Online Continual Knowledge Learning for Language Models

Yuhao Wu[1][*] **Tongjun Shi**[1][*] **Karthick Sharma**[2][†] **Chun Wei Seah**[1] **, Shuhao Zhang**[3][‡]

[1]Singapore University of Technology and Design; [2]University of Sri Jayewardenepura;
[3]Nanyang Technological University

{yuhao_wu, chunwei_seah, tongjun_shi}@sutd.edu.sg
en93899@sjp.ac.lk
shuhao.zhang@ntu.edu.sg

Abstract

Large Language Models (LLMs) serve as repositories of extensive world knowledge, enabling them to perform tasks such as question-answering and fact-checking. However, this knowledge can become obsolete as global contexts change. In this paper, we introduce a novel problem in the realm of continual learning: *Online Continual Knowledge Learning* (**OCKL**). This problem formulation aims to manage the dynamic nature of world knowledge in LMs under real-time constraints. We propose a new benchmark and evaluation metric designed to measure both the rate of new knowledge acquisition and the retention of previously learned knowledge. Our empirical evaluation, conducted using a variety of state-of-art methods, establishes robust baselines for OCKL. Our results reveal that existing continual learning approaches are unfortunately insufficient for tackling the unique challenges posed by OCKL. We identify key factors that influence the trade-off between knowledge acquisition and retention, thereby advancing our understanding of how to train LMs in a continually evolving environment.

1 Introduction

Large Language Models like LLaMa2 (Touvron et al., 2023) and GPT-3 (Brown et al., 2020) have demonstrated remarkable proficiency in Knowledge Intensive Language Tasks (KILT) (Petroni et al., 2022). Despite their success, a significant limitation is their struggle to adapt to rapidly evolving global knowledge, as highlighted by their performance in the LAnguage Model Analysis (LAMA) framework (Petroni et al., 2019). The conventional method of augmenting LLMs with an external, continually updated knowledge base introduces latency and reduces the models' capac-
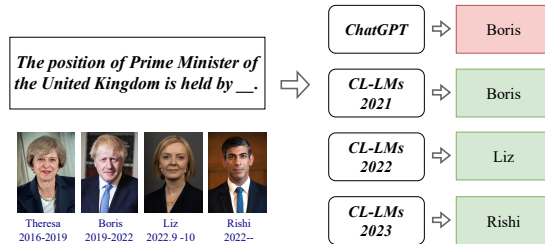

Figure 1: Static models like T5 (Raffel et al., 2020), due to their knowledge cutoff, cannot provide timely answers to time-variant knowledge.

ity for real-time contextual understanding (Raffel et al., 2019).

As illustrated in Figure 1, current LLMs often yield outdated or incorrect answers when confronted with time-sensitive questions. Addressing these constraints necessitates a framework that enables real-time, internal knowledge updating in LLMs. While some studies have begun addressing this need (Mitchell et al., 2022; De Cao et al., 2021; Zhu et al., 2021; Dhingra et al., 2022; Jang et al., 2022), they have not fully resolved the challenges associated with rapid LLM updating.

To tackle these issues, we propose the *Online Continual Knowledge Learning* (OCKL) framework. OCKL differentiates itself from existing Continual Knowledge Learning (CKL) by prioritizing immediate, ongoing internal knowledge updates within significantly shorter timeframes, ranging from days to mere seconds. This approach, divergent from offline methods, requires single-pass updates (epoch=1) due to the high velocity and volume of incoming data (Aljundi et al., 2019). OCKL presents a novel challenge in LM updates, for which a robust benchmark is essential yet currently absent.

To evaluate LMs within our framework, we introduce two new metrics: KNOWLEDGE ACQUISITION RATE (KAR) and KNOWLEDGE GAP (KG), assessing text processing efficiency, knowl-

---

edge acquisition, retention, and real-time learning variances. Additionally, we examine the suitability of existing continual learning methods for OCKL, exploring model architectures and training methodologies that mitigate knowledge "forgetting" (He et al., 2021; Hu et al., 2022). We also utilize advanced coreset selection techniques to enhance knowledge acquisition and reduce computational load (Yoo and Kweon, 2019; Li et al., 2022). This comprehensive approach establishes a robust baseline for the OCKL framework and lays the foundation for future research aimed at optimizing the adaptability and efficiency of LMs in real-time data environments.

Contributions. This paper presents multi-faceted contributions:

1. **Novel Framework and Benchmark**: The introduction of **OCKL**, a new framework designed to better understand the shortcomings of current LM updating practices facing high velocity data streams.

2. **Baseline Approaches and Metrics**: Innovative metrics like Knowledge Acquisition Rate and Knowledge Gap are introduced for nuanced evaluation of LMs' learning conditions.

3. **Empirical Insights and Challenges**: Exhaustive analyses within the OCKL benchmark reveal crucial insights and challenges. For instance, rehearsal methods consistently outperform other approaches despite buffer inefficiencies.

2 Related Work

Language Models (LMs), particularly when augmented with external knowledge sources like Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Guu et al., 2020; Ram et al., 2023), aim to stay current by updating these sources or querying the internet for the latest information. However, these methods grapple with challenges, notably "hallucination", where models generate incorrect information as factual, a problem amplified in larger LMs (Zhang and Choi, 2021; Longpre et al.).

While retraining language models from scratch with updated data offers a comprehensive solution, it is computationally intensive and environmentally unsustainable (Patterson et al., 2021).
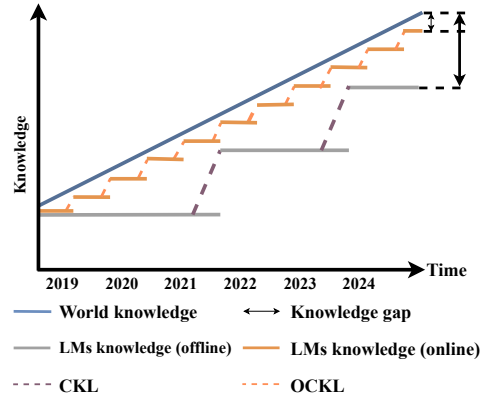

Figure 2: Comparative Analysis of Knowledge Synchronization in Offline (CKL) and Online (OCKL) Learning Paradigms.

A more feasible alternative, incremental pretraining on smaller, updated corpora, faces the risk of "catastrophic forgetting", where newly acquired knowledge supersedes prior learning (De Lange et al., 2021). Existing mitigation strategies such as regularization (Buzzega et al., 2020; Caccia et al., 2021), rehearsal-based methods (Chaudhry et al., 2019; Aljundi et al., 2019; Yoon et al., 2021; Chrysakis and Moens, 2020; Chaudhry et al., 2021), and parameter expansion (Rusu et al., 2016; Hu et al., 2018; Mallya and Lazebnik, 2018) are primarily designed for offline contexts, often requiring multiple passes through data and thus not aligning with real-time, online scenarios.

In the specific realm of CKL for LMs, various approaches like those proposed by (Jang et al., 2022; Luu et al., 2022; Dhingra et al., 2022; Margatina et al., 2023) have been developed to update and retain knowledge over time. However, similar to the broader CL methodologies, these CKL strategies are predominantly offline, iterating over the same data multiple times and lacking in their ability to swiftly adapt to the dynamic and evolving nature of streaming data contexts. The limitations of these traditional methods in the context of our framework are further discussed in Appendix 6.1.

3 Online Continual Knowledge Learning

In this section, we will provide the illustrations of the task formulation, conduct dataset analysis, and present the proposed metrics for online continual knowledge learning.

3.1 OCKL Formulation

Motivation and Challenges The evolution of real-world information necessitates the continual update of LMs. Despite the capacity of LMs to act as reservoirs of global knowledge through pretraining on extensive corpora, this knowledge is susceptible to obsolescence. This leads to the formulation of a novel Continual Learning problem–*Online Continual Knowledge Learning*, which aims to harmonize the acquisition of new knowledge with the preservation of existing information under real-time constraints. Figure 2 contrasts the shortcomings of traditional Continual Knowledge Learning methods (Jang et al., 2022) with the promise of OCKL. CKL allows LMs to accumulate extensive knowledge but does not offer a mechanism for real-time adaptation. OCKL seeks to rectify this by facilitating frequent, incremental updates to the LM, enabling it to adapt to dynamically changing information landscapes.

Objectives and Scope Against the challenges intrinsic to OCKL, we outline two key objectives that guide this study: 1) to establish a real-time evaluation framework with a comprehensive suite of metrics to measure the rate of new knowledge acquisition and the effectiveness of existing knowledge retention in LMs; 2) to refine LM training protocols with a focus on real-time knowledge updates. We employ coreset selection techniques to optimize both speed and efficiency in knowledge acquisition.

Formal Framework The continual updating of LMs is represented as transitions between model states: $\theta_{t-1} \rightarrow \theta_t$. Knowledge updates are sourced from a data stream $\{X_t^i\}_{i=1}^n \sim D_t$, and factual verification is performed via a QA stream $\{(Q_t^i, A_t^i)\}_{i=1}^n \sim D_t$. At each time step t, two activities are conducted:

1. Updating: the model θ_t is updated using $\{X_t^i\}_{i=1}^n \sim D_t$.

2. Evaluating: the model θ_t is evaluated through the QA stream, generating answers $\{\widetilde{A_t^i}\}_{i=1}^n$ for questions $\{Q_t^i\}_{i=1}^n$.

3.2 Efficiency-Driven Coreset Selection in OCKL

Coreset selection is a strategy aiming at isolating a representative and informative subset from a comprehensive training dataset. Its utility lies in enabling models to approximate the performance achieved when trained on the complete dataset. In the context of rehearsal-based continual learning, coreset selection has previously been employed as a mechanism to counteract catastrophic forgetting through the retention of a knowledge buffer (Tiwari et al., 2022; Yoon et al., 2021).

In this work, we enhance the application of coreset selection, specifically targeting the acceleration of OCKL. We define a subset $S \subseteq \{1, 2, \dots, n\}$, which consists of indices that select specific elements X_t^i from the original dataset. Consequently, the chosen subset from $\{X_t^i\}_{i=1}^n$ is denoted as $\{X_t^i : i \in S\}$. By selecting pertinent data samples, our approach seeks not only to preserve antecedent knowledge but also to alleviate the computational demands inherent in the training process, thereby contributing to more efficient online learning systems. The ensuing discussion will explore three specific methods for coreset selection—Random Sampling, K-Center selection, and Model-Based (Yoo and Kweon, 2019) selection—each with its advantages and limitations in the context of OCKL. These methods are illustrated further in Appendix 6.2.

3.3 Dataset Characteristics and Methodology

Overview To facilitate the construction of an adaptive Language Model capable of processing time-sensitive knowledge, our dataset is divided into two primary components: the **Knowledge Stream** and the **QA Stream**. These streams are generated based on Wikidata's exhaustive knowledge base, spanning from 2019 to 2023. The dataset features both time-invariant and time-variant facts, parsed and structured through SLING*.

Data Preprocessing & Construction The primary data source, a raw Wikidata dump, is initially mapped onto data frames. These are subsequently refined to distill facts related to specific entities. Facts are then categorized based on their property attributes, separating them into time-invariant and time-variant types. A detailed visual representation of this preprocessing methodology is available in Appendix 6.3. We construct **Knowledge Stream** by integrating the refined facts with corresponding property templates and generating a **corpus** along with an associated date for each en-

---

*A natural language frame semantics parser by Ringgaard Research

try. Likewise, the **QA Stream** is constructed in alignment with established research methodologies (Margatina et al., 2023; Dhingra et al., 2022), comprising the **query**, **answer**, and **date**.

Data Statistics Analysis Following the construction of the data streams, we present key statistical attributes. The Cumulative Distribution Function (CDF) of the token and date changes over time is shown in Appendix 6.4. The Knowledge Stream contains 94,568 data points, with average text and token lengths of 70.1 and 12.7, respectively. It comprises 37.6% time-invariant facts and 62.4% time-variant facts. For the latter, the average text and token lengths stand at 74.8 and 13.7, respectively. For a comparative evaluation, we further refine our initially constructed datasets to eliminate redundancies, ensuring that each fact is represented only once. This non-redundant version includes 1,929,045 texts, with average token and text lengths of 11.24 and 62.54, respectively. Notably, 99.97% of these texts are time-variant facts, while a mere 0.03% are time-invariant.

3.4 Evaluation Metrics

For a comprehensive and unbiased assessment of the OCKL problem, we have devised two novel performance metrics and refined the standard BWT (Backward Transfer) and FWT (Forward Transfe) approaches, together with EM (Exact Match). The details of EM, BWT and FWT are provided in Appendix 6.7.

Knowledge Acquisition Rate (KAR) The assessment of a model's knowledge acquisition rate is nontrivial due to its dependency on various factors. Multiple factors contribute to the varying speeds at which different methods process tokens. Key contributors include the volume of data processed (rehearsal-based), differences in the number of trainable parameters (expansion parameter-based), and additional computations required (regularization-based). These intricacies are further compounded by divergences in knowledge acquisition and retention capabilities across different learning algorithms. To navigate these complexities, we introduce a comprehensive formula for quantifying the effective learning rate of a model. We leverage previously defined metrics: Forward Transfer quantifies the incorporation of new knowledge, while Backward Transfer serves as an index of knowledge retention or forgetfulness. The sum of these metrics provides a net knowledge gain per learning iteration. Accordingly, the genuine knowledge acquisition rate can be calculated using the following equation:

$$\text{KAR} = (\text{FWT} + \text{BWT}) \times \frac{\text{Total Tokens}}{\text{Training Time}} \quad (1)$$

Knowledge Gap (KG) The Knowledge Gap metric provides a tangible measure of knowledge dynamics within a Language Model by employing vector representations and distance metrics. We leverage the concept of parametric knowledge, which asserts that an LM encodes world knowledge within its parameters (Izacard et al., 2022; Jang et al., 2022) for our evaluation.

Operationalization: As a case study, consider the T5 model. Its embedding layer vectorizes input data before the encoding phase, offering a representation of "world knowledge" denoted as T_w. Further, the final encoder layer extracts hidden states from the input query, encapsulating "model knowledge" denoted as T_m.

Metric Formulation: We compute the KG metric as a similarity distance between two sequences, T_1 and T_2, which can represent either world or model knowledge. The framework allows for the following configurations:

1. When $T_1 = T_w$ and $T_2 = T_m$, the Euclidean distance $d(T_1, T_2)$ quantifies the alignment between model knowledge and world knowledge. A smaller value indicates better fidelity.

2. When $T_1 = T_{m_{(t-1)}}$ and $T_2 = T_{m_{(t)}}$, where t denotes the time stamp, $d(T_1, T_2)$ evaluates the ***forgetting*** behavior, capturing knowledge attrition during online learning.

3. When $T_1 = T_{m_{(t+1)}}$ and $T_2 = T_{m_{(t)}}$, $d(T_1, T_2)$ measures the ***updating*** behavior, assessing knowledge acquisition or refinement during online learning.

By applying the KG metric in these configurations, we offer an assessment of the model's capacity for knowledge retention and acquisition, which is particularly vital for evaluating models deployed in dynamic, real-world scenarios.

3.5 Continual Learning Method

In this paper, we have selected a variety of CL methods for OCKL, specifically including:

Vanilla refers to a standard setup utilized for additional training, as discussed by Gururangan et al. (2020).

RecAdam (Chen et al., 2020) falls into the category of regularization methods. It differs from traditional regularization methods like EWC (Kirkpatrick et al., 2017) by imposing a more stringent independence assumption among model parameters.

Knowledge Distillation (Gou et al., 2021) is to speed up model inference by minimizing the representation gap between two models.

Mix-Review (He et al., 2021) falls into the category of rehearsal methods, which assumes access to the initial pretraining corpus and mixes in random subsets of the initial pretraining data during continued pertaining.

LoRA (Hu et al., 2022) is a parameter-expansion technique presented by Hu et al. (2022). Instead of updating the original LM parameters, it introduces trainable rank-decomposition matrices in each layer for continued pertaining.

K-Adapter (Wang et al., 2021) is another parameter expansion method that freezes the original LM parameters while adding a *k* number of new layers, namely *adapters*, that are updated during continued pre-training.

Modular (Jang et al., 2022) is a novel parameter expansion strategy tailored for encoder-decoder models. The original pretrained encoder is preserved, but a supplementary, randomly initialized encoder is introduced for updates during the next phase of training.

4 Experiments

In this section, we will begin by presenting our experimental setup followed by an in-depth discussion of the outcomes from our experiments.

4.1 Experimental Setup

We conducted extensive experiments using the T5 encoder-decoder model (Raffel et al., 2019). This Language Model, with around 737M parameters, was originally pretrained on the C4 dataset from April 2019 and the Wikipedia dataset from May 2020. For our analysis of the OCKL benchmark, we established several baseline methods. These methods are categorized into three main groups: *regularization* (i.e., RecAdam, KD as Knowledge-Distillation), *rehearsal* (i.e., Mix-Review), and *parameter-expansion* (i.e., LoRA, Kadapter, and

Modular). The detailed descriptions of the CL setups and our hyper-parameters used for each method are provided in Appendix 6.5 and 6.6 respectively.

4.2 Experiments Result

4.2.1 Main Result

Table 1 presents the outcomes of our experiments carried out on the OCKL benchmark. We concentrated on three pivotal metrics: Exact Match (EM), Knowledge Gap (KG), and Knowledge Acquisition Rate (KAR). The T5 models underwent an initial pretraining phase using the C4 dataset, which entailed roughly 1 trillion token updates, along with Wikipedia data up until 2019. This served as our foundational model for a series of continuous learning tests.

We initiate our analysis using a dataset containing a considerable volume of redundant data, reflecting the redundancy commonly found in real-world textual streams. As delineated in Table 1 under the *Redundancy* subsection, the method named T5-Mix-Review excels in EM and KAR compared to alternative approaches. In contrast, the parameter expansion method e.g. T5-LoRA, performs admirably in CKL (Jang et al., 2022) but falters in OCKL. This performance drop could be ascribed to T5-LoRA's restricted trainable parameters, as it freezes the primary network parameters, limiting both retention and learning capabilities. In the *Redundancy-Free* section of Table 1, both the naive approach T5-Vanilla and the rehearsal-based method T5-Mix-Review exhibit efficacy in learning from the online knowledge stream, as evidenced by their high EM scores. This observation underscores the varying effectiveness of rehearsal methods in CL, parameter-expansion methods in CKL, and their overall limitations in our OCKL setup. The only exception is the rehearsal method, which surpasses the simple T5-vanilla approach. Although T5-KD demonstrates commendable performance, as highlighted by Jin et al. (2022), it lags in OCKL due to the time-intensive nature of distillation-based methods. To achieve superior performance, one must rely on the teacher model's outcomes. Furthermore, as T5-KD loss influences the embeddings of both the teacher and student models directly, it registers the lowest score in the Knowledge Gap metric.

We extend our evaluation by scrutinizing various CL methods in Table 2, focusing on

Corpus	Method	# of Params (Trainable / Total)	EM (%)↑	KG↓	KAR↑
	T5-Initial	0M / 222M	0.50	91.601	–
	T5-Vannila	222M / 222M	<u>21.41</u>	91.792	<u>3997.07</u>
	T5-RecAdam	222M / 222M	20.48	91.634	3282.73
REDUNDANCY	T5-Mix-Review	222M / 222M	**23.26**	91.802	**4264.82**
	T5-LoRA	113M / 223M	16.42	<u>91.453</u>	2031.78
	T5-K-Adapter	141M / 251M	15.21	91.601	1711.34
	T5-Modular	198M / 307M	13.61	91.601	980.80
	T5-KD	222M / 222M	15.02	**89.38**	669.04
	T5-Initial	0M / 222M	0.34	16.571	–
	T5-Vannila	222M / 222M	<u>15.09</u>	16.769	**4471.01**
	T5-RecAdam	222M / 222M	14.03	16.729	3240.77
REDUNDANCY-FREE	T5-Mix-Review	222M / 222M	**15.38**	16.697	<u>4105.50</u>
	T5-LoRA	113M / 223M	14.53	16.768	3363.83
	T5-K-Adapter	141M / 251M	12.43	**16.571**	3699.39
	T5-Modular	198M / 307M	13.39	16.571	3343.83
	T5-KD	222M / 222M	7.09	**12.480**	267.30

Table 1: Redundant and redundant-free datasets results. Different continual learning strategies come with their own set of limitations due to the constraints set by online learning and the dense emergence of new information in data streams. ↓ represents the lower the better; ↑ represents the higher the better. The best results for each task and metric are shown in bold. The second-best results are underlined with a horizontal line.

Method	EM ↑	BWT ↑	FWT ↑	KG ↓
T5-Vanilla	15.09	-4.18	9.27	16.769
T5-RecAdam	14.03	-4.48	8.98	16.729
T5-Mix-Review	**15.38**	-4.49	**9.28**	16.697
T5-LoRA	14.53	-4.34	8.46	16.768
T5-Kadapter (k=2)	12.43	-3.59	7.58	16.576
T5-Modular	13.39	-3.96	7.97	16.570
T5-KD	7.09	**-2.82**	4.96	**12.480**

Table 2: Comparison of knowledge forgetting and updating for different methods.

Method	# of Params (Trainable / Total)	EM (%)↑	KG↓	KAR↑
T5-initial	0M / 737M	0.49	7.11	-
T5-Vanilla	737M / 737M	16.19	7.02	2378.68
T5-RecAdam	737M / 737M	14.85	7.04	1840.27
T5-Mix-Review	737M / 737M	16.18	7.01	2658.96
T5-LoRA	403M / 738M	**17.17**	7.12	**2761.98**
T5-Kadapter (k=2)	427M / 762M	6.71	7.11	1542.39
T5-Modular	438M / 773M	8.55	7.11	1765.85
T5-KD	737M / 737M	6.30	**5.50**	122.95

Table 3: The experiment results of T5-large model in the redundancy-free stream data.

redundancy-free stream data. Remarkably, as indicated in this table, both T5-Kadapter and T5-Modular methods—attributable to their frozen encoder components—register an identical KG metric when compared to their original model. It is evident from Table 2 that no single method excels across all metrics. Our analysis suggests that regularization methods outperform others in mitigating the phenomenon of knowledge forgetting in an online learning context. Conversely, parameter expansion methods uniformly show suboptimal performance, as reflected in their notably weaker Forward Transfer scores, possibly due to the constraints imposed by their set of trainable parameters. Rehearsal-based methods exhibit robust FWT results which is the determinant in their superior Exact Match performance. Of all the evaluated methods, the T5-KD approach yields the smallest Backward Transfer, aside from its relatively lower EM, which may imply its aptness for offline continual learning scenarios where Epoch > 1.

4.2.2 T5-large Experiment Result

Table 3 presents the experimental results for larger model scales in T5-large, and the outcomes were unexpectedly impressive to a certain extent. Most methods, including T5-Vanilla and T5-Mix-Review, produced results consistent with those shown in Table 1, maintaining high performance according to the FWT and KAR metrics. Interestingly, the performance of T5-loRA (Hu et al., 2022) significantly outperformed that of the T5-base experiments. Surprisingly, other methods for parameter expansion, such as T5-Kadapte and T5-Modular, did not yield corresponding improvements. This suggests that the enhancement achieved by loRA cannot be solely attributed to an increase in the number of trainable parameters. One potential explanation for this phenomenon may stem from LoRA's unique characteristics when contrasted with other parameter expansion techniques. LoRA's distinctiveness is manifested through the incorporation of trainable rank-decomposition matrices in every layer throughout the continuous pretraining process.

4.2.3 Impact of Coreset Selection Strategies on OCKL

To attain an optimal balance between training velocity and model performance, we investigate the influence of coreset selection on OCKL in both redundant and non-redundant data stream scenarios. The performances of distinct coreset selection strategies are elaborated in Table 4.

Redundancy In the redundant data streams, the K-Center method emerges as the most effective, whereas random sampling is demonstrably inferior. Both K-Center and the predictive network approach proposed by Yoo and Kweon (2019) leverage the repetitive nature of the redundant data stream. However, K-Center excels in isolating representative samples from the stream. It is noteworthy that none of the coreset methods manage

Corpus	Method	EM	BWT	FWT	KG	Forget	Update	KAR
	w/o selection	21.41	0.59	0.64	91.79	0.17	0.17	3997.07
REDUNDANCY	Random	9.44	0.16	0.16	91.48	0.20	0.20	1202.28
	K-Center	15.44	0.34	0.37	91.86	0.16	0.16	1769.58
	(Yoo and Kweon, 2019)	11.69	0.17	0.18	91.61	0.24	0.24	156.52
	w/o selection	15.09	-4.18	9.27	16.77	0.03	0.02	4471.01
REDUNDANCY-FREE	Random	12.28	-2.43	6.36	16.70	0.03	0.03	**5207.31**
	K-center	12.22	-2.51	6.49	16.69	0.03	0.03	4912.63
	(Yoo and Kweon, 2019)	9.82	-2.80	6.34	16.72	0.03	0.04	1639.60

Table 4: Comparison between different coreset selection methods with two stream settings. *Random* represents the random selection for each mini-batch.

Ratio	EM	BWT	FWT	DTW	KAR
r=0.25	9.60	-1.94	4.74	16.68	4369.71
r=0.50	12.22	-2.51	6.49	16.69	**4912.63**
r=0.75	14.13	-3.99	8.23	16.64	4604.69
r=1.00	15.09	-4.18	9.27	16.77	4471.01

Table 5: Varying different coreset selection ratios based on the best coreset selection method.

Method	EM	BWT	FWT	KG
T5-Vanillla	7.48	**-0.84**	4.07	40.53
T5-Recadam	10.05	-2.86	6.28	16.73
T5-Mixreview	11.00	-2.22	**6.83**	16.70
T5-LoRa	**11.06**	-1.87	6.31	16.69
T5-Kadapter	9.88	-1.96	6.60	**16.57**
T5-KD	2.13	-3.20	-0.94	40.57
T5-Modular	10.31	-1.85	5.77	**16.57**

Table 6: Comparison of knowledge forgetting and updating between different methods in the time limitation setup.

to enhance the knowledge acquisition rate in the online setting.

Redundancy-free Contrastingly, in the absence of data redundancy, simple random sampling surpasses other methods in terms of both training speed and final performance. This can be ascribed to a misalignment between the methodological underpinnings of K-Center and the predictive network approach and the intrinsic properties of our data stream, particularly the continuous influx of distinct and evolving data. In both redundant and non-redundant settings, the Exact Match score experiences a decline. However, the knowledge acquisition rate sees an improvement in the non-redundant case, while deteriorating under redundant conditions. This can be attributed to a greater rate of information assimilation relative to forgetting in a redundancy-free environment, even though the forgetting rate remains substantial. Furthermore, as delineated in Table 5, varying coreset selection ratios in the K-Center method are explored, revealing improved performance as the ratio increases.

4.2.4 Fast Knowledge Stream Arrival Rate

In this work, drawing upon Ghunaim et al. (2023), we established a scenario where the rate of incoming data stream surpasses the processing capacity of the current GPU cluster. This challenging yet achievable setting allowed us to assess the effectiveness of various continual learning strategies under time constraints. Unlike previous experiments focused on the continual learning capabilities of different methods, this investigation emphasizes a more practical scenario with time limitations. We enforced a uniform computation time across all methods. Consequently, any data exceeding this time frame was discarded.

The outcomes of this scenario are detailed in Table 6, where the time constraint is set at half the duration of Kadapter's processing time. This effectively means that, using Kadapter's processing speed as a benchmark, the model must omit half of the training data. Notably, LoRA and Mixreview outperformed under these conditions. Contrary to previous findings, LoRA exceeded Mixreview in performance, likely due to its inherently greater computational efficiency. Mix-Review exhibited outstanding FWT abilities and notable success in EM performance. Thus, we suggest that in real-world settings, the choice of a CL method should be contingent on the specific environmental demands. In cases where the rapid pace of data flow necessitates discarding data, methods like LoRA (parameter expansion), are recommended. Alternatively, when the system can process most of the
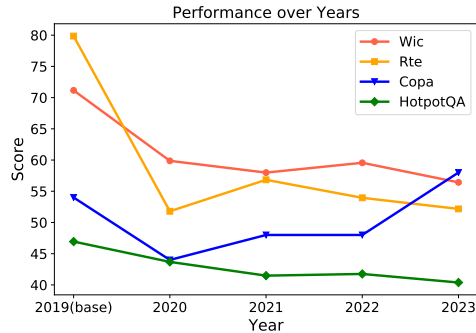
Figure 3: Changes in LM's fundamental capabilities during the OCKL process.

Method	EM	BWT	FWT	KG	KAR
Initial	0.217	–	–	0.278	–
Vanilla	0.192	0.000	-0.014	0.853	-4.996
RecAdam	0.115	0.011	-0.004	0.609	1.191
Mix-Review	0.182	0.003	-0.007	0.802	-1.519
LoRA	0.242	-0.005	-0.002	0.887	-3.180
Kadapter	0.180	-0.032	0.020	0.311	-5.589

Table 7: Failure performance of decoder-only models GPT2 initially pretrained on Dec 2019 dump of Webtext.

data, Mix-Review tends to provide superior outcomes.

4.2.5 Impact of OCKL on Language Model Fundamental Abilities

In this section, we focus on how continuous knowledge acquisition impacts the fundamental capabilities of LMs. To explore this, we have utilized LMs that has been subject to continual pre-training. Subsequently, we fine-tuned and evaluated this model to probe for any potential changes or improvements in its core abilities, including reasoning and discerning the meanings of words. Figure 3 presents the performance of the pre-trained models over different periods. It is observed that there is a general, albeit slight, decline in performance metrics, with the notable exception of the COPA dataset, which shows an initial dip followed by a partial recovery. In contrast, the decline in the HotpotQA dataset is relatively mild, indicating that the OCKL has a minimal impact on the model's reasoning capabilities. However, the RTE dataset exhibits a more significant drop, which may be attributed to the OCKL's more uniform input format, potentially impairing the LM's comprehension abilities.

4.2.6 Exploring How OCKL Methods Transfer Across LM Architectures

Following the approach of (Jang et al., 2022), our experiments involved GPT-2 Large (Radford et al., 2019) of 774M parameters, which is initially pre-trained on WebText (Dec 2019)[†], and later trained on our dynamic knowledge stream for one epoch at a learning rate of 1e-4. The results, presented in Table 7, were not as anticipated, with all strategies underperforming compared to the baseline GPT-2. The underperformance in our GPT-2 Large experiments is mainly due to the rapid nature of online training and the model's decoder-only architecture. Online training restricts GPT-2's exposure to each batch of data, limiting its adaptability to the rapidly changing content typical of streaming data environments. Additionally, the model's focus on generating responses rather than deeply encoding inputs hampers its capacity to process complex, evolving information in our knowledge stream. The brevity of our textual data, rich in content but limited in length, further challenges the model. These insights point towards the need for modifications in both data handling and model architecture to enhance learning efficacy in dynamic online environments.

5 Conclusion

In this paper, we underline the imperative for real-time evaluation metrics in Continual Learning methods, with a particular focus on the continuous pre-training of Language Models. We unveil *Online Continual Knowledge Learning*, a novel framework furnished with a dynamically adaptive benchmark dataset. We propose two innovative metrics: Knowledge Acquisition Rate and Knowledge Gap. Our empirical evaluation reveals that prevailing CL strategies are ill-suited for real-time, stream-based applications. With these insights, we urge the academic community to utilize OCKL as a foundational resource for the future refinement and innovation of adaptive, continually updating LMs.

References

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019. Gradient based sample selection for online continual learning. In *Proceedings of NeurIPS*.

[†]WebText comprises 8 million documents, Wikipedia excluded.

Zalán Borsos, Mojmir Mutny, and Andreas Krause. 2020. Coresets via bilevel optimization for continual learning and streaming.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam Mc-Candlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark experience for general continual learning: a strong, simple baseline.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. 2021. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*.

Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. 2021. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of AAAI*.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, P Dokania, P Torr, and M Ranzato. 2019. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of EMNLP*.

Aristotelis Chrysakis and Marie-Francine Moens. 2020. Online continual learning from imbalanced data. In *Proceedings of ICML*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of EMNLP*.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. In *Proceedings of TACL*.

Yasir Ghunaim, Adel Bibi, Kumail Alhamoud, Motasem Alfarra, Hasan Abed Al Kader Hammoud, Ameya Prabhu, Philip HS Torr, and Bernard Ghanem. 2023. Real-time evaluation in online continual learning: A new hope. In *Proceedings of CVPR*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. In *Proceeding of IJCV*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: adapt language models to domains and tasks. In *Proceedings of ACL*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of ICML*.

Tianxing He, Jun Liu, Kyunghyun Cho, Myle Ott, Bing Liu, James Glass, and Fuchun Peng. 2021. Analyzing the forgetting problem in pretrain-finetuning of open-domain dialogue response models. In *Proceedings of EACL*.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2018. Lifelong learning via progressive distillation and retrospection. In *Proceedings of ECCV*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. 2018. Overcoming catastrophic forgetting for continual learning via model adaptation. In *Proceedings of ICLR*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. In *Proceedings of ICLR*.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong pretraining: Continually

adapting language models to emerging corpora. In *Proceedings of NAACL*.

Tyler B Johnson and Carlos Guestrin. 2018. Training deep models faster with robust, approximate importance sampling. In *Proceedings of NeurIPS*.

Angelos Katharopoulos and François Fleuret. 2018. Not all samples are created equal: Deep learning with importance sampling. In *Proceedings of ICML*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of NeurIPS*.

Yiming Li, Yanyan Shen, and Lei Chen. 2022. Camel: Managing data for efficient stream learning. In *Proceedings of SIGMOD*.

Zhizhong Li and Derek Hoiem. 2018. Learning without forgetting. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. In *Proceedings of ICLR*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of EMNLP*.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Proceedings of NeurIPS*.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2022. Time waits for no one! analysis and challenges of temporal misalignment. In *Proceedings of NAACL*.

Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of CVPR*.

Katerina Margatina, Shuai Wang, Yogarshi Vyas, Neha Anna John, Yassine Benajiba, and Miguel Ballesteros. 2023. Dynamic benchmarking of masked language models on temporal concept drift with multiple views. In *Proceedings of EACL*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *Proceedings of ICLR*.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2022. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of NAACL*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *Proceedings of EMNLP*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. In *Proceedings of JMLR*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. In *Proceedings of TACL*.

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

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of ICLR*.

Samarth Sinha, Jiaming Song, Animesh Garg, and Stefano Ermon. 2022. Experience replay with likelihood-free importance weights. In *Proceedings of Learning for Dynamics and Control Conference*.

Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. 2022. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of CVPR*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2021. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Proceedings of ACL*.

Donggeun Yoo and In So Kweon. 2019. Learning loss for active learning. In *Proceedings of CVPR*.

Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. 2021. Online coreset selection for rehearsal-based continual learning. In *Proceedings of ICLR*.

Michael J.Q. Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of EMNLP*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2021. Modifying memories in transformer models. In *Proceedings of ICLR*.

6 Appendix

6.1 Traditional Continual Learning

Continual learning methodologies have evolved to mitigate catastrophic forgetting, a core challenge in this domain. These strategies can be broadly categorized into three types: *regularization*, *replay-based*, and *parameter expansion* methods.

Regularization techniques, such as those proposed by Buzzega et al. (2020) and Caccia et al. (2021), aim to modify the classification objective to preserve previously learned representations, thus fostering more robust and versatile knowledge retention. Replay-based methods, detailed in works like Chaudhry et al. (2019) and Aljundi et al. (2019), prioritize the selection and retention of representative memory samples, applicable in both offline and online training settings. Parameter expansion strategies, explored by Rusu et al. (2016) and Hu et al. (2018), involve training specific task parameters while keeping others static, allowing for more focused learning.

Studies such as Luu et al. (2022) and Dhingra et al. (2022) have further enriched the field by examining the effects of temporal misalignment in learning. While Luu et al. (2022) delves into the impact of various tasks over extended periods, Dhingra et al. (2022) introduces a method for concurrently modeling text with its associated timestamp, enhancing the memorization of known facts and the calibration of predictions for unseen ones. Extending this approach, Margatina et al. (2023) increased the granularity of evaluations to better understand LMs' handling of evolving factual knowledge. However, these methods predominantly function in an offline capacity, requiring multiple passes over the data—a constraint that limits their practicality in rapidly changing data stream scenarios.

6.2 Coreset Selection

Random Selection A straightforward method for reducing the size of the training data is random selection, which involves randomly choosing a subset of samples from the original dataset. This method's principal advantage is its simplicity and computational efficiency, as it obviates the need for complex selection criteria or domain-specific heuristics. This makes it especially suitable for scenarios involving large datasets where computational resources and time are limited.

Nonetheless, it is essential to acknowledge the limitations of random selection. Specifically, this method may inadvertently exclude informative or representative samples that could be crucial for model training. Moreover, it is ill-suited for applications involving imbalanced datasets since it does not take the distribution of the data into account.

K-Center Selection The K-Center problem, a classical issue in combinatorial optimization and graph theory, has recently been applied innovatively in active learning (Sener and Savarese, 2018; Borsos et al., 2020). In this approach, K centroids are identified to form a coreset, thereby enhancing the quality of the selected data. Extending this methodology, Li et al. (2022) introduced the $\delta - cover$ concept to efficiently prune redundant data in image data streams. This innovation sets an upper bound on the objective function without the need for gradient or loss value computations, effectively reducing training overhead due to redundant or less-informative data instances.

However, transitioning this approach to text data and leveraging embeddings for coreset selection introduces several challenges that we addressed in this work. One such issue is the potential for loss of nuanced details during the embedding process, which can compromise the accuracy of K-center selection. Such an omission can result from the inadequacy of the embedding space in capturing the subtleties inherent in the original data. Another challenge lies in the distortion of inter-point distances within the original dataset when translated into the embedding space, thereby affecting the efficacy of greedy K-center algorithms. This could lead to selections that do not accurately represent the intrinsic structure of the original data.

Model-Based Selection Employing model states, such as loss and gradients, serves as a viable strategy for the selection of representative samples for training. Importance sampling techniques (Johnson and Guestrin, 2018; Katharopoulos and Fleuret, 2018; Sinha et al., 2022) have shown promise in hastening the training process of deep neural models. Specifically, Yoo and Kweon (2019) introduced small trainable modules within the model architecture to predict the loss values for untrained samples. These modules enable the subsequent utilization of samples with low predicted loss values for

further training. Complementing this, Yoon et al. (2021) employed gradient analysis to gauge sample similarity, diversity, and affinity, allowing for the informed selection of data samples in each training iteration.

While these approaches potentially reduce the dataset size and expedite training, it is imperative to consider the computational costs associated with such techniques. Tasks like the training of auxiliary modules and the computation of data-affinity coefficients add an additional computational layer, which must be weighed against the benefits of the accelerated training process.

6.3 Data Construction Process

A visual representation of our data construction process is shown in Figure 4. Leveraging SLING, we utilize the extensive knowledge base of Wikidata to construct these streams. The construction process begins by mapping the Wikidata dump from 2019 to 2023 into multiple data frames. After being Filtered by the wikidata property and using the relation template, these structured frames are transformed into descriptive facts. These narratives are our *Knowledge Stream* containing corpus and dates. For the *QA Stream*, we using to a method that parallels previous works (Margatina et al., 2023; Dhingra et al., 2022). We convert factual narratives into question-answer formats thus structuring the *QA Stream* with questions, answers, and dates for real-time evaluation.

6.4 Data Stream Analysis

As shown in Figure 5, 61.9% of time-varying data has token changes of no less than 7, which indicates that most of the data have significant changes over time.

6.5 CL Setups

Vanilla refers to a standard setup utilized for additional training, as discussed by Gururangan et al. (2020). In this context, the domain represents new knowledge. The Language Model (LM) undergoes further pretraining without the adoption of specialized training techniques, instead leveraging the original training strategy for incremental learning.

RecAdam (Chen et al., 2020) falls into the category of regularization methods. It differs from traditional regularization methods like EWC (Kirkpatrick et al., 2017) by imposing a more stringent independence assumption among model pa-
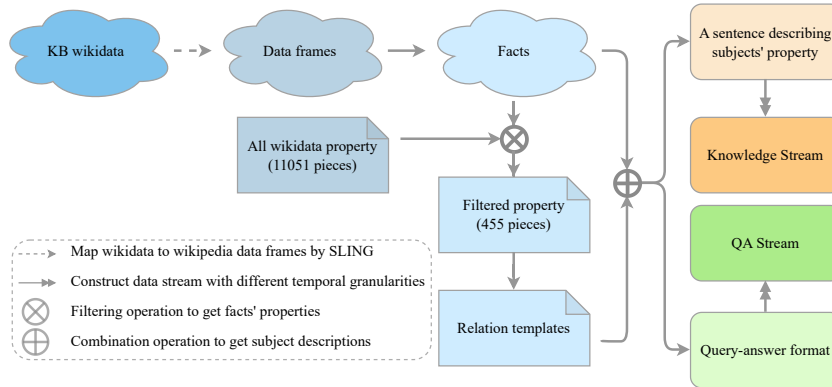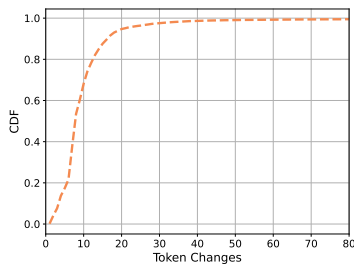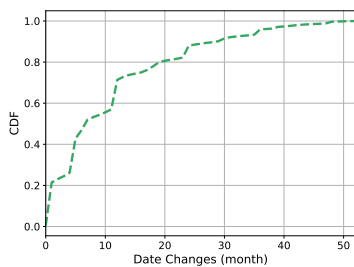
Figure 4: The process of constructing knowledge stream and QA stream for the training and evaluation.

(a) The CDF of the token change

(b) The CDF of the date change

Figure 5: The Cumulative Distribution Function (CDF) of the token and date changes indicates the knowledge stream varies over time.

rameters. Notably, it doesn't use the initial pretraining corpus for regularization during continued training. The regularization strength diminishes as training advances, achieved by annealing the optimizer.

Knowledge Distillation: Gou et al. (2021) introduced a widely-used method to speed up model inference by minimizing the representation gap between two models. Building on this concept, studies by Li and Hoiem (2018); Rebuffi et al. (2017); Hou et al. (2018) have implemented distillation techniques in continuous learning. They do this by designating the model from an earlier task or domain as the teacher, and the model currently being trained as the student. This strategy effectively combats catastrophic forgetting. Recently, Jin et al. (2022) expanded the use of knowledge distillation to the pre-training of language models, with encouraging outcomes. We extend this approach to the OCKL with the subsequent abbreviation T5-KD.

Mix-Review (He et al., 2021) falls into the category of rehearsal methods, which assumes access to the initial pretraining corpus and mixes in random subsets of the initial pretraining data during continued pretraining, depending on the mix-ratio at the current time step. As the training process advances, the mix ratio gradually decreases toward 0, resulting in a reduced presence of the original mixed data with each iteration.

LoRA (Hu et al., 2022) is a parameter-expansion technique presented by Hu et al. (2022). Instead of updating the original LM parameters, it introduces trainable rank-decomposition matrices in each layer for continued pretraining. While Hu et al. (2022) applied this to decoder-only models like GPT-2 (Radford et al., 2019), Jang et al. (2022) application extends to encoder-decoder models.

K-Adapter (Wang et al., 2021) is another parameter expansion method that freezes the original LM parameters while adding a k number of new layers, namely *adapters*, that are updated during continued pre-training. Wang et al. (2021) have shown successful injection of *factual* and *linguistic* knowledge for encoder-only models, BERT (Devlin et al., 2019) & RoBERTa (Liu et al., 2019), while we also apply it to an encoder-decoder model, T5, and decoder-only model, GPT-2.

Modular (Jang et al., 2022) is a novel pa-

Hyper-parameters	Value
Input length	18
Output length	18
epoch	1
Training batch size	8
Testing batch size	512
Model	T5-base
Optimizer	Adafactor
Weight decay	5e-04
# of GPUs	1
GPU	A40 48GB
Learning rate	2e-03

Table 8: Hyper-parameters for T5 experiments

rameter expansion strategy tailored for encoder-decoder models. The original pretrained encoder is preserved, but a supplementary, randomly initialized encoder is introduced for updates during the next phase of training.

6.6 Hyperparameter and Device Settings

Table 8 is our hyperparameter settings and GPU device.

6.7 Evaluation Metrics

Exact Match (EM) The "Exact Match" (EM) metric is employed to assess the fidelity of the model's generated answers in the Question-Answering (QA) context. An answer is designated as an "exact match" if it perfectly aligns with the ground-truth answer in terms of wording, sequence, and punctuation. The mathematical formulation for this metric is as follows:

$$EM = \frac{\text{Number of exact matches}}{\text{Total number of examples}} \quad (2)$$

Backward Transfer (BWT) This metric quantifies the performance degradation on prior tasks after a model has been trained on new tasks, a phenomenon known as "forgetting" in the context of continual learning. Traditional BWT approaches (Lopez-Paz and Ranzato, 2017) compute the mean performance decrement across all preceding tasks after the acquisition of new tasks, offering a generalized measure of the model's retention abilities. To delve deeper into the problem of forgetting, particularly with respect to tasks that are temporally adjacent, we introduce an alternate formulation for BWT as follows:

$$BWT = \frac{1}{T-1} \sum_{i=2}^{T} (R_{i,(i-1)} - R_{(i-1),(i-1)}), \quad (3)$$

where $R_{i,(i-1)}$ denotes the test accuracy of the model trained on tasks $[1, i]$ when evaluated on the preceding $i - 1$ tasks. This adapted metric allows us to focus on performance shifts related to forgetting in consecutive tasks. Given that our data stream exhibits temporal continuity and task correlation—e.g., tasks may share hierarchical or categorical relationships—this revised metric furnishes a more nuanced understanding of how the model's performance evolves, especially concerning the forgetting issue, as it transitions from one task to another.

Forward Transfer (FWT) This metric evaluates the positive impact on the performance of new tasks when a model has completed training on preceding tasks. Traditional FWT approaches (Lopez-Paz and Ranzato, 2017) assess the performance differential on a new task between models that have been trained on all prior tasks and a baseline model that has not undergone any training. While this method offers an aggregate view of how prior learning influences new tasks, it lacks granularity in task-specific contexts. To address this limitation, we introduce an alternative formula for FWT as follows:

$$FWT = \frac{1}{T-1} \sum_{i=2}^{T} (R_{i,i} - R_{i-1,i}), \quad (4)$$

where $R_{i,i}$ represents the test accuracy of the model trained on tasks $[1, i]$ when evaluated on the new i-th task. This refined metric enables us to focus on the performance increments related to task-specific updates in consecutive tasks. Given the temporal continuity and task correlations in our data stream, the new metric serves to better scrutinize how the acquisition of new knowledge influences or disrupts model learning at each transitional phase. This approach thus provides a more nuanced, fine-grained evaluation of the model's ability to integrate new knowledge without adverse effects. To further enhance the depth of our analysis, we also employ vector representations to quantify the updating issue, offering a more comprehensive evaluation of knowledge assimilation.