2). Below, we present the summary generator and the summary updator. An overview of CIS is given in Figure 1. All prompts used are given in Appendix A.

Summary generator (M_r): We also call this system the *reflector*. It is used only when a new task arrives. Since the data is discarded after it is learned, we propose to use the *verbal summary* to represent each class. Specifically, given the data from a task t, $\mathcal{D}_{sub}^{(t)}$, the LLM is prompted to generate a summary $s_i^{(t)}$ for each new class i in the task using the data of the class $\mathcal{D}_{sub,i}^{(t)}$, which is expected to represent the knowledge of the class,

$$s_i^{(t)} = M_r(\mathcal{D}_{sub,i}^{(t)}). \quad (1)$$

The resulting summary $s_i^{(t)}$ is saved. We call all saved summaries the ***summary base***.

Summary Updator (M_u)**.** As *new* data $\mathcal{D}_{new}^{(pre)}$ from *pre*vious tasks may come later at any time. The summaries of the classes involved in the new data need to be updated. This is done by the Updator. This update can potentially cause forgetting of existing knowledge in the summaries. We call this *prompt-based forgetting*. Specifically, for each class j (from a previous task p) involved in the new data $\mathcal{D}_{new}^{(pre)}$, its summary $s_j^{(p)}$ is updated with the new data of class j, $\mathcal{D}_{new,j}^{(pre)}$. This incremental summarization is also done via prompts.

$$\hat{s}_j^{(p)} = M_u(s_j^{(p)}, \mathcal{D}_{new,j}^{(p)}). \quad (2)$$

where $p \in pre$. The updated summary, $\hat{s}_j^{(p)}$, replaces the original summary $s_j^{(p)}$ in the summary base. Some summary examples are given in Appendix B.

Solver (M_s) **for Testing.** Due to the prompt length/token limit of LLMs and the fact that a test

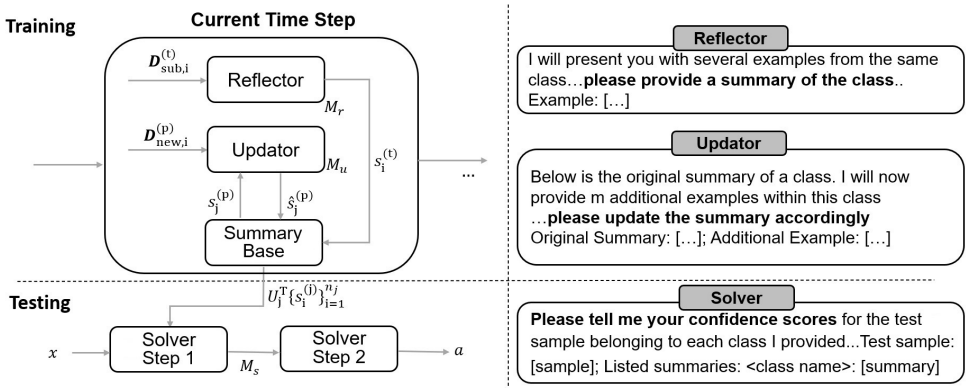

Figure 1: **(1) Left:** Overview of CIS in CLOB. **(2) Right:** Prompts used in each component of learning. Full prompts can be found in Appendix A. Some example summaries are given in Appendix B.

sample x can be from any learned class, testing is done in two steps. In Step 1, we divide all classes into multiple chunks. Each chunk consists of summaries of k classes in the summary base (no saving of any of their data) that can fit within the length limit of the LLM. We prompt the LLM to generate its confidence that x belongs to each class in the chunk j, $C_j = \cup_{i=1}^{k}\{s_i\}$, where s_i is the summary of class i in the chunk (task information is not needed),

$$conf_j = M_s(x, C_j). \tag{3}$$

In Step 2, the system first identifies the top k classes with the highest confidence values from all the chunks in Step 1, and then prompts the LLM again with only the resulting k classes, as before, to select the class with the highest confidence for x.

Theoretical Justification. It has been shown theoretically that good within-task prediction (WP) and good out-of-distribution (OOD) detection of each task within the tasks that have been learned so far is *necessary* and *sufficient* conditions for good CIL (Kim et al., 2022, 2023). In our approach, since we represent each class with its summary in the classification, effectively, each task has only one class at the inference time. Then, WP probability is 1 and CIL depends only on OOD detection of each class. Since each class's summary describes each class's key content, anything that does not belong to the class is dissimilar to the summary, which means the summary helps achieve a good OOD detention effect for each class. Although we don't use OOD detection in the final classification, the LLM is probably making some kind of similarity comparison between each test sample and each class summary.

3 Experiments

Datasets. We use four datasets. **(1) Banking-77** with 77 classes (Casanueva et al., 2020), **(2) CLINC-80** (Larson et al., 2019) with 80 classes from 10 domains, **(3) DPpedia-14** (Lehmann et al., 2014) with 14 classes, and **(4) Reuters-14** (Lewis, 1997) with the top-14 most frequent classes. Due to budget constraints and the use of OpenAI's paid GPT-3.5 API, we limited our study to 80 of the 150 classes in the CLINC dataset. However, CIS is capable of handling all 150 classes. Additionally, we focus on classification datasets since CIL is not well-suited for various NLP tasks like summarization, question-answering, and sentiment analysis.

For training, we select 7 random samples per class for all datasets. 7 was chosen to balance the need for variability in our settings with cost considerations. ***Each task consists of 2 classes***, except for the last task of Banking-77, which has only one class. For testing, we use 50 samples per class to save money. Tasks are ordered randomly, and the arrival times of new samples are also randomized in the Blurry setting. We perform 3 random runs for each experiment and report the average accuracy.

Blurry Setting. It is denoted by 'M / N *Blurry*', where $M + N = 7$, M is the number of training samples per class used when a task first appeared and N is the number of additional training samples per class that randomly appear later. Note that although the total number of training samples per class is 7 in this paper due to the budgetary constraint, this blurry setting allows CIS to handle any number of training samples per class and any possible M and N values.

Baselines. We compare CIS with several baselines: **(1)** EWC, a regularization-based algorithm to

address CF (Kirkpatrick et al., 2017); **(2)** LAMOL, a pseudo-replay method that generates data from previous tasks using GPT-2 (Sun et al., 2020); **(3)** VAG, a state-of-the-art parameter-updating CIL method (Shao et al., 2023) that fine-tunes the BART (Lewis et al., 2020). None of these baselines can work with the blurry setting. Our primary baseline is VAG, as it outperforms other CIL methods. We can only use its original implementation as it does not work with more advanced LLMs like Llama.

We also include **(4)** 'Joint: zero-shot,' where we use only the class names but no examples in the prompt; **(5)** 'Joint: prompting,' where Joint means to combine the data of all classes in each dataset into a single task and use them in a single prompt; **(6)** 'Joint: fine-tuning,' similar to (5), but we use all available data from all classes in each dataset to fine-tune the LLM. Joint systems are not CL systems but the upper bounds of CL.

Large language models (LLMs): We experimented with three LLMs as black-box models: (1) *Mistral-7B-Instruct-v0.3*, (2) *Llama-3.1-8B-Instruct*, and (3) *GPT-3.5-turbo-instruct*. To ensure comparability, we use the same settings for each LLM. The temperature is set to 0 to make the output deterministic. Each **summary** is limited to 3 sentences with less than 100 tokens.

Ablations. **(1)** Different M/N combinations within CIS; **(2)** '7 Non-blurry,' where all 7 samples per class for a task are seen simultaneously, representing the batch CIL approach with clear task boundaries; **(3)** CIS (classify), where, in Step 2 of CIS, we ask the LLM to directly output the predicted class rather than the confidence.

Implementation details. Due to the token limit of GPT-3.5, we split Banking-77 into 3 chunks of sizes 26/26/25 and CLINC-80 into 4 chunks of sizes 20/20/20/20. For the other two datasets, a single chunk suffices. In testing in Step 2, we select *top k* classes for each test sample, with $k = 5$. We also tried other k values with $k = 5$ being the best.

Different prompts may give different results, similar to how different hyper-parameters may affect deep learning models. We choose the most effective from several prompting strategies on dataset CLINC and DBpedia and apply them to all datasets. This shows the robustness of our approach, as we did not finetune the prompts for each dataset. The prompts used are given in Appendix A.

3.1 Evaluation Results and Analysis

CIS Ablations	3/4-Blurry	4/3-Blurry	5/2-Blurry	7 Non-blurry
Banking-77				
CIS (Mistral)	66.47 ±5.88	67.42 ±3.74	69.54 ±4.68	67.91 ±1.91
CIS (Llama)	78.75 ±1.68	79.23 ±2.50	77.77 ±1.49	79.93 ±1.74
CIS (GPT)	85.20 ±2.02	85.26 ±2.12	85.58 ±1.91	85.85 ±2.06
CIS (classify - GPT)	**85.58** ±0.57	**85.48** ±0.60	**86.05** ±0.82	**86.37** ±1.45
CLINC-80				
CIS (Mistral)	75.33 ±6.75	78.71 ±6.28	77.54 ±7.54	77.17 ±8.14
CIS (Llama)	91.51 ±4.35	90.40 ±5.46	91.29 ±3.61	92.14 ±4.51
CIS (GPT)	**93.88** ±3.76	**93.53** ±3.84	**93.94** ±3.82	**94.22** ±4.25
CIS (classify - GPT)	89.99 ±5.37	89.35 ±6.14	89.43 ±4.85	90.93 ±5.92
DBpedia-14				
CIS (Mistral)	80.43 ±2.83	85.32 ±0.15	79.29 ±2.83	78.50 ±3.33
CIS (Llama)	**92.07** ±1.07	**92.26** ±0.76	**93.52** ±0.73	**92.95** ±0.76
CIS (GPT) (ours)	88.19 ±3.20	90.79 ±1.85	89.43 ±2.64	86.45 ±3.04
CIS (classify - GPT)	88.79 ±3.85	89.19 ±4.34	87.14 ±5.38	86.43 ±3.30
Reuters-14				
CIS (Mistral)	74.18 ±1.29	71.86 ±4.56	77.27 ±1.54	71.73 ±8.87
CIS (Llama)	**83.97** ±1.08	**84.61** ±1.24	83.27 ±1.85	**84.97** ±1.15
CIS (GPT)	82.02 ±0.75	82.29 ±2.41	**83.94** ±1.55	84.53 ±1.15
CIS (classify - GPT)	76.72 ±5.08	80.02 ±3.53	78.00 ±4.21	78.61 ±3.88

Table 1: Ablation results in *Last Accuracy*.

Ablation results. We report ablation results first as they are more interesting. We use the test accuracy after all tasks are learned in Table 1, called the **Last Accuracy**. *Average Incremental Accuracy* and *Forgetting Rate* are also commonly used, but they require accuracy after each task is learned. In the blurry setting, where task boundaries are not clearly defined, these metrics are hard to apply.

CIS is effective. In the 3/4-blurry, 4/3-blurry, and 5/2-blurry settings, the accuracy results are comparable to or even exceed (as seen with DBpedia) those obtained when all 7 training samples per class are presented simultaneously (7 Non-blurry).

Forgetting. Comparing the accuracy results of CIS with those of 7-Non-blurry helps quantify the amount of *prompt-based forgetting* resulting from incremental summarization. We observe minimal forgetting. For example, for Banking-77 with GPT, forgetting is only about 0.27-0.65%; for DBpedia-14 with GPT, incremental summary updating (M/N-Blurry) even outperforms seeing all

	CIS (Llama)		**Joint (Llama)**				**CL Baselines**					
	3/4-Blurry	4/3-Blurry	Zero-shot	Prompting	Fine-tuning		EWC		LAMOL		VAG	
	7 samples	7 samples	no sample	7 samples	7 samples	full data	7 samples	full data	7 samples	full data	7 samples	full data
Banking-77	78.78 ±1.68	79.23 ±2.50	50.22 ±0.00	87.92 ±0.60	69.39 ±0.17	91.19 ±0.08	2.14 ±0.35	9.09 ±0.84	3.50 ±0.04	33.43 ±0.18	36.25 ±3.80	55.19 ±1.54
CLINC-80	91.51 ±4.35	90.40 ±5.46	80.67 ±0.00	95.10 ±2.51	91.18 ±0.46	97.92 ±0.06	1.14 ±0.33	8.26 ±0.76	17.60 ±0.19	52.20 ±0.09	64.75 ±0.69	80.68 ±0.72
DBpedia-14	92.07 ±1.07	92.26 ±0.76	93.36 ±0.00	90.50 ±0.40	93.74 ±0.11	99.00 ±0.00	6.55 ±0.73	23.14 ±1.55	0.70 ±0.14	28.61 ±0.02	55.36 ±3.30	56.58 ±1.22
Reuters-14	83.97 ±1.08	84.61 ±1.24	92.55 ±0.48	77.82 ±2.99	82.64 ±0.33	92.55 ±0.48	7.70 ±0.70	12.79 ±0.14	0.95 ±0.07	29.93 ±0.17	44.08 ±0.27	58.71 ±1.92

Table 2: *Last Accuracy* of CIS and baselines (2 classes per task for each dataset). CIS's results are copied from Table 1.

data at once (7 Non-blurry) by 2-3%. This may be because LLMs are sensitive to noises, causing in-context learning to be more easily distracted when seeing more samples (Shi et al., 2023).

Confidence vs. classification. In general, using classification in step 2 (**CIS (classify)**) gives poorer results, though it is slightly better in some cases.

Random order of classes in inference. Zheng et al. (2024) showed that LLMs are sensitive to the order of options. However, CIS is much less impacted by this because we ask for confidence scores, not for a class. See details in Appendix C.

CIS outperforms baselines (Table 2). Here, we use Llama 3.1 as we cannot fine-tune or do Joint prompting on GPT-3.5 due to its small token limit. Compared to the state-of-the-art baseline VAG, CIS dramatically outperforms it with only 7 samples per class. Even when VAG is trained with the full dataset, its performance is considerably below that of CIS. We are aware that this comparison is somewhat unfair, as VAG could not work with recent LLMs. However, our CIS is not far from the results of *Joint prompting* and *Joint fine-tuning* using all 7 samples per class, which are often regarded as the upper bounds of CIL. Joint prompting is actually weaker than CIS on the last two datasets. Joint fine-tuning is only better with the full dataset.

4 Conclusion

This paper made two contributions: (1) proposing a new CL paradigm, called CLOB, which works with black-box LLMs using only verbal prompts. (2) proposing a novel CL method CIS based on incremental summarization. Evaluations show that CIS not only has almost no CF but also outperforms baselines by a large margin.

We believe that there are two main reasons for the strong results. First, there is zero model parameter-based catastrophic forgetting, which the baselines have, because they require training or fine-tuning the full or part of the network for each new task. This is a significant advantage of LLMs. By aiming to cover comprehensive knowledge, LLMs can ensure that no essential information is overlooked during summary updates, effectively mitigating the risk of forgetting caused by updating summaries. Second, by using very compact summaries, our classification process can access summaries of all classes or in chunks simultaneously, but the baselines do not have access to the representations of all tasks at the same time as they cannot use summaries. Note that the summary of each class can be seen as its representation. Although replay-based baselines have access to some raw data of earlier classes/tasks, the amount of the raw data is too small for the system to handle parameter-based forgetting.

The experiments conducted in this paper stay within the scope of category-based text classification, as our work focuses on the CIL setting, which requires learning an increasing number of new classes over time. Relation classification is also appropriate for the proposed setting and, thus, a potential future extension of this work.

5 Limitations

There exists a concern that if each document is too long to fit in the token limit of the LLM, it will be difficult to produce a summary for the document. In such a case, we may need to break the document into multiple chunks and summarize each chunk first. After that, we summarize the chunks of summaries. However, it is unclear whether the approach will be effective. To address this problem, we could also use long-context LLMs. However, at this point, we cannot draw any conclusion about using very long documents as we do not have such documents. New experiments will be needed to test using very long documents to find the most appropriate solution, leading to future work.

Another limitation is that the proposed method may be hard to apply to computer vision applications due to the use of summaries. It is unclear how to replace the text summary with some form of image summary. These all form interesting future research directions, which are worthy of exploration because the proposed approach is highly effective and has almost no forgetting, which has plagued the existing fine-tuning or adaptation-based CL.

6 Ethics Statement

We believe that our work has no ethical issues or risks as we are using public-domain datasets and our task is simply classification based on the class labels already provided in the datasets.

Acknowledgments

This work was supported in part by four NSF grants (IIS-2229876, IIS-1910424, IIS-1838770, and CNS-2225427) and a research contract from KDDI Research.

References

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.

Zhiyuan Chen and Bing Liu. 2018. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*.

Binzong Geng, Fajie Yuan, Qiancheng Xu, Ying Shen, Ruifeng Xu, and Min Yang. 2021. Continual learning for task-oriented dialogue system with iterative network pruning, expanding and masking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*.

Google. 2024. Gemini.

Yanhui Guo, Shaoyuan Xu, Jinmiao Fu, Jia Liu, Chaosheng Dong, and Bryan Wang. 2024. Q-tuning: Queue-based prompt tuning for lifelong few-shot language learning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2595–2622, Mexico City, Mexico. Association for Computational Linguistics.

Yiduo Guo, Bing Liu, and Dongyan Zhao. 2022. Online continual learning through mutual information maximization. In *International conference on machine learning*, pages 8109–8126. PMLR.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics.

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.

Zixuan Ke and Bing Liu. 2022. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*, 34.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *The Eleventh International Conference on Learning Representations*.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *CoRR*, abs/2210.02406.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. 2022. A theoretical study on solving continual learning. In *Advances in Neural Information Processing Systems*.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, and Bing Liu. 2023. Learnability and algorithm for continual learning. *arXiv preprint arXiv:2306.12646*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*.

Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. 2022. Online continual learning on class incremental blurry task configuration with anytime inference.

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.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2014. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6.

David D. Lewis. 1997. Reuters-21578 text categorization test collection.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Dingcheng Li, Zheng Chen, Eunah Cho, Jie Hao, Xiaohu Liu, Fan Xing, Chenlei Guo, and Yang Liu. 2022. Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Qingbin Liu, Xiaoyan Yu, Shizhu He, Kang Liu, and Jun Zhao. 2021. Lifelong intent detection via multi-strategy rebalancing. *CoRR*, abs/2108.04445.

Tianlin Liu, Lyle Ungar, and João Sedoc. 2019. Continual learning for sentence representations using conceptors. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual learning in task-oriented dialogue systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. ELLE: Efficient lifelong pre-training for emerging data. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2789–2810, Dublin, Ireland. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. icarl: Incremental classifier and representation learning. In *CVPR*.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Continual-t0: Progressively instructing 50+ tasks to language models without forgetting. *CoRR*, abs/2205.12393.

Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*.

Yijia Shao, Yiduo Guo, Dongyan Zhao, and Bing Liu. 2023. Class-incremental learning based on label generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1263–1276, Toronto, Canada. Association for Computational Linguistics.

Zhenmei Shi, Junyi Wei, Zhuoyan Xu, and Yingyu Liang. 2023. Why Larger Language Models Do In-context Learning Differently?

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. Lamol: Language modeling is all you need for lifelong language learning. In *ICLR*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024a. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schuetze. 2024b. Rehearsal-free modular and compositional continual learning for language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 469–480, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024c. InsCL: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 663–677, Mexico City, Mexico. Association for Computational Linguistics.

Zirui Wang, Sanket Vaibhav Mehta, Barnabás Póczos, and Jaime Carbonell. 2020. Efficient meta lifelong-learning with limited memory. In *EMNLP*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In

The Twelfth International Conference on Learning Representations.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11641–11661, Bangkok, Thailand. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual prompt tuning for dialog state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland. Association for Computational Linguistics.

A Prompts Used in CIS

A.1 Reflector

I will present you with several examples from the same class. Based on these examples, please provide a summary of the class in no more than *3* sentences. Note that your summary should not include any of the examples.
Examples: [list of examples]

A.2 Updator

Below is the original summary of a class. I will now provide *m* additional examples within this class. Based on these, please update the summary accordingly. Ensure that the updated summary does not exceed *3* sentences.
Original summary: [summary text]
Additional examples: [list of examples]

A.3 Solver - Classification

Please classify the provided test sample into one of the listed classes based on the summaries provided for these classes. The summaries are formatted as '<class name>: [summary].' Your response should include only one class name as the answer. This name must exactly match one of the classes given. Do not include the original test sample in your response.
Test sample: [sample]
Listed summaries: <class name>: [summary]

A.4 Solver - Confidence

Please tell me your confidence scores for the test sample belonging to each class I provided. I will present you a list of summaries for these classes as your reference. The summaries are formatted as '<class name>: [summary].' Your response should include only the class names and the corresponding confidence scores in decimal as the answer. The name must exactly match one of the classes given. Do not include the original test sample in your response.
Test sample: [sample]
Listed summaries: <class name>: [summary]

B Summary Examples

Routing: "The category is requesting routing numbers for various banks and accounts. These numbers are used to identify the specific bank and account when making transactions. Customers can find their routing numbers through their bank's website or by contacting their bank directly."

Transactions: "The category is about requesting information on past transactions, including the ability to list recent transactions, check on a specific transaction, and view transaction history within a certain time frame."

Min_payment: "The category is about minimum payments for various bills, such as truck payments, M&T bills, power bills, credit card bills, and phone bills. Customers are seeking information on the minimum amount they need to pay for each bill."

C Random Order Issue

As discussed, the order of classes provided to the LLMs affects the results. The top-k confidence scores are no exceptions.

To investigate this issue, we conducted experiments using DBpedia-14 dataset with our CIS system. The experiments using Llama-3.1-8B show the following results:

3/4-blurry: 90.32 $\pm$0.76
4/3-blurry: 90.18 $\pm$3.89
5/2-blurry: 90.18 $\pm$1.57
Non-blurry: 90.57 $\pm$1.41

The results show a slight decline compared to the original results with Llama, with higher standard deviations. We also ran similar experiments with 3/4-blurry and 4/3-blurry using GPT-3.5, which are reported below:

3/4-blurry: 92.41 $\pm$0.56
4/3-blurry: 93.56 $\pm$0.81

In this case, the standard deviations are low, indicating that the class order has a minimal effect on the final classification results. We also notice that these new results are better than those reported in the paper, but this is possibly due to the update of OpenAI's system.

We investigated the outputs and found that though the top-3 output classes may vary in confidence scores or order, they still appear as the top-3 predictions. This allows our system CIS to pick the correct top-3 class. In some cases, the correct class is no longer the top-1 class, but it is still in the top-3 list.

Below, we show three test samples with their top-3 output classes ranked by confidence.

Test sample 1:

St Nicholas' Church Tuxford is a Grade I listed parish church in the Church of England in Tuxford.

Original top-3 output classes from step 1 in the experiments for the paper:
Top-1: building; 2: village; 3: natural_place

Top-3 output classes from the new experiments with 3 random orderings of classes:
1. Top-1: building; 2: village; 3: natural_place
2. Top-1: building; 2: village; 3: natural_place
3. Top-1: building; 2: village; 3: natural_place

Test sample 2:

Chris Phillips (born March 9 1978) is a Canadian professional ice hockey player for the Ottawa Senators of the National Hockey League (NHL). He has been a member of the Ottawa Senators for his entire career which began with the 1997–98 season. He also serves as their alternate captain and is regarded as a stay-at-home defenceman. The Senators drafted him first overall in the 1996 NHL Entry Draft. He was raised in Fort McMurray Alberta.

Original top-3 output classes from step 1 in the experiments for the paper:
Top-1: athlete; 2: office_holder; 3: company

Top-3 output classes from the new experiments with 3 random orderings of classes:
1. Top-1: athlete; 2: office_holder; 3: company
2. Top-1: athlete; 2: company; 3: office_holder
3. Top-1: athlete; 2: office_holder; 3: company

Test sample 3:

The Himalayan agama (Paralaudakia himalayana) is an agamid lizard found in Central Asia and South Asia.

Original top-3 output classes from step 1 in the experiments for the paper:
Top-1: animal; 2: natural_place; 3: plant

Top-3 output classes from the new experiments with 3 random orderings of classes:
1. Top-1: plant; 2: animal; 3: natural_place
2. Top-1: animal; 2: natural_place; 3: village
3. Top-1: animal; 2: natural_place; 3: plant