IN-CONTEXT LEARNING CAN PERFORM CONTINUAL LEARNING LIKE HUMANS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) can adapt to new tasks via in-context learning (ICL) without parameter updates, making them powerful learning engines for fast adaptation. While extensive research has examined ICL as a few-shot learner, whether it can achieve long-term retention and cross-task knowledge accumulation when multitasks arrive sequentially remains underexplored.h Motivated by human memory studies, we investigate the retention characteristics of ICL in multitask settings and extend it to in-context continual learning (ICCL), where continual learning ability emerges through task scheduling and prompt rearrangement. Experiments on Markov-Chain and regbench based benchmarks demonstrate that, for specific large-language models, ICCL benefits from distributed practice (DP) in a manner analogous to humans, consistently revealing a spacing "sweet spot" for retention. Beyond retention performance, we propose a human-retention similarity metric to quantify how closely a continual-learning (CL) method aligns with human retention dynamics. Using this metric, we show that linear-attention models such as MAMBA and RWKV exhibit particularly human-like retention patterns, despite their retention performance lagging behind that of Transformer-based LLMs. Overall, our results establish ICCL as cognitively plausible and practically effective, providing an inference-only CL paradigm that enables LLMs to maintain long-term retention of a target task under cross-task interference without parameter updates.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of domains, with in-context learning (ICL) emerging as one of their most distinctive properties (Brown et al., 2020). Via ICL, LLMs adapt new tasks on the fly by conditioning on demonstrations embedded in the prompt without parameter update. This inference-only property turns them into versatile, efficient solvers that cope with zero-shot (Ouyang et al., 2022), few-shot (Brown et al., 2020), and many-shot (Agarwal et al., 2024) regimes, spanning Bayesian inference (Xie et al., 2024), regression (Raventós et al., 2023; Wu et al., 2023), and reinforcement learning (Laskin et al., 2023; Lee et al., 2023; Wang et al., 2025a). Such flexibility suggests that ICL could evolve into a general-purpose learning engine rather than remain a mere few-shot adapter (Wang et al., 2025b; Fan & Liu, 2025). However, prior work predominantly focuses on single-task or one-off adaptation behavior. Only limited studies investigate multi-task ICL settings, and these works typically optimize short-term prediction by mixing or reordering demonstrations rather than examining whether knowledge can persist across tasks. Thus, it remains unclear whether ICL can support continual learning, where knowledge must be maintained and reused across sequentially presented tasks.

In the broader literature, catastrophic forgetting (CF) has long been recognized as a fundamental challenge in gradient-based continual learning (GBCL). To mitigate CF, a rich body of GBCL has been developed, including regularization-based approaches that reshape the objective function (Li et al., 2020; Evilevitch & Ingram, 2021), replay-based approaches that reorganize the data stream (Rolnick et al., 2019; Buzzega et al., 2021), and approaches that directly modify the optimization process (Arous et al., 2021; Poggio et al., 2011). Although effective on many benchmarks, GBCL approaches fundamentally depend on parameter modification or explicit memory buffers and often struggle to achieve a desirable balance between stability and plasticity. This naturally raises a broader question: *does a similar form of forgetting also arise in parameter-free, inference-only regime of LLMs?*

Table 1: Property comparisons among GBCL, ICL and ICCL.

Property	**GBCL**	**ICL**	**ICCL**
Parameter updates	Yes	No	No
Update rules	Gradient Descent	Blackbox Computation	Blackbox Computation
Knowledge carrier	Parameters	Memory/Hidden States	Memory/Hidden States
Task settings	Multiple Tasks	Single Task	Multiple Tasks

In other words, when tasks are presented sequentially within a long context, can a LLM stably retain the mapping of a target task while processing intervening tasks? To investigate this question, we introduce **in-context continual learning (ICCL)** as a framework for studying inference-time retention in LLMs. Unlike multi-task ICL, which jointly learns all tasks, ICCL focuses on how the ordering and distributed scheduling of target-task and intervening-task demonstrations affect the model's ability to preserve the target-task behavior throughout extended inference, without any parameter updates.

Several recent studies have begun to explore directions related to ICCL, often by extending ICL with prompt modification (e.g., prompt tuning) or external augmentation (e.g., memory retrieval) (Wang et al., 2022; Gao et al., 2023; Momeni et al., 2024; Shinwari & Usama, 2025). While these approaches show promise, they also present several limitations. First, most efforts remain confined to single-task adaptation, without systematically addressing multitask sequential adaptation and cross-task knowledge accumulation. Second, they often rely on heuristic prompt engineering or auxiliary components that raise complexity yet remain vulnerable to memory bloat and retrieval errors, and plateau when procedural memory integration rather than fragmented retrieval is required. Third, current evaluation benchmarks are often either too simple, resulting in trivial zero-shot solutions, or too complex, exceeding the capacity of LLMs to learn within context. As a result, they fail to adequately support systematic analyses of multitask learning dynamics.

In this paper we focus on ICCL, which exploits the LLM's built-in memory to flexibly control both learning and retention rather than external plug-and-play modules and manually specified learning rules, as summarized in Table 1. To compare GBCL and ICCL and to expose ICCL's characteristics, we empirically examine the following perspectives: First, in his pioneering studies of human memory, Hermann Ebbinghaus deliberately used "nonsense syllables" in serial-learning tasks (Ebbinghaus, 2013) to strip away prior knowledge and semantic associations, thereby isolating the core mechanisms of memory. Guided by the same principle, we design two benchmarks for evaluating LLMs whose parameters already encode vast human knowledge: randomly generated Markov chains and regular language generated by random finite automata. These chains preserve essential dynamical-system properties (Attal, 2010; Froyland, 2001) while minimizing contamination from pre-existing knowledge. Second, cognitive science shows that structured practice schedules enhance long-term memory of human (Cepeda et al., 2006; 2008; Pavlik Jr & Anderson, 2005; Bothell, 2020), underscoring the need to introduce task scheduling in machine continual learning. Guided by this, we inject explicit task identifiers and cognition-inspired schedules to systematically probe how scheduling and prompts shape the retention characteristics of both LLMs and GBCL. Those includes single practice (SP), multiple practice (MP), and distributed practice (DP). Experiments show that ICCL retains information more robustly than GBCL baselines and consistently privileges DP over SP and MP, mirroring cognitive findings on human memory. Moreover, linear-attention models such as Mamba and RWKV exhibit retention dynamics that more closely parallel human behavior, even though their absolute retention lags behind standard self-attention. Our contributions are threefold:

- ICCL provides, to our knowledge, the first framework that systematically investigates whether forgetting arises purely at inference time for LLMs and how to enable them to maintain long-term retention of a target task under cross-task interference without parameter updates.
- We show that ICCL benefits from DP in a manner analogous to humans, consistently exhibiting a spacing sweet spot that enhances retention and provides a more favorable balance between stability and plasticity compared to GBCL baselines.

[†]Corresponding to:

- We introduce evaluation benchmarks that combine retention performance with human-retention similarity, enabling systematic analysis of multitask retention and revealing that linear-attention models (e.g., MAMBA, RWKV) display particularly human-like retention patterns, despite lower absolute performance than Transformer-based LLMs.

2 METHODOLOGY

2.1 FORMALIZATION OF ICCL

We now formalize the general protocol of ICCL. A task is denoted by $\tau : y_i^\tau \sim p_\tau(\cdot|x_i)$, which specifies a conditional distribution mapping an input x_i to an output y_i^τ. A segment of φ experiences on task τ is represented as $\mathcal{D}_\varphi^\tau = \{x_1^\tau, y_1^\tau, \ldots, x_\varphi^\tau, y_\varphi^\tau\}$. To model ICCL, we consider a historical sequence of such experience segments $\mathcal{C}_t = \bigoplus_{i=1}^N \mathcal{D}_{\varphi_i}^{\tau_i}$, where $t = \sum_{i=1}^N \varphi_i$ and $\oplus$ denotes ordered concatenation. Unlike a random collection, the order of segments is essential since ICCL is sensitive to the task order arrangement. Given a query input x and the historical sequence $\mathcal{C}_t$, ICCL aims to approximate the output of an unknown target task τ^*. Formally, we define:

$$\hat{p}_\theta(y|x, \mathcal{C}_t) = \mathcal{F}_\theta\left(\bigoplus_{i=1}^N \mathcal{D}_{\varphi_i}^{\tau_i},\ x\right) \ \Rightarrow\ p_{\tau^*}(y|x), \tag{1}$$

where $\mathcal{F}_\theta$ denotes a sequential prediction structure parameterized by θ, such as a large language model. Here, $p(y|x, \mathcal{C}_t)$ represents the probability of predicting output y given x and $\mathcal{C}_t$, and the goal is to approximate the true conditional distribution $p_{\tau^*}(y|x)$. Intuitively, ICCL leverages prior task experiences in context to infer the correct behavior on the target task. Finally, tasks τ_i and τ^* are elements of a task collection $\mathcal{T}$, which defines the general scope of tasks considered in ICCL.

Induction of task identifier. Previous zero-shot instruction-following studies typically assume the availability of a task description d^τ that supplements the prompt (Tanaka et al., 2024; Lou & Yin, 2023). However, in practice, such meta-information is rarely known a priori and tasks must often be inferred implicitly. To accommodate this more realistic setting, we relax d^τ into a *task identifier*, which need not encode semantic details but merely serves to distinguish one task from another. This abstraction aligns well with real-world applications: for instance, in preference modeling, the system may not have access to a user's full profile but can instead rely on a user ID combined with the interaction history. Under this formulation, ICCL can be reformulated as

$$\hat{p}_\theta(y|x, \mathcal{C}_t) = \mathcal{F}_\theta\left(\bigoplus_{i=1}^N \left(d^{\tau_i}, \mathcal{D}_{\varphi_i}^{\tau_i}\right),\ d^{\tau^*}, x\right), \tag{2}$$

where task identifiers d^{τ_i} and d^{τ^*} act as lightweight labels that allow the model to keep track of task boundaries without requiring explicit semantic descriptions.

Evaluation of retention. Retention in ICCL refers to the model's ability to preserve and use earlier demonstrations within long contexts. Given a historical sequence $\mathcal{C}_t$, we next evaluate how much information about the target task τ^* is retained after the model processes an intervening block $\mathcal{D}_{\varphi_D}^{\tau'}$ from another task $\tau' \neq \tau^*$. Specifically, this evaluation quantifies the extent to which the model "remembers" the target task τ^* after the t-step sequence $\mathcal{C}_t$ and an additional φ_D-step exposure to the distracting task τ' :

$$R_{\tau^*,\tau',\mathcal{C}_t,\theta}(t + \varphi_D) = \mathbb{E}_{x,\tau'}\left[M\left(\hat{p}_\theta(\cdot|x, \mathcal{C}_t \oplus D_{\varphi_D}^{\tau'})),\ p_{\tau^*}(\cdot|x)\right)\right], \tag{3}$$

where $M(p_1, p_2)$ is a metric that quantifies the similarity between two probability distributions p_1 and p_2. We use *Normalized Performance* (denoted by the symbol R) to assess the accuracy of $\hat{p}_\theta$, with $R(t) = 0$ indicating a complete loss of memory and $R(t) = 1$ indicating a perfect match between the estimator and the ground truth p_{τ^*}.

Scheduling strategy. We formalize a scheduling strategy to capture how task demonstrations are arranged in a historical sequence $\mathcal{C}_t$. Specifically, different practice conditions correspond to different constructions of $\mathcal{C}_t$, as summarized as follows:

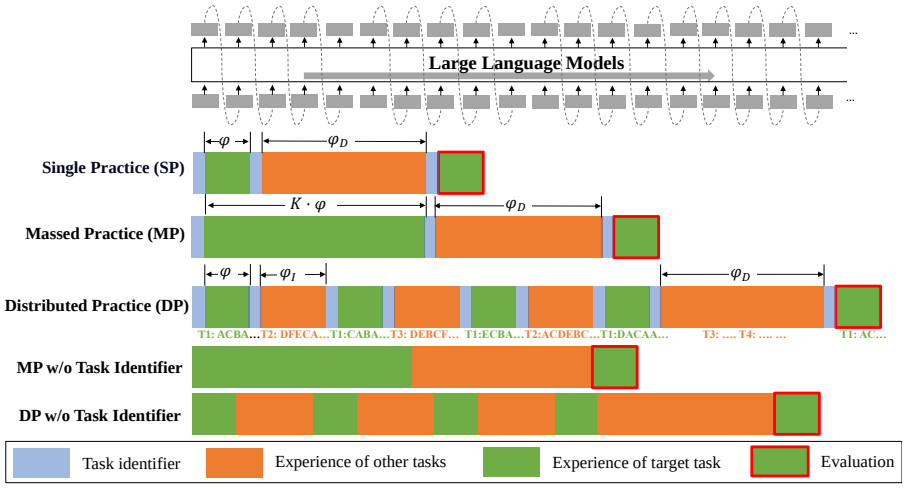


Figure 1: Illustration of how target and other task demonstrations are arranged under SP, MP, and DP, together with their variants without explicit task identifiers in a historical sequence. During the interleaved period, MP contains only target-task demonstrations (with $\varphi_I = 0$) while DP includes both target-task and intervening-task demonstrations ($\varphi_I > 0$). The post-exposure period (φ_D) always involves multi-task interference and is used to evaluate retention decay.

$$\begin{aligned} \text{Single Practice (SP):} \quad & \mathcal{C}_t = \mathcal{D}_{\varphi}^{\tau^*} \\ \text{Massed Practice (MP):} \quad & \mathcal{C}_t = \mathcal{D}_{K\varphi}^{\tau^*} \\ \text{Distributed Practice (DP):} \quad & \mathcal{C}_t = \bigoplus_{i=1}^{K} \left(\mathcal{D}_{\varphi}^{\tau^*}, \mathcal{D}_{\varphi_I}^{\tau_i}\right), \quad \forall \tau_i \in \mathcal{T},\ \tau_i \neq \tau^*. \end{aligned} \tag{4}$$

Eq. (4) characterizes three canonical scheduling conditions, which are illustrated in Figure 1. In *SP*, demonstrations of the target task τ^* are presented only once in a contiguous block of length φ, after which the sequence moves to other tasks. In *MP*, the demonstrations of the target task are extended to a larger block of length $K\varphi$, corresponding to the repeated rehearsal of τ^* without interruption. In *DP*, the sequence alternates between the target task and other tasks: each target task block of length φ is followed by another task block $\mathcal{D}_{\varphi_I}^{\tau_i}$ with $\tau_i \neq \tau^*$. This setup more closely resembles human retention patterns (spacing is most effective when intervening content is distinct from the target) and allows ICCL to retain the target task knowledge in a manner analogous to human retention. Therefore, we maximize differences between target and other tasks to create clear contextual boundaries and reliable reactivation cues. Compared with independent ICL, which must repeatedly include large blocks of target-task demonstrations to sustain retention, ICCL achieves comparable performance using far fewer demonstrations. Under DP, the target-task pattern is naturally reinstated through spaced exposure, enabling the model to maintain high retention even when only a limited number of target-task demonstrations are available. This advantage becomes essential when retention must persist over long content lengths.

To further investigate the role of explicit task identifiers, we additionally evaluate two variants where d^{τ} is removed from the prompt, denoted as *MP w/o instruction* and *DP w/o instruction*, respectively. These two variants test whether the presence of explicit task identifiers contributes to multitask sequential adaptation beyond the task order arrangement.

2.2 Retention Dynamics in ICCL

A well-documented phenomenon in human retention is the spacing effect (Anderson et al., 2004; Cepeda et al., 2008): MP triggers rapid decay, whereas DP yields superior long-term retention. To account for this effect, the ACT-R cognitive model (Borst & Anderson, 2017; Pavlik Jr & Anderson, 2005; Petrov, 2006) describes the resulting learning and retention curves. Suppose the learner practices (observes) the target task τ^* at the time stamp of $t_1, t_2, \ldots, t_{K\varphi}$ and is exposed to other tasks otherwise; the ACT-R model predicts the probability of retaining the task at any subsequent

time $t = t_{K\varphi} + \varphi_D$ by the following equation:

$$\hat{R}(t) = 1/[1 + \exp(-\frac{w(t) - \gamma}{s})], \quad w(t) = \ln \sum_{i=1}^{K\varphi} [\kappa \cdot (t - t_i)]^{-d} \tag{5}$$

where $w(t)$ represents the memory activation that decays over time when no new practice of the target task occurs, and $\hat{R}_{\tau^*}(t)$ denotes the probability of successfully accomplishing task τ^* at time t given the memory activation $w(t)$. Notice that in our settings, for MP, $t_i = i$, for DP, $t_i = i + \left\lfloor \frac{i}{\varphi} \right\rfloor \varphi_I$. The parameters s, γ, and d are hyper-parameters to be determined. To bridge the gap between the actual time-dependency of human retention and the context length in ICCL, we introduce an additional hyper-parameter κ. Our experimental results in Section 3.4 reveal an interior optimum of $\hat{R}(t)$ under DP, indicating the existence of a spacing sweet spot that maximizes the retention probability. This empirical pattern mirrors decades of findings in human memory research and provides practical guidance for prompt scheduling that avoids both massed and overly spaced practice. This retention arises not from short-term contextual reuse, but from spaced reactivation of the target-task pattern, which counteracts interference and long-context attention decay.

2.3 Quantifying Human Retention Similarity via Mahalanobis Distance

We now ask a related question: *to what extent do LLMs resemble human memory in their retention characteristics?* To enable a direct human–model comparison, we introduce Human Retention Similarity via Mahalanobis Distance (HRS-MD), which evaluates how closely the fitted ACT-R parameters of a model fall within parameter distributions reported in cognitive psychology. Specifically, Let the fitted parameter vector be $\hat{\boldsymbol{\theta}} = [\hat{d},\ \hat{s},\ \hat{\gamma}]^\top$, where $\hat{d}$ denotes the decay rate, $\hat{s}$ the activation noise, and $\hat{\gamma}$ the retrieval threshold estimated from retention curves. Human reference distributions are characterized by a mean vector $\boldsymbol{\mu} = [\mu_d,\ \mu_s,\ \mu_\gamma]^\top$ and covariance matrix $\boldsymbol{\Sigma}$, derived from prior experimental studies (Pavlik Jr & Anderson, 2005; Lewis & Vasishth, 2005; Bothell, 2020; Borst & Anderson, 2017; Said et al., 2016; van der Velde et al., 2022). HRS-MD is then defined as $(\hat{\boldsymbol{\theta}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\mu})$, which quantifies the discrepancy between the model-fitted parameters and human baselines while accounting for parameter scales and correlations. A smaller HRS-MD indicates closer alignment between LLMs and human memory retention. This metric enables systematic comparisons across ICCL and GBCL methods, revealing which model classes most closely reproduce human-like retention pattern. Details of $\boldsymbol{\Sigma}$ and the ACT-R parameters, along with their reference distributions and the rationale for the Mahalanobis metric, are provided in Appendix A.2.

3 Experiment Results

3.1 Experiment Settings

Models and Baselines. We evaluate two categories of methods: inference-only ICCL models and GBCL baselines. The ICCL group consists of four representative open-source LLMs: LLaMA3-8B, DeepSeek-R1, MAMBA, and RWKV-7. Note that ICCL focuses on inference-time retention within standard LLMs and does not rely on architectural or external-memory mechanisms; thus, comparisons with memory-enhanced models fall outside the scope of this study. The GBCL group includes three classical approaches and one modern method: Stochastic Gradient Descent (SGD) (Arous et al., 2021), Experience Replay (ER) (Rolnick et al., 2019), Elastic Weight Consolidation (EWC) (Evilevitch & Ingram, 2021), and Prompt Gradient Projection (PGP) (Qiao et al., 2024). For ICCL, experiences serve as the context in LLMs without updating any parameters; for GBCL, we construct a small predictive model that maps the input & task-identifier to an output and sequentially refine it with the arriving experiences. The detailed model architecture and hyperparameters are provided in the Appendix A.3. To ensure fair comparisons, all models are evaluated on the same input sequences and prompt structures, with greedy decoding applied during inference. Each experiment is repeated 16 times with different random seeds, and results are reported as stable averages.

Benchmark Tasks. Discrete Markov Chains (DMC) are widely adopted benchmarking tasks (Zekri et al., 2024; Akyürek et al., 2024; Liu et al., 2024) because they capture essential properties of dynamical systems while minimizing contamination from pre-existing knowledge, thus providing

a clean and controllable environment for analyzing retention and interference dynamics. In our benchmark, two levels of complexity are considered: *simple* tasks with $N_s = 4$ states and *complex* tasks with $N_s = 8$ states. For each setting, transition matrices are generated to sample states and construct experience sequences. Each target and interference tasks are defined by a transition matrix $P_{target} \in \mathbb{R}^{N_s \times N_s}$ and a transition matrix $P_{interference} \in \mathbb{R}^{N_s \times N_s}$, and the model receives a state sequence as the main prompt content to predict the next state. Prompts are prepared under two conditions: (i) with task identifiers, where each sequence segment is prefixed with a label (e.g., `[TARGET_TASK]`, `[INTERFERENCE_TASK]`), and (ii) without identifiers, where the same content is presented unlabeled. This setting enables systematic evaluation of whether explicit identifiers facilitate multitask sequential adaptation. We apply the scheduling strategies introduced in Section 2.1 to form prompts in SP, MP, and DP formats. We include a detailed example in the Appendix A.4 illustrating how a DMC prompt is constructed, along with its corresponding transition matrices. We further design a complementary benchmark built upon the `RegBench` framework (Akyürek et al., 2024) to evaluate ICCL in natural language–style environments. `RegBench` introduces linguistically structured inputs and richer syntactic variation, which make `RegBench` closer to real-world sequential learning scenarios. Appendix A.4 provides complete details on the prompt construction process and ICCL's retention performance on this new benchmark.

Parameter Settings. We configure the DMC benchmark with $\varphi = 100, K = 5, \varphi_I \in \{10, 50, 100, 200, 400, 600\}, \varphi_D \in (0, 700)$, considering the token limits of different LLMs, e.g., `DEEPSEEK-R1` supports up to approximately 4096 tokens, whereas `LLAMA3-8B` allows up to 8192 tokens. These settings ensure that all models operate within their valid context windows while maintaining comparable information exposure across architectures. They also enable controlled studies on *interference length* (φ_I) and *duration* (φ_D), providing a consistent basis for analyzing retention and interference as the effective context grows. This unified configuration guarantees that each model runs at its optimal performance regime without exceeding its token budget.

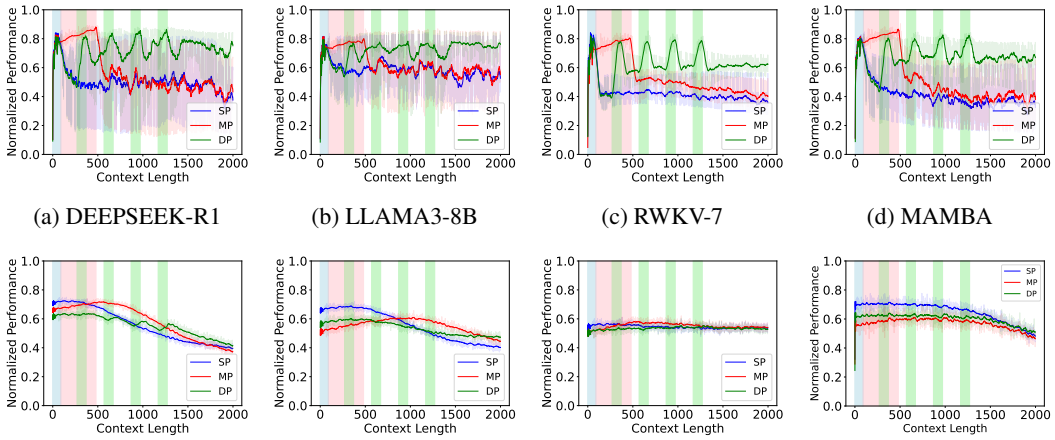
Evaluation Metrics. A bottleneck in DMC evaluation is that performance divergence across tasks often stems from intrinsic task differences rather than prediction accuracy. To suppress this noise, we first introduce a normalized performance metric. Inspired by the normalization schemes in (Cha, 2007), we define the normalized retention performance metric M (from eq. (3)) by rescaling the Bhattacharyya distance (Liu et al., 2021; Zekri et al., 2024) $D_{Bah}(p_1, p_2)$ to the interval $[0, 1]$:

$$M(\hat{p}_\theta, p_{\tau^*}) = \frac{\exp\left[D_{Bah}(p_{\tau^*}, p_{\tau^*}) - D_{Bah}(\hat{p}_\theta, p_{\tau^*})\right] - 1}{\exp\left[D_{Bah}(p_{\tau^*}, p_{\tau^*}) - D_{Bah}(p_{\mathcal{U}}, p_{\tau^*})\right] - 1}, \tag{6}$$

where D_{Bah} is the Bhattacharyya distance and $p_{\mathcal{U}}$ denotes the uniform distribution, representing complete forgetting. This normalization makes retention scores comparable across tasks by rescaling the model–target divergence between the best case $D_{Bah}(p_{\tau^*}, p_{\tau^*})$ and the worst case $D_{Bah}(p_{\mathcal{U}}, p_{\tau^*})$. Bhattacharyya distance is chosen because it directly measures overlap between predictive and target distributions, though its raw scale varies with label entropy and task complexity. The exponential transform further linearizes distance differences and amplifies meaningful gaps when distributions are close, following standard similarity-normalization practices (Cha, 2007). Under this mapping, $M = 1$ corresponds to perfect retention, $M = 0$ to uniform guessing, and intermediate values reflect retention quality on a consistent cross-task scale. All reported results are given as mean $\pm$ 95% CI. We additionally employ HRS-MD to quantify alignment with human memory dynamics.

3.2 Retention Performance Under SP, MP and DP

We first evaluate how the retention performance of ICCL models and GBCL baselines varies on simple and complex tasks with task identifiers and $\varphi_I = 200$, under SP, MP, and DP scheduling. The experimental results on the complex task are shown in Figure 2. For both SP and MP, we observe that ICCL models exhibit a sharp performance drop immediately after the last target-task block, followed by a small continued decline as φ_D increases, whereas some GBCL methods show a more gradual but consistent decrease–underscoring their susceptibility to catastrophic forgetting. Under DP, however, ICCL benefits substantially: DP effectively mitigates forgetting and maintains higher retention, while GBCL baselines show little improvement. We also observe that once LLM transitions from the target-task block to intervening tasks, its retention on the target task drops abruptly, even though the target demonstrations were encountered only moments earlier and remain relatively close



(a) DEEPSEEK-R1 (b) LLAMA3-8B (c) RWKV-7 (d) MAMBA

(e) SGD (f) ER (g) EWC (h) PGP

Figure 2: Retention performance on complex task under SP, MP, and DP settings (with task identifiers, $\varphi_I = 200$). Target task blocks are highlighted in light blue (SP), pink (MP), and green (DP).

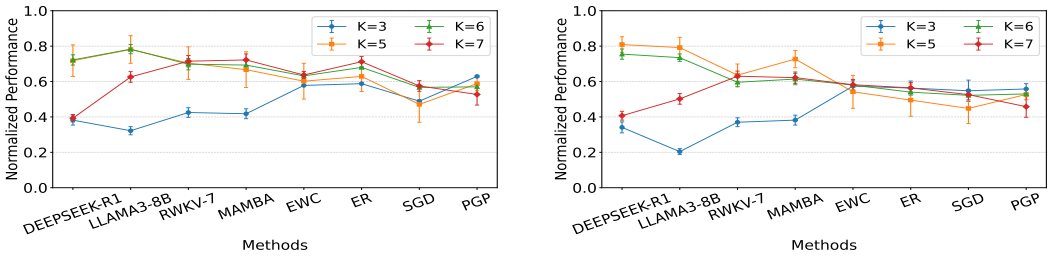
Table 2: Retention performance (mean ± 95% CI) under DP for simple ($N_s = 4$) and complex ($N_s = 8$) Markov-chain tasks across varying intervals φ_D.

Simple Task (markov chain with $N_s = 4$)					
Model Types	$\varphi_D = 0$	$\varphi_D = 100$	$\varphi_D = 200$	$\varphi_D = 400$	$\varphi_D = 600$
LLAMA3-8B	$\mathbf{0.819 \pm 0.086}$	$\mathbf{0.737 \pm 0.090}$	$\mathbf{0.865 \pm 0.030}$	$\mathbf{0.833 \pm 0.067}$	$\mathbf{0.781 \pm 0.078}$
MAMBA	0.668 ± 0.106	0.655 ± 0.105	0.678 ± 0.094	0.685 ± 0.100	0.667 ± 0.101
SGD	0.606 ± 0.100	0.581 ± 0.084	0.558 ± 0.086	0.507 ± 0.098	0.471 ± 0.101
ER	0.608 ± 0.095	0.613 ± 0.075	0.617 ± 0.077	0.620 ± 0.081	0.629 ± 0.085
EWC	0.617 ± 0.101	0.612 ± 0.086	0.604 ± 0.088	0.601 ± 0.099	0.602 ± 0.101
PGP	0.665 ± 0.064	0.652 ± 0.066	0.637 ± 0.059	0.609 ± 0.069	0.562 ± 0.079
Complex Task (markov chain with $N_s = 8$)					
Model Types	$\varphi_D = 0$	$\varphi_D = 100$	$\varphi_D = 200$	$\varphi_D = 400$	$\varphi_D = 600$
LLAMA3-8B	$\mathbf{0.784 \pm 0.054}$	$\mathbf{0.776 \pm 0.056}$	$\mathbf{0.753 \pm 0.086}$	$\mathbf{0.783 \pm 0.057}$	$\mathbf{0.792 \pm 0.058}$
MAMBA	0.750 ± 0.059	0.700 ± 0.061	0.608 ± 0.103	0.724 ± 0.064	0.726 ± 0.048
SGD	0.577 ± 0.072	0.540 ± 0.076	0.491 ± 0.080	0.464 ± 0.073	0.448 ± 0.085
ER	0.519 ± 0.081	0.501 ± 0.086	0.472 ± 0.088	0.479 ± 0.080	0.495 ± 0.091
EWC	0.563 ± 0.079	0.551 ± 0.087	0.526 ± 0.091	0.537 ± 0.083	0.542 ± 0.093
PGP	0.608 ± 0.057	0.600 ± 0.061	0.584 ± 0.060	0.549 ± 0.063	0.515 ± 0.060

to the query in absolute context position. This sharp decline shows that the dominant failure mode is not insufficient recency, but rather the model's susceptibility to interference and long-range attention decay during continued context processing.

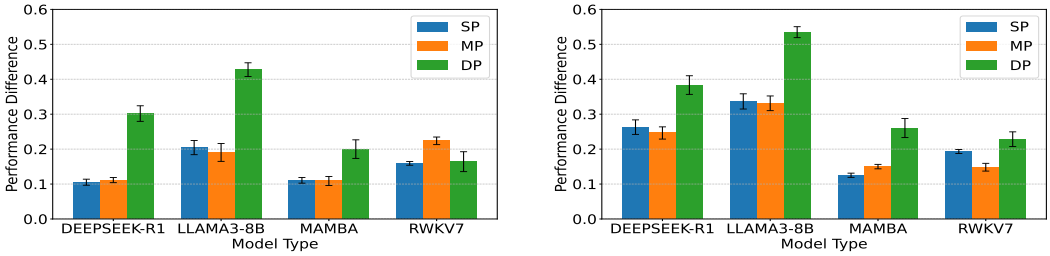
Building on the above results, Table 2 compares the retention performance of ICCL models and GBCL baselines under DP across simple and complex tasks as φ_D increases from 0 to 600. ICCL consistently outperforms all GBCL methods, and although retention declines with larger φ_D, ICCL remains in a high-retention regime while GBCL methods operate at substantially lower levels. Although ER and EWC show relatively small numerical variations across φ_D, this pattern arises from early saturation at a low-retention regime. By contrast, ICCL maintains much higher retention with a slower degradation rate, demonstrating stronger resistance to extended interference.

Ablation Study on Target Task Block Numbers K. To further understand how the number of target-task blocks K influences model retention, we conduct an ablation study where K varies in $\{3, 5, 6, 7\}$ while fixing $\varphi_D = 600$. Figure 3 presents the normalized retention performance of ICCL and GBCL methods under DP. We observe that K has a much stronger impact on ICCL methods than on GBCL methods. For transformer-based LLMs (e.g., `DEEPSEEK-R1` and `LLAMA3-8B`),



(a) Simple Task (markov chain with $N_s = 4$) (b) Complex Task (markov chain with $N_s = 8$)

Figure 3: Retention performance (with $\varphi_D = 600$) under DP for simple ($N_s = 4$) and complex ($N_s = 8$) Markov-chain tasks across different target task block numbers ($K \in \{3, 5, 6, 7\}$).



(a) Simple Task (markov chain with $N_s = 4$) (b) Complex Task (markov chain with $N_s = 8$)

Figure 4: Retention performance difference of ICCL models with and without task identifiers under SP, MP, and DP ($\varphi_I = 200$ and $\varphi_D = 600$). Positive values indicate higher retention when task identifiers are included in the prompt.

retention performance first increases and then decreases as K grows, because a larger K yields longer effective contexts that approach or exceed the transformer's token limit. In contrast, linear-attention LLMs (e.g., `RWKV-7` and `MAMBA`) show continuously improving and eventually stable retention performance, reflecting their superior scalability to long sequences. Overall, the results confirm that context length ($K\varphi$) interacts with model architecture to determine how efficiently previous task information is retained. This trend consistently appears in both simple and complex Markov-chain tasks, highlighting K as a key factor shaping long-term memory dynamics in LLMs.

3.3 Effect of Task Identifier on ICCL Retention

Figure 4 illustrates retention performance difference of ICCL models when prompts include task identifiers versus when they do not, under SP, MP, and DP conditions. Here, the plotted value represents the difference between ICCL's retention performance with and without identifiers; a positive value indicates an improvement when identifiers are used. As shown in Figure 4, incorporating task identifiers consistently enhances retention across all models and scheduling strategies, with the most pronounced gains observed under DP. The effect is more substantial in complex DMC tasks, suggesting that explicit identifiers become increasingly important as task complexity grows.

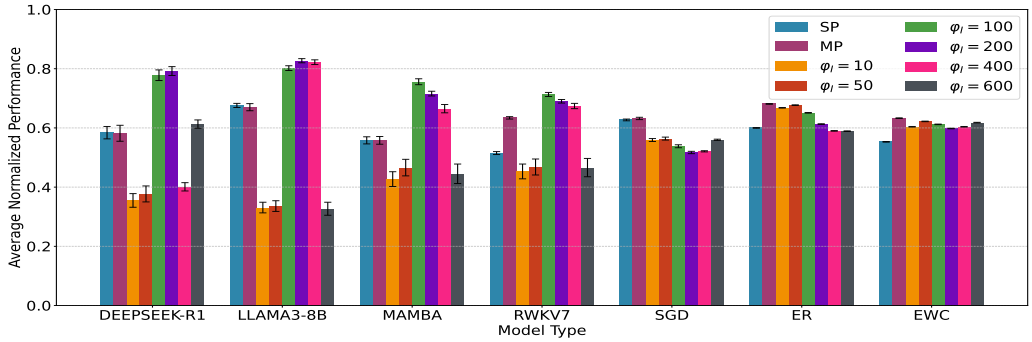
3.4 Spacing Sweet Spot in ICCL under DP

Figure 5 reports the average retention performance of ICCL and GBCL models under DP across different intervals φ_I, evaluated on both simple and complex tasks. Here, average retention performance is defined as the mean retention over φ_D, thereby capturing memory retention beyond the immediate target-task exposure. The results reveal a clear contrast between ICCL and GBCL. For ICCL, DP induces a characteristic *spacing sweet spot*, where average retention performance peaks at an intermediate φ_I lying between 100 and 400. This sweet spot consistently emerges across both simple and complex tasks, although average retention performance is higher in the simpler setting. These findings empirically support the presence of a spacing sweet spot and further demonstrate that

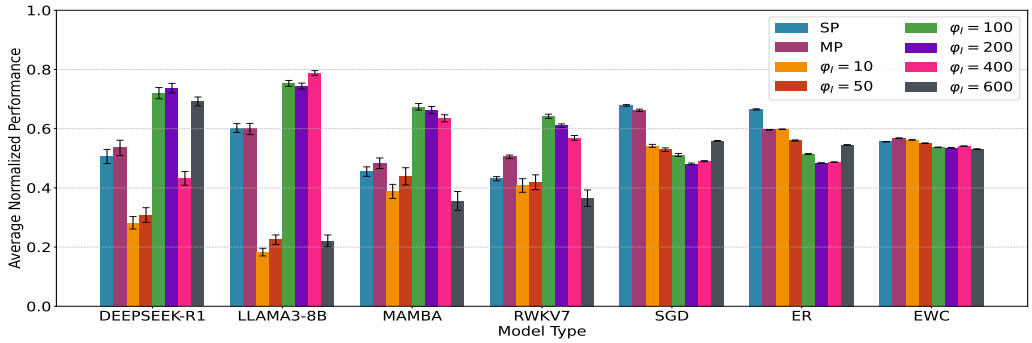
Table 3: Fitted ACT-R parameters with corresponding MSE and HRS-MD values.

Model	Decay Rate (d)	Activation Noise (s)	Scaling Factor (κ)	Threshold (γ)	MSE	HRS-MD
LLAMA3	0.14	2.00	1.20	1.01	0.0024	500.22
DEEPSEEK-R1	0.35	1.69	0.43	-0.24	0.0155	302.39
RWKV-7	0.27	1.62	0.94	0.59	0.0052	**287.57**
MAMBA	0.29	1.59	0.76	0.33	0.0049	**272.74**
SGD	0.41	2.00	0.77	0.15	0.0036	445.07
ER	0.27	2.00	1.36	1.02	0.0061	466.74
EWC	0.22	2.00	1.31	1.65	0.0021	481.53

ICCL under DP surpasses the performance achieved with SP and MP. Thus, ICCL benefits from DP in a manner analogous to human memory, and careful tuning of φ_I can substantially enhance retention. In contrast, GBCL baselines exhibit no such sweet spot: their performance remains largely flat or gradually declines as φ_I increases, indicating that DP provides little to no advantage.



(a) Simple Task (markov chain with $N_s = 4$)



(b) Complex Task (markov chain with $N_s = 8$)

Figure 5: Average retention performance comparison among ICCL and GBCL methods under DP with different intervals (φ_I) on both simple and complex tasks.

3.5 HRS-MD Quantification

To investigate how the aforementioned retention curves align with human retention, we employed the human-retention-validated ACT-R model to fit the ICCL and GBCL curves. Specifically, we used eq. (5) to model the normalized ICCL/GBCL performance obtained from eq. (3). Table 3 summarizes the fitted ACT-R parameters, mean squared error (MSE) during parameter fitting, and HRS-MD values for both ICCL and GBCL methods based on retention performance results on both simple and complex tasks under DP across different intervals φ_I. Several consistent patterns emerge. First, ICCL models–except for LLaMA3–achieve lower HRS-MD values than GBCL baselines, indicating

that their retention dynamics more closely approximate human memory. In particular, MAMBA and RWKV-7 yield the lowest HRS-MD scores (272.74 and 287.57, respectively), demonstrating the strongest alignment with human retention patterns. By contrast, GBCL methods produce substantially higher HRS-MD values, reflecting greater divergence from human memory dynamics. Second, the fitted parameters reveal distinctive tendencies across the two paradigms. ICCL models generally converge to moderate decay rates, relatively low activation noise, and thresholds near zero—indicating efficient and accessible retrieval dynamics, particularly evident in MAMBA and RWKV-7. In contrast, GBCL methods exhibit more extreme parameter configurations: SGD yields the highest decay rate ($d = 0.41$), consistent with rapid forgetting, while ER and EWC converge to elevated thresholds ($\gamma \geq 1.0$), implying stricter retrieval conditions and reduced accessibility of stored knowledge. These patterns highlight the poor balance between stability and plasticity in GBCL approaches.

4 Related Work

GBCL is the classical approach for mitigating catastrophic forgetting through parameter updates. Methods typically fall into three categories: EWC, ER, and SGD. EWC constrains updates to parameters important for previous tasks, using Fisher-information–based scores (Li et al., 2020; Evilevitch & Ingram, 2021) or online importance estimates such as synaptic intelligence (Zenke et al., 2017; Zenke et al.). ER maintains a small memory buffer and interleaves past examples with new data (Rolnick et al., 2019; Buzzega et al., 2021). SGD (Arous et al., 2021; Poggio et al., 2011) enables rapid adaptation but suffers severe forgetting due to its lack of preservation mechanisms. Recent extensions emphasize representation stability, but remain incompatible with our setting: EC-ACE (Xiong et al., 2024) relies on vision-specific multi-view consistency losses, while InsCL (Wang et al., 2024) adapts LLMs via instruction-tuned parameter updates—both outside the scope of inference-only ICCL.

ICL allows LLMs to adapt to tasks from in-prompt demonstrations without parameter updates. Prior work has analyzed its empirical sensitivities (Zhao et al., 2021; Liu et al., 2021; Lu et al., 2021; Liu et al., 2023) and offered theoretical views of ICL as Bayesian inference, gradient-based adaptation, or kernel regression (Xie et al., 2021; Akyürek et al., 2022; Von Oswald et al., 2023; Han et al., 2023). Many-shot studies further show extension to more complex tasks (Agarwal et al., 2024; Garg et al., 2022). Yet most research focuses on single-task adaptation, leaving sequential multitask retention largely unexplored. Recent continual-ICL efforts adjust prompts incrementally (Wang et al., 2022; Xiong et al., 2024) or use retrieval modules (Gao et al., 2023; Momeni et al., 2024; Shinwari & Usama, 2025), but these rely on heuristics or auxiliary components prone to memory growth and retrieval noise. While (Xiong et al., 2024) mixes demonstrations across tasks for short-term gains, ICCL instead examines long-term retention under sequential exposures.

Human memory research offers a rich foundation for understanding retention dynamics. Ebbinghaus (Ebbinghaus, 2013) introduced the forgetting curve, later refined into power-law formulations (White, 2001). The *spacing effect* shows that distributed practice yields superior long-term retention, a finding supported by large-scale meta-analyses (Cepeda et al., 2006; 2008). These phenomena are formalized in the base-level learning equation (Pavlik Jr & Anderson, 2005) and its ACT-R extension (Pavlik & Anderson, 2008; Bothell, 2020; Borst & Anderson, 2017), which link recall to the frequency and recency of exposures. Our work leverages these principles to test whether LLMs display analogous retention patterns when demonstrations are distributed across prompts.

5 Conclusion

In this work, we investigated the retention characteristics of ICL in multi-task settings and extended it to ICCL through prompt scheduling and rearrangement. Experiments on benchmarks showed that ICCL benefits DP in a manner analogous to humans, consistently revealing a spacing sweet spot that enhances retention beyond SP and MP settings. Beyond retention performance, we introduced the HRS-MD metric to quantify alignment with human memory dynamics, and found that linear-attention models such as MAMBA and RWKV display particularly human-like retention patterns. Overall, our results establish ICCL as both cognitively plausible and practically effective, providing an inference-only paradigm that mitigates catastrophic forgetting and addresses the stability–plasticity dilemma in conventional continual learning methods.

Ethics Statement

This work adheres to the ICLR Code of Ethics. Our study focuses on the development and evaluation of in-context continual learning methods using synthetic and publicly available benchmark datasets. No human subjects or sensitive personal data are involved. The proposed methods aim to improve retention dynamics in large language models without introducing harmful applications or privacy risks. All datasets used are either synthetic (Markov-chain sequences) or publicly accessible, and no proprietary or restricted data are employed. In addition, the large language models used in our experiments–LLaMA3-8B, DeepSeek-R1, MAMBA, and RWKV-7–are employed strictly in compliance with their open-source licenses. We are not aware of any conflicts of interest, sponsorship influence, or ethical risks associated with this research.

Reproducibility Statement

We have taken multiple steps to ensure the reproducibility of our results. The main text provides a complete description of the proposed ICCL framework, the baseline methods, and the evaluation metric. Details of additional ablation studies are included in the Appendix. All datasets used in the experiments are synthetically generated Markov chain sequences or are publicly available. Hyperparameter choices and training configurations for both GBCL baselines and large language model baselines are fully documented. To further support replication, we will release our anonymized source code as supplemental materials submitted with the paper.

References

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Ekin Akyürek, Bailin Wang, Yoon Kim, and Jacob Andreas. In-context language learning: Architectures and algorithms. In *International Conference on Machine Learning*, pp. 787–812. PMLR, 2024.

John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological review*, 111(4):1036, 2004.

Gerard Ben Arous, Reza Gheissari, and Aukosh Jagannath. Online stochastic gradient descent on non-convex losses from high-dimensional inference. *Journal of Machine Learning Research*, 22 (106):1–51, 2021.

Stéphane Attal. Markov chains and dynamical systems: the open system point of view. *arXiv preprint arXiv:1010.2894*, 2010.

Jelmer P. Borst and John R. Anderson. A step-by-step tutorial on using the cognitive architecture act-r in combination with fmri data. *Journal of Mathematical Psychology*, 76:94–103, 2017. URL `https://www.jelmerborst.nl/pubs/Borst2017.pdf`.

Dan Bothell. Act-r 7.21+ reference manual. *Pittsburgh (PA): Carnegie Mellon University*, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking experience replay: a bag of tricks for continual learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2180–2187. IEEE, 2021.

Nicholas J Cepeda, Harold Pashler, Edward Vul, John T Wixted, and Doug Rohrer. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological bulletin*, 132(3): 354, 2006.

Nicholas J Cepeda, Edward Vul, Doug Rohrer, John T Wixted, and Harold Pashler. Spacing effects in learning: A temporal ridgeline of optimal retention. *Psychological science*, 19(11):1095–1102, 2008.

Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

DeepSeek. Deepseek coder 7b instruct v1.5. `https://huggingface.co/deepseek-ai/deepseek-coder-7b-instruct-v1.5`, 2024.

Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155, 2013.

Anton Evilevitch and Robert Ingram. Avoiding catastrophic forgetting in continual learning through elastic weight consolidation, 2021.

Wang Fan and Shaoshan Liu. Putting the smarts into robot bodies. *Commun. ACM*, 68(3):6–8, 2025. doi:10.1145/3703761. URL `https://doi.org/10.1145/3703761`.

Gary Froyland. Extracting dynamical behavior via markov models. In *Nonlinear dynamics and statistics*, pp. 281–321. Springer, 2001.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

Chi Han, Ziqi Wang, Han Zhao, and Heng Ji. Explaining emergent in-context learning as kernel regression. *arXiv preprint arXiv:2305.12766*, 2023.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, et al. In-context reinforcement learning with algorithm distillation. In *The Eleventh International Conference on Learning Representations*, 2023.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:43057–43083, 2023.

Richard L Lewis and Shravan Vasishth. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3):375–419, 2005.

Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780*, 2020.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

Toni Jb Liu, Nicolas Boulle, Raphaël Sarfati, and Christopher Earls. Llms learn governing principles of dynamical systems, revealing an in-context neural scaling law. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15097–15117, 2024.

Renze Lou and Wenpeng Yin. Toward zero-shot instruction following. *arXiv preprint arXiv:2308.03795*, 2023.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

Meta. Llama 3.1 8b instruct. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct, 2024.

Saleh Momeni, Sahisnu Mazumder, Zixuan Ke, and Bing Liu. In-context continual learning assisted by an external continual learner. *arXiv preprint arXiv:2412.15563*, 2024.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Philip I Pavlik and John R Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101, 2008.

Philip I Pavlik Jr and John R Anderson. Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. *Cognitive science*, 29(4):559–586, 2005.

Alexander A Petrov. Computationally efficient approximation of the base-level learning equation in act-r. In *Proceedings of the seventh international conference on cognitive modeling*, pp. 391–392. Trieste, ITA Edizioni Goliardiche, 2006.

Tomaso Poggio, Stephen Voinea, and Lorenzo Rosasco. Online learning, stability, and stochastic gradient descent. *arXiv preprint arXiv:1105.4701*, 2011.

Jingyang Qiao, Xin Tan, Chengwei Chen, Yanyun Qu, Yong Peng, Yuan Xie, et al. Prompt gradient projection for continual learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Allan Raventós, Mansheej Paul, Feng Chen, and Surya Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. *Advances in neural information processing systems*, 36:14228–14246, 2023.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

RWKV. Rwkv-4 world 7b. https://huggingface.co/RWKV/rwkv-4-world-7b, 2024.

Nadia Said, Till Nagel, and Frank Rebitschek. Applying mathematical optimization methods to an act-r model: The sugar factory. *PLOS ONE*, 11(7):e0158832, 2016. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0158832.

Haseeb Ullah Khan Shinwari and Muhammad Usama. Memory-augmented architecture for long-term context handling in large language models. *arXiv preprint arXiv:2506.18271*, 2025.

Ryota Tanaka, Taichi Iki, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. Instructdoc: A dataset for zero-shot generalization of visual document understanding with instructions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19071–19079, 2024.

TII. Falcon-mamba 7b. https://huggingface.co/tiiuae/falcon-mamba-7b, 2024.

Maarten van der Velde, Florian Sense, Jelmer P. Borst, Leendert van Maanen, and Hedderik van Rijn. Capturing dynamic performance in a cognitive model: Estimating act-r memory parameters with the linear ballistic accumulator. *Topics in Cognitive Science*, 14(4):889–903, 2022. doi:10.1111/tops.12614. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC9790673/.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Fan Wang, Pengtao Shao, Yiming Zhang, Bo Yu, Shaoshan Liu, Ning Ding, Yang Cao, Yu Kang, and Haifeng Wang. Omnirl: In-context reinforcement learning by large-scale meta-training in randomized worlds. *arXiv preprint arXiv:2502.02869*, 2025a.

Fan Wang, Bo Yu, Pengtao Shao, Zhiyuan Chen, Hungchyun Chou, Liuwang Kang, Shenbao Shao, Xuan Xia, Ning Ding, Haoyi Xiong, et al. In-context learning as general-purpose learning: A comprehensive survey and new perspectives. 2025b.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. Inscl: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. *arXiv preprint arXiv:2403.11435*, 2024.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022.

K Geoffrey White. Forgetting functions. *Animal Learning & Behavior*, 29(3):193–207, 2001.

Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2024.

Zheyang Xiong, Ziyang Cai, John Cooper, Albert Ge, Vasilis Papageorgiou, Zack Sifakis, Angeliki Giannou, Ziqian Lin, Liu Yang, Saurabh Agarwal, et al. Everything everywhere all at once: Llms can in-context learn multiple tasks in superposition. *arXiv preprint arXiv:2410.05603*, 2024.

Oussama Zekri, Ambroise Odonnat, Abdelhakim Benechehab, Linus Bleistein, Nicolas Boullé, and Ievgen Redko. Large language models as markov chains. *arXiv preprint arXiv:2410.02724*, 2024.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Improved multitask learning through synaptic intelligence. In *Proceedings of the International Conference on Machine Learning*, pp. 3987–3995.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. PMLR, 2017.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. PMLR, 2021.

A Appendix

A.1 Use of LLMs

We used large language models (LLMs) only as an auxiliary tool to improve the clarity and presentation of this paper. The assistance was limited to:

- **Language refinement:** grammar checking, wording suggestions, and improving sentence fluency while preserving the authors' original technical content.
- **Mathematical support:** verifying the correctness and readability of some derivations and notations, without introducing new technical results.

No LLM was used for generating research ideas, designing experiments, analyzing results, or writing original scientific content. All conceptual and technical contributions were made by the authors. The authors take full responsibility for the entire content of this paper, and LLMs are not eligible for authorship.

Table 4: Statistical reference means and standard derivations for human's ACT-R model parameters according to (Pavlik Jr & Anderson, 2005; Lewis & Vasishth, 2005; Bothell, 2020; Borst & Anderson, 2017; Said et al., 2016; van der Velde et al., 2022).

Parameter	Mean value (μ)	Standard derivation (σ)
Decay Rate (d)	$\mu_d = 0.50$	$\sigma_d = 0.05$
Activation Noise (s)	$\mu_s = 0.32$	$\sigma_s = 0.08$
Threshold (γ)	$\mu_\gamma = -0.50$	$\sigma_\gamma = 0.71$

A.2 ACT-R Model and HRS-MD Calculation

To complement Section 2.3, we provide the full ACT-R retention model together with an intuitive explanation of its parameters and their roles in memory dynamics.

ACT-R base-level activation and retrieval probability. In ACT-R model (Pavlik Jr & Anderson, 2005; Lewis & Vasishth, 2005; Bothell, 2020), each practice creates a memory trace. If the learner is exposed to the practice at times $t_1, t_2, \ldots, t_k$, *memory strength* or *base-level activation* at a later time t is:

$$w(t) \;=\; \ln \sum_{i=1}^{k} [\Delta t_i]^{-d}\,, \tag{7}$$

where d controls how quickly each memory trace decays (larger d implies faster forgetting) and Δt_i is the time elapsed since the i-th exposure.

Memory activation is converted to a probability of successful recall using a logistic function:

$$P(t) \;=\; \frac{1}{1+\exp\!\left(-\frac{w(t)-\gamma}{s}\right)}, \tag{8}$$

where γ is the retrieval threshold (the minimum activation required for successful recall), s is the activation noise controlling stochastic variability in retrieval. Together, equation 7 and equation 8 fully specify how ACT-R model predicts forgetting curves as a function of practice timing.

Why Mahalanobis distance? Human experiments report many (d, s, γ) triplets, forming an ellipsoidal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. We evaluate the plausibility of the fitted ACT-R model parameters $\hat{\boldsymbol{\theta}} = [\hat{d},\, \hat{s},\, \hat{\gamma}]^\top$ with respect to human reference distribution characterized by $\boldsymbol{\mu} = [\mu_d,\, \mu_s,\, \mu_\gamma]^\top$ and a covariance matrix $\boldsymbol{\Sigma}$. Given $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, the squared Mahalanobis distance is

$$D^2 = (\hat{\boldsymbol{\theta}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{\theta}} - \boldsymbol{\mu}),$$

which normalizes deviations by considering parameter scales and correlations. We report an interpretable similarity score $\mathrm{Score} = \exp(-\frac{1}{2}D^2)$, interpretable as a Gaussian RBF under the Mahalanobis metric, to account for differences in parameter scales and their empirical correlations. Score measures how close a model's retention behavior is to that exhibited by humans. A small value indicates that the model's retention parameters fall within typical human ranges, whereas a large HRS-MD indicates atypical or non-human-like retention dynamics.

Estimating the covariance matrix $\boldsymbol{\Sigma}$. Table 4 reports the reference means and standard deviations of human ACT-R parameters from prior studies. Since only marginal statistics are available and raw human samples (hence empirical correlations) are not, we construct the covariance matrix from these reported scales. Let $\boldsymbol{\sigma} = [\sigma_d,\, \sigma_s,\, \sigma_\gamma]^\top$ denote the standard deviations and $\mathbf{D} = \mathrm{diag}(\boldsymbol{\sigma})$.

By default, we adopt the independence approximation, assuming zero cross-covariances:

$$\boldsymbol{\Sigma} = \mathrm{diag}\big(\sigma_d^2,\, \sigma_s^2,\, \sigma_\gamma^2\big).$$

This choice reflects the fact that no reliable correlation estimates are available and reduces the Mahalanobis distance to a multidimensional z-score distance.

A.3 Model architecture and parameters

GBCL. SGD, ER, and EWC employ the neural network structure. The neural network takes as input the current discrete state and a task identifier. By default, both are embedded through separate layers of size $\frac{d}{2}$ (with $d = 64$), concatenated, and passed to a multilayer perceptron with two hidden layers and ReLU activations: The output produces logits over the next-state space, which are transformed into probabilities via softmax. Weights of linear and embedding layers are initialized with Xavier uniform initialization, and biases are set to zero. Unless otherwise specified, we set $d = 64$, `num_states`$= 4/8$, and `num_tasks`$= 2$. Optimization in all cases uses stochastic gradient descent with cross-entropy loss. While the network architecture is identical across methods, their training hyperparameters differ. For SGD, the model is updated online using only the current sample with a learning rate of 0.001. In ER, a replay buffer is employed with a capacity of 8000 samples, a replay ratio of 0.5, and a batch size of 32. For EWC), the model is trained with the same online updates as SGD but augmented with a quadratic penalty weighted by the diagonal Fisher information matrix; the optimal regularization strength is set to $\lambda_{\text{EWC}} = 700$.

LLMs. LLaMA3-8B (Meta, 2024), DeepSeek-R1 (DeepSeek, 2024), MAMBA (TII, 2024), and RWKV-7 (RWKV, 2024) are used in their publicly released configurations without further fine-tuning. LLaMA3-8B is a transformer-based decoder model with approximately 8 billion parameters and a context length of 8K tokens. DeepSeek-R1 is a transformer model with roughly 7 billion parameters, optimized for reasoning tasks, supporting up to 8K tokens. MAMBA is a selective state-space model with about 8 billion parameters and an extended context length of 16K tokens. RWKV-7 adopts a hybrid RNN–transformer architecture, with around 7 billion parameters and a context length of 4K tokens. For all large models, we adopt the default inference hyperparameters recommended by their authors: greedy or temperature-controlled decoding with temperature in $[0.6, 0.8]$, maximum generation length matched to the context window, and no additional training beyond the released checkpoints.

A.4 ICCL Benchmarks

DMC benchmark: To illustrate how a Discrete Markov Chain (DMC) prompt is constructed, we provide an example containing both the target task transition matrix P_{target} and the interference task transition matrix $P_{interference}$. P_{ij} represents the transition probability from state i to state j. During prompt construction, samples are first generated based on P to form the target task sequences, and later partially replaced by samples drawn from $P_{interference}$ to simulate interference phases.

Table 5: Example transition matrices for the DMC benchmark with total 4 states. P_{target} denotes the target task transition matrix, and $P_{interference}$ denotes the interference task transition matrix.

P_{target} (Target Task)			
0.2694	0.2987	0.2681	0.1637
0.2161	0.0106	0.3997	0.3736
0.2489	0.2278	0.2645	0.2588
0.3021	0.2828	0.2460	0.1691

$P_{interference}$ (Other Task)			
5.6×10^{-8}	0.50	0.50	5.6×10^{-8}
5.6×10^{-8}	0.50	0.50	5.6×10^{-8}
5.6×10^{-8}	0.50	0.50	5.6×10^{-8}
5.6×10^{-8}	0.50	0.50	5.6×10^{-8}

Examples of SP, MP and DP are shown below. Each prompt begins with the instruction *"Learn pattern, predict next state. [SWITCH_TO_INTERFERENCE] = interference pattern starts. [SWITCH_TO_NORMAL] = target pattern resumes. Predict next:"*, followed by a continuous sequence of sampled states. The control tokens `[SWITCH_TO_INTERFERENCE]` and `[SWITCH_TO_NORMAL]` mark the onset and offset of the interference period, respectively, allowing the model to experience alternating target and interference tasks within a single context.

SP prompt example (DMC benchmark)

```
Learn pattern, predict next state. [SWITCH_TO_INTERFERENCE] =
interference pattern starts. [SWITCH_TO_NORMAL] = target pattern
```

```
resumes.  Predict next:
013110301001313012331213202301331010200102331222100120122123222323012
2231320223210202233032023201203[SWITCH_TO_INTERFERENCE]21211222122112
211211111222122122111111121222111211111212221122122222211111222121111
11111211111222222221222222121222112111222222112211221211121121211121
2222221112211121221111221122121111111111212112221121112221121211212211
12211121111212122112221122221211211211112111222121222112221111111111
222112221211111121122222121212111111211221122222211221221211111122212
121212122111211222112222211122111211222112112212122221112121121112112
212221212221222211112211221222121112112211122122121112222211222222211
2111222112122221112222112211111111112112112221122122222222222112222212121
2122211221212112122112221111112111222222222222221122211121211212122221
111112121121211222211221222221222122222112222212212221111212211212222221
2121121221111212212212222221212121211112222121221121212121212112222122
11221222212121121122221211121221112222121222121111111222211222112122221
122221112212222122221111121112111212211211121212212222112111121122112
222112112221221211211121122121221122111112112222221122111111212122121
212221221111111221211221212112112222121211111211112222121221211122122
222112121221121121121212122212211112222122211212221111222212122211222
1212122211211112121211111122222222211112122121212112211122222121111111121
222211111211122221122212212112122121122221112222111222221222111211211
1111112112212112111111221121122111121221112111222212111222121111212121
211122222121212112222121211221222212112222122221211222211222112121111
22112121221222222111112112112211211122111222221122112221112121211222221
1122222211212222212222122212112112222122122122211212211221122121212111
122211122222221222221111112212121111122221112121122121111122212122222
2212112211112211221221121211212211222211121222212222221211212111121222
122221222121121211211122222222222212212221111122212222212212211222111121
1122112222112212122212112212212112112222122211122112111212112112212111
1122112211211112222122122212221122121121111222122211111112121221222221221
212222121211222122
```

MP prompt example (DMC benchmark)

```
Learn pattern, predict next state.  [SWITCH_TO_INTERFERENCE] =
interference pattern starts.  [SWITCH_TO_NORMAL] = target pattern
resumes.  Predict next:
013110301001313012331213202301331010200102331222100120122123222323012
223132022321020223303202320120330212223022002301312000122133132100010
121223121200001303321022023223310101233030100001210300001223231320233
323020222213120231322013201221212130021301220323222010230013013011232
013312010010100313013221230102312130201313300133120212002230230132301
332303102013010030123131301312022201212012101323002331301012131023133
130213121010200122013332310220321310120023202030202022121310222202232
21212310020033200[SWITCH_TO_INTERFERENCE]2112212122221112121121112112
212221212221222211112211221222121112112211122122121112222211222222211
2111222112122221112222112211111111112112112221122122222222222112222212121
2122211221212112122112221111112111222222222222221122211121211212122221
111112121121211222211221222221222122222112222212212221111212211212222221
2121121221111212212212222221212121211112222121221121212121212112222122
11221222212121121122221211121221112222121222121111111222211222112122221
122221112212222122221111121112111212211211121212212222112111121122112
222112112221221211211121122121221122111112112222221122111111212122121
212221221111111221211221212112112222121211111211112222121221211122122
222112121221121121121212122212211112222122211212221111222212122211222
1212122211211112121211111122222222211112122121212112211122222121111111121
222211111211122221122212212112122121122221112222111222221222111211211
1111112112212112111111221121122111121221112111222212111222121111212121
211122222121212112222121211221222212112222122221211222211222112121111
22112121221222222111112112112211211122111222221122112221112121211222221
1122222211212222212222122212112112222122122122211212211221122121212111
122211122222221222221111112212121111122221112121122121111122212122222
2212112211112211221221121211212211222211121222212222221211212111121222
```

```
12222122212112121121121112222222222221221222111112221222212212211222111121
1122112222112212122212112212212112112222122211122112111212112112212111
1122112211211122221221222122211221211211122212211111112121221222221221
2122212121211222122
```

DP prompt example (DMC benchmark)

```
Learn pattern, predict next state.  [SWITCH_TO_INTERFERENCE] =
interference pattern starts.  [SWITCH_TO_NORMAL] = target pattern
resumes.  Predict next:
0131103010013130123312132023013310102001023312221001201221232222323012
22313202232102022330320232012033[SWITCH_TO_INTERFERENCE]1211222122112
211211111222122122111111121222111211111212221122122222211111222121111
111111211111222222221222222121222112111222222112211221211121121211121
2222221112211121221111221122121111111111212112221[SWITCH_TO_NORMAL]230
102312130201313300133120212002230230132301332303102013010030123131301
31202220121201210132300233130[SWITCH_TO_INTERFERENCE]1111121122222121
212111111211221122222211221221211111122212121212122111211222112222211
122111211222112112212122221112121121112112212221212221222211112211221
2221211121122111221221211122222112222222211211[SWITCH_TO_NORMAL]232210
213321213131231012001000131020022212312222231232101323213020313121222
23131231321033310012131001[SWITCH_TO_INTERFERENCE]2222222222221122211
1212112121222211111121211212112221122122222122212222112222122122221111
2122211212222221212112122111121221221222222121212121111222121221121212
1212121122221221122122222121211211222212111[SWITCH_TO_NORMAL]303310031
312123202000100032221023300303323012222120320323212322101203012301021
33002001313032023312131[SWITCH_TO_INTERFERENCE]1112112211222211211222
12212112111211221212211221111121122222211221111112121222121212221221111
1112212112212121121122221212111112111122221212212111221222221121212211
2112112121212221221111222212221121222[SWITCH_TO_NORMAL]12123233130233
013231313133300301220203010000323222231210203213020300220123133130121
200130232312001301[SWITCH_TO_INTERFERENCE]122221122212212112122121122
221112222111222221222111211211111111211221211211111122112112211112122
111211122221211122212111212121211122222121212112222121211221222212112
222122221211222211222112121111221121212212222221111121121122112111221
112222112211222111212121122221112222221121222221222212221211211222212
21221222112122112211221212121111222111222222212222111111122121211112
222111212112212111112221212222222121122111122112212211212112122112222
111212221222222121121211112122212222122212112121121112222222222212212
221111122212222122122112221111211122112222112212122212112212212112112
222122211122112111212112112212111112211221121112222122122212221122121
121112221221111111212122122222122121222121211222122
```

REGBENCH benchmark: We also construct a symbolic sequence benchmark based on RegBench, where both the target and interference tasks are generated by probabilistic finite automata (PFAs) implemented as deterministic finite automata (DFAs). Concretely, we first sample a target DFA and an interference DFA with different numbers of states, alphabet sizes, and random seeds. Each DFA induces a stationary stochastic process over a finite alphabet, and we generate short examples by running the corresponding DFA for a random length between $L_{\min}$ and $L_{\max}$ symbols. We concatenate such examples into a single long sequence and insert control tokens `[SWITCH_TO_INTERFERENCE]` and `[SWITCH_TO_NORMAL]` whenever the generator switches between the target DFA and the interference DFA. This yields a context in which the model must continuously track the active PFA and predict the next symbol in the sequence.

SP, MP and DP examples are explained as follows. In our setup, the target task is produced by a DFA with *4 states*, *alphabet size 10*, *maximum 2 outgoing edges per state*, and a fixed random seed (`seed=42`) to ensure reproducibility. The interference task is generated by a distinct DFA with *6 states*, *alphabet size 20*, and *maximum 3 outgoing edges per state*, ensuring a substantially different symbolic distribution and transition structure. Each DFA defines a stationary symbolic process, and we generate short examples by running the corresponding automaton for a random length sampled from $L \sim \mathrm{Uniform}(10, 15)$. We concatenate $N_t = 320$ such examples to form a prompt. We additionally generate a separate test split consisting of $N_{\text{test}} = 10$ pure target-DFA examples, whose

lengths are sampled from Uniform$(8, 10)$. This test set assesses how much target-pattern ability the LLM retains after experiencing last historical demonstrations in a prompt.

SP prompt example (Regbench benchmark)

```
Learn pattern from probabilistic finite automate(PFA) sequence.
[SWITCH_TO_INTERFERENCE] = interference pattern starts.
[SWITCH_TO_NORMAL] = target pattern resumes.  Predict next:  m n m n m n
mnmssw,mssmswmnmsswmn,msswmssmsw,mnmssmsmswmssw,mssmsmswmss,mssmsw
mssmswmss,mnmsswmnmnm,mnmssmsmsms,mnmssmsmswmnm,mssmsmsmsw,mnmsswm
nmnmsswm,mssmswmssmswmss,msswmssmsmswms,mnmnmsswmssm,msswmsswmn,mnm
nmssmsm,msswmssmsmsmsms,msswmnmnmnmn,msswmnmsswmnmss,msswmnmnmsswm,
[SWITCH_TO_INTERFERENCE], nvemxxdrux,dpaxuqrtem,dxdrdnmqrure,dxuqnmqd
pe,rtemxpevevaxdn,ngdpegdxdn,ruxpapaxuqr,nvapaxtevaxddxd,ngngngrteve
mra,dpaxdnvegdx,rtaxtevaxura,nvegrtapeve,ngrdruqrtemx,ngrtapaxuqdx,nv
emqrtaxure,ruqdxuqngnmrax,nmxpaxddxdnv,rtaxuxpevapaxd,dxdruqdxurapemr
,ruqnmregrtev,dxtevaxurapa,rdnvemremxxddp,ngdpaxdngng,dpaxtemqdpevem
,dpegnmregrdr,dxdngnvapevap,ruxxtaxureveva,ngruqdpemregrdd,dxtaxddpev,
nmqrddpegnveg,nmxpegnmxp,dxurapegdpemxp,dxurevevaxdngrd,rddpevemregnv
,dxdngdxdngru,rdnmxxdrdrte,ngdpapapemq,rtapevegdpapeg,rurapemxpegruxx,
dxuqdpegnva,dxtapapegngdx,nvegruqdxdnvape,dxtegrtegnmqdpe,ruxxuxxtemq,
ngrtevegnmra,dpaxuqruxxdnvax,ruxpemraxddp,rddxuxxuqnmrev,nvegrtapegdx
ddp,dpapemrapegdpap,nmrevemraxuxpev,dxdrdngdxd,rtegdpemxxur,nmqngruqn
mxpem,rdrdnvaxddxdd,dxddpegdpapaxur,dxdruqrdrura,nvemrevapaxu,dpemrem
xxuqngrd,nvemqnmrevap,nmqdpapape,nmraxddxtap,rtaxurapevemxp,rtaxuxxdr
teveve,ngdxtaxtegruqrt,rtapaxtaxta,nvapapapapapeg,dpegdpemqdpa,dpegnm
rape,dxdrdnvapaxtev,dpaxuxpapa,rtemrevemqnmxx,dxdnmqnmxpapa,ruqngrdng
rdnm,nvapaxurapaxd,dxtapaxuqrddx,ngngnmqdxddpa,dxuxxddpaxdr,nmqrdrdru
remxx,ngrtemremr,nvevevevapapa,nvapevegrt,dxuqrtegnvevegd,rdnvevaxtap,
ngnmregnmq,dxtegdxuxpa,ngruxpevegdxura,dxuqdpemrapegru,dpevemqngruqnm,
nmqnvemxpe,nvegdpaxdn,ruraxddxdng,dxddxtemxpapemx,dxurapaxddxddxu,dxu
qdxuqngn,ngdxdrurevegdx,dpaxdrdrdn,dpemrapape,dxuremxxurem,dpevapaxuq
rdd,rdruraxureg,nvaxtevapapa,nvevemqruqnveg,rddpapaxddpap,dxuqdxtapax,
dxtapaxurapax,ngruqrdrta,ngngrddpeg,dpegnvaxurax,ruqdxdrddxd,rtemxxta
xtaxdn,rdruraxdruregdx,nmqngruxpa,ngdxtapaxtaxtap,ngnvegdpap,ngrtapev
axteve,dxtevapegnmxpe,dpaxuxxddpeg,ngdxtemqdpegnmq,nvapegdpaxur,rdrux
papemrem,dpapevaxte,dpemraxuqnvap,ngdxdruxxdn,nvapevevevapemx,ngnmreg
rteg,dpaxuqrdruxxdnm,rtevevevevegdxd,rdrdrddxtegrdr,nvemqnvapapegdp,r
tevemreva,nvaxuqruqdpe,ngruxxuxpapemr,nvevegrdruxpe,dxddxtevapaxur,ng
nmqrtegdpapem,ngdxddxdnvevax,rddpegnvaxuremq,nmxxtevaxdnveve,dxtegdxd
ru,nmraxdnvegnmr,ngdxddxdruxpape,nvaxuqnmqr,nmqnmxxuxpevax,nmremqrtev
axu,dpaxtapapemra,rdngdpevap,rtaxdrtemqdxta,ruregrurapegdxu,dpevaxure
v,rtaxuqrtapegdx,rdngruxxdrdngr,dxtaxdngdpemrem,dpaxddxddpaxd,dpapape
gnvax,rdngrddxuraxdd,rddpaxurap,dxuxpapaxdr,dpegngdxuqrdn,nmrevaxdngr
tem,dxdnvemxxdrur,dpevegnvevapapa,rtegrtemqrtapa,rtevaxuxpaxu,rtemqdp
apaxt,nmqruqrurapemq,nvaxuremxpeg,nmxpaxtemxpa,ngruqdxtegr,nmqngnveva
x,rurapemrape,rdrtapaxtapaxu,dxtapemxxdnm,rtapaxuqnmraxux,dpaxddxuxxt
e,nvegngdxta,rdngdxddxuxp,nvaxdrurapaxdrd,nvapevemraxtev,dpapemqnvapa
,dxtevapaxt,rtaxtegrdr,nvevemxxtaxtem,dxuqngdxdngng,nvapemremra,ruxxt
apevev,ngngngnmra,dpaxuxpevap,dxdngnvegdpe,rtegruraxtegrt,ngnmqrtemqr
ur,dxuremraxdrdngd,dpegdxuqrdnm,dxtapevapemqn,dxuqdpaxdn,ngnvegnmxpem
qnm,ruxpegnvaxdrd,rdrtemqruxpaxte,rddxuremxpax,rtevaxuraxdr,ngngrtegn
gdxu,ngdpevapem,dpegruqruqdpax,rdnmqdxuregrte,rtemqdpevevaxu,rddpegdx
dnveve,dxuxxtegrddpemq,dxddpevapape,nvemxpapeg,nmqdpapapegn,nmraxurem
regrt,nmrapapapapem,ngdxdnvaxte,ngngngngnmraxdn,dxtapaxdnmx,dpemraxur
ax,rddpapapevax,nvapevaxddpax,nvegrdrtegru,dxtaxtapaxuregn,dpegdxtegr,
nmraxddpapeva,ngruqngrur,dpemraxuraxdrdn,rdnmxpaxdd,dxuxxtapaxdrd,ngd
pevapegnvax,nvemrapapegrt,nvaxuxpevaxtax,dxtaxurapapev,dpegdxdnmq,dxd
rtegdpaxu,nmraxtapeg,ngrtemqdxtemxp,nvaxuxxuxpe,dxdnmrevaxd,ruxpemrax
u,ngdxdnmqrtev,dpemqngruqd,rurapaxtaxte,dxtemraxdd,dpemqnmremqnmra,dp
egnveveg,dxtevaxtemqrd,nmqnvaxuxpaxu,rdrddpevaxddxd,nvegrurevax,dxtax
tegruqru,ngrtemregrtapa,rdngdpemxxtapa,rddxdruraxdrure,dxddxddpax,ngn
gdxdnmreg,ngruxpapaxuq,rtegngngdpem,dxdrdrdruqdxd,dxdrtapegdxuxpa,ngr
taxdnva,nvaxdrtaxtegrta,rdrtaxtaxu,rdngnmxpegdpape,dpaxuraxtaxtemr,rd
```

```
ruxxurapapem,rdngrdngrdnva,rddpaxuxxu,ruregdxddxtap,nmxpevapegrdrdn,n
vemxxddpemx,ruxxdruxxd,dpapegngrdrte,dxdnmqrtegrtapa,ruqdpaxtemrape,d
xddpaxuqrta,dpaxuqrddpe,nmrapemqruqd,nvegrdnmxxdrtem,dpaxdruqru,nmqdx
drdngdpemr,dxtemraxddpa,dxdnvaxuxxura,ruxpemxpaxuqdpe,nmremqnmremx,dx
ddpapaxtap,dpegdpemqrteve,dxdrurapegrd,ngdxtegruqrdrd,ruxpapegdxta,rd
dxtevaxdn,ruraxuqnvaxuqng,dpaxtevapaxta,dxdnmxxuraxdd,ngrtaxtaxdngngd
,nmqruqnmxpape,nmxxtevaxura,nvevapemqruqd,ruxpevevegdxuq,ngdpevevemqd,
rddpapemxxu,nvapemremr,dpevevemqru
```

MP prompt example (Regbench benchmark)

```
Learn pattern from probabilistic finite automate(PFA) sequence.
[SWITCH_TO_INTERFERENCE] = interference pattern starts.
[SWITCH_TO_NORMAL] = target pattern resumes.  Predict next:  m n m n m n
mnmssm,msswmnmnmssmsm,mssmswmssm,mnmsswmssmswmn,msswmssmswm,msswmnms
swmnmss,mnmssmsmsms,mnmsswmsswm,mnmsswmssmsms,msswmsswmn,mnmssmsmsms
wmnm,msswmnmsswmnmss,mssmswmsswmnms,mnmnmssmswms,mssmswmnmn,mnmnmssw
ms,mssmswmsswmsswm,mssmsmsmsmsm,mssmsmswmnmnmss,mssmsmsmswmnm,mnmnms
swms,msswmnmnmn,mnmnmsswmssm,mssmsmsmsm,mssmswmsswmssw,mnmssmswms,ms
smswmssms,mnmnmnmssmsmsms,mnmssmswmnmnms,msswmnmnmss,mssmsmsmsmsw,mn
mnmsswmss,mnmsswmsswmn,mnmsswmnmnms,mnmnmsswmnms,msswmnmssmswmn,mnms
swmssmsm,mssmsmsmswmssw,mnmnmsswmssmsms,mssmsmsmswmn,mnmnmnmssmsm,ms
swmnmnmsswms,mnmnmnmssms,mnmssmsmsmswms,msswmsswmnms,mnmnmssmsmsms,m
ssmswmnmnmnmn,mnmnmssmsmsmswm,mssmsmsmsm,mnmssmswmssms,mnmsswmssw,ms
smswmnmnmssw,mssmswmsswmnmss,msswmnmsswmnm,mnmnmssmswms,msswmnmsswmn
,mnmnmsswmnm,mssmsmswmssmsw,mssmsmsmswmsswm,mssmswmnmss,mssmsmsmswmnm
,mnmnmssmswmnmnm,mssmsmswmnmssms,msswmssmswm,mnmsswmnmssm,mnmssmsmsm
smswm,mssmswmsswms,msswmssmsmswmn,mnmnmssmswmnmnm,mnmssmswmsswmnm,mn
mssmswmnmsswm,mssmswmssw,mssmswmnmssw,mnmssmsmswmssw,msswmsswmnmnm,m
nmnmssmswmsswm,mssmswmssmsm,mnmnmsswmnms,mnmsswmssmswmss,mnmnmsswmnm
n,mnmssmsmsw,mnmssmsmsms,mssmsmsmsmswmn,mssmsmsmswmssms,mnmssmsmsmsm
sms,mssmsmswmnm,mnmnmnmssmswmn,mnmsswmnmssw,msswmsswmn,mssmswmsswmnm
n,mnmssmsmsw,mssmswmssmswmn,mssmswmnmssms,msswmnmnmnmss,mnmnmnmssmsm
s,mnmnmnmnmssms,mnmnmnmnmnmss,mnmnmnmssmsw,mnmssmswmsswms,mnmnmssmsw,
[SWITCH_TO_INTERFERENCE], dxtegdpemxxte,dxdruqnvax,rdrdnmraxdrdrur,ng
dxuqdxdrt,dxdnvaxdnm,ruqrurevape,dpaxtapegruraxd,rurevapaxuxpapa,rtax
taxdrtemqn,dpemqrtapa,dxtapaxdnm,nvaxdnvegdp,rdnvemxpapegngn,rdruraxu
xpegrur,rdrdrdrddpe,dxuqngruxxurem,ruqdpapape,rtapemregn,rdruxpaxurap,
rtaxuxxuregnv,ngnvaxdruxx,dxddxtemxxuq,dxtevaxtaxddpa,ngrtegruxxura,r
urevemqruq,rddxtevevevem,dxtaxtapem,dpevapapax,rtapemqdpaxd,nvevegngr
dn,nmxpaxddxdruxx,ngnvemqngruxxuq,dpemraxtax,dxuqdxtemqdxddp,dpemxxux
xt,dxtemxpemregdp,rddpevapaxtapa,ruqdpaxtegnv,dxuqdpaxuxpaxte,dxdrure
grure,ngnvapegrteva,ruqrtemqdp,rtapemremqdxt,dpemxpegdpe,dxuxpevegdxt
apa,dpegrtapegr,rtemregngrtegdp,nmxxtegdpeveveg,ngngngruqnvapa,dxtaxd
dxtegrura,nmxxtapaxu,dxuqrdnvevap,dxtapevapegrte,dxuqnvapaxtap,rdnvem
xxdruqru,dxdnmxxuregdpap,dpemxxdnmqnmre,ngrtaxddxuregnm,dpaxdnmregdxt
ev,ruqnvaxtax,dpevegdxuregr,dxurapegnvapegr,dxuremqnmx,dpemxpevapaxur,
dpapaxtegdxdr,rtemqdpegngrd,ngdpapaxdr,nmregnmxpevemq,nvapapevevapaxu,
ruxxddxure,nmqruqnmqrurax,ngdxtegdpapeva,rddxdnmrapapegn,rtemxxuxxtem
x,rtegrtapemqd,ngdpapemraxdnv,ngrtemremr,rdrtaxtemxp,ruxpevaxuqngd,dp
apevegdxteva,ruxxdrtaxtaxu,rtaxtaxdnmqnvax,nmrapevaxtevax,nmxxddxtaxd
r,nmraxuxxtemq,dpegrdnvaxurax,dxddxuxpapap,dpapemqngngr,dpapevemreg,d
pemremxxdr,nvaxuxpemxp,ngnmqnmqdxtemr,rddxtapaxteg,nmraxdrddpaxdrt,rt
emxxuraxdr,dxurevaxdr,ngdxuqnvevap,dxddpaxuqnmqngn,dxdrtevaxdrdnm,rte
grtemqruq,ruqrdrtemq,nmqdxtegng,dxtegngrdruqrt,rdrddpemxxuqr,dxdngnva
pax,nvaxddpaxte,dpevevegrd,rtemrapaxdn,rdnmremraxta,nmxpapaxuqnvax,dx
dnmxxuxxtax,rdruxpaxdngnmra,rtaxuqrdngdp,rddxuremxpaxd,rdrdruqdpe,dpe
mraxtapapemx,nvapaxddxdngn,ngnmraxtegnmqdp,ngrdruxpapem,nmxxdrurevap
,dpevaxurevemr,dpegrdrtap,rtapapegrdrtem,ngdpevevegrtev,nmxpevapegdxd
r,ngruxxuraxdrdn,rdrtemxxtapapax,ruxxuxxdnmxp,dxtapaxuxp,dpevaxtegnve
,dpevevapapaxte,dpaxuraxtevap,dpemxxddxdr,dxdrdrdrdnvemxx,rddxuregdpa,
rtapevevemq,ngruqnvapemq,dxuraxddpevem,dxurapaxurap,rddxuqrtemrevem,r
tapemqrux,dpaxuxxtegnmq,dpapemraxu,rtapemrapemxpax,ngdpapemxx,ruraxdd
```

pemxpa,dxuxpemrapemqd,dxuxxuxxtapax,dxddxtapemqdxd,rddxdruraxure,ruxx
uqnmxx,rdngdpevaxdr,dpaxddxtax,dxtegnmremxpap,dxddxtevapa,rdnmxpaxddp,
nvapapemra,dxuraxtegdpe,ruraxdrtaxu,nvaxuqrdruqr,ruqrtevegr,rtapemxpa
pemxpe,rtapemregr,rddpevemxpegn,dpemqdxuxpemr,ngngrtaxuraxdn,dxtegrur
emq,rddxddpapemxx,dpaxtapaxtevax,ngdxuxpapemrax,ngruxpapevapeva,rdnve
grtem,dxdruqnmxpap,dxtapaxuqdpe,nmxxdrdruxpa,rdngngngrdrdn,rdngdxurax
drtax,dpaxuraxdr,dxddpaxddxtegdx,ngnmqrddxu,ngdxdngngruxxta,rtemremre
mregnv,ngnvaxurevegng,ngdxtaxuxpemq,ngrtemraxu,nvegruraxuqdp,dpapaxdn
grtapax,dxtapegruxpa,nvaxtapaxt,rtegruqrtapev,rdnmxxtegrtevax,nvevaxu
qngrdru,rdnvaxdrdnmr,ruqdpegnvap,dpaxtapegrdr,dxtapaxtaxtaxux,rtemxpa
xta,dpevegngdxuxpax,rddpapemxxur,ruxxddxuxxure,nvapapapemrevap,dpegnm
qnvapa,rdnvaxuremqn,ruxpapapegdpev,ruxpevevapap,dxuqdpegrdngng,nvapeg
ruremq,ngrddpemqnm,nvemqrddxdruqdp,rtemqnmqrddxd,rdnmxpevevegr,dxtemr
emqnmreve,dpegrddpapegn,dpaxdnmqdxur,dxtevapaxtaxu,nvapaxtegruqrd,dpe
gnmxxtaxu,ngrtegngnmr,dxdnvapapa,rtaxtegnmxp

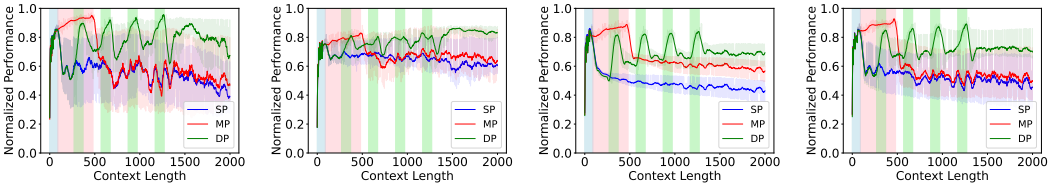
DP prompt example (Regbench benchmark)

Learn pattern from probabilistic finite automate (PFA) sequence.
[SWITCH_TO_INTERFERENCE] = interference pattern starts.
[SWITCH_TO_NORMAL] = target pattern resumes. Predict next:
mnmnmnmnmssw,mssmswmnmsswmn,msswmssmsw,mnmssmsmswmssw,mssmsmswmss,mss
mswmssmswmss,mnmsswmnmnm,mnmssmsmsms,mnmssmsmswmnm,mssmsmsmsw,mnmsswm
nmnmsswm,mssmswmssmswmss,msswmssmsmswms,mnmnmsswmssm,msswmsswmn,mnmn
mssmsm,msswmssmsmsmsms,msswmnmnmnmn,msswmnmsswmnmss,msswmnmnmsswm,mnm
nmssmsm, [SWITCH_TO_INTERFERENCE], nvaxuqdxdd,nvegdxdrdnve,nmraxdruxp,
dxuxpegruqnmxx,rurevapapa,dxddxdrtemx,rtapaxtemrapemx,ruqnmremrapeva,
ngnvemqdpem,dxtaxdruqnva,rtevemqdpap,ruxpevapapeg,ruxxtemqnmra,rtegrd
rtapap,dpapapaxuqdpax,rddpaxdnmxxt,dxuqngnvaxurap,nmxpemrapemxpev,dxt
apapegrdr, [SWITCH_TO_NORMAL], mnmnmnmsswmn,mssmswmnmssmsm,mnmnmnmssw
m,mnmsswmnmnmssm,mssmsmsmswms,mnmnmsswmnmnm,msswmsswmnmnmn,mnmnmsswmn
mnmss,msswmnmnmn,mnmsswmssmswm,mnmssmsmsm,msswmsswmnmssm,msswmssmsmsw
mss,mssmswmssmswm,mnmnmsswmssm,mssmswmssmsw,mnmnmssmswm,msswmnmssmswm
s,msswmnmnmssmsms,msswmsswmss,msswmnmnmsswm, [SWITCH_TO_INTERFERENCE],
rtevevaxtemqngd,nmqnvemxpapapeg,dpemrevapem,ruxxdrdnmxxu,nvaxuqdxdrd
rtax,dxddpevaxdnv,dpemremraxdngr,rtevemregnvapap,nvapegnvegnvemr,rd
nvapevaxuxpem,nmxpemraxd,dxdnvegdpape,rdrtegnmqdpegd,dpegrtaxdnvev,n
mxxddpapapaxur,nmxpaxddpaxt,rtegngrtemqn,ngdpapegnvaxuxp,rtegrtevemq
d, [SWITCH_TO_NORMAL], mnmsswmnms,mnmsswmnmnm,msswmnmnmnmssw,msswmnmn
mssmswm,mnmsswmnmnmnmnm,msswmnmsswm,mnmnmnmsswmssw,mnmssmswmssm,mssm
smsmsw,msswmssmsmswmn,mnmsswmnms,msswmssmswmssw,msswmsswmsswm,mssmswm
nmnmss,mnmnmnmsswmnm,mnmnmnmnmsswm,mnmnmnmnmnmss,mnmnmnmsswms,mnmssw
mssmsmsm,mnmnmsswms,mnmnmsswmsswm, [SWITCH_TO_INTERFERENCE], rtapemxx
dr,nmrapapemqnmrax,dpaxuqrtaxd,ruqdpevapa,nvapapapegn,ruxxdnvapaxura
x,nvaxuxpevapevev,ngdxddxtapaxte,rdrtapegng,rtevegnmxx,dpemqdpapap,n
mxxtapegnvegdp,nmrevemrevapaxu,nmraxtaxtem,rurapevapapeva,nvaxddpegr
,ngdpaxureg,nmrapegrureg,ngdxuxxurapev, [SWITCH_TO_NORMAL], mssmsmswm
ss,mnmnmnmnmnms,mnmnmsswmssmsw,mssmswmsswmss,mnmnmsswmss,mnmnmnmnmss
wm,mnmnmssmsw,mnmsswmnms,mnmssmsmswmn,mssmswmsswm,msswmssmsmswmn,mss
msmswmnmnmnm,mnmsswmnms,mnmnmnmnmnmnmss,mnmsswmnms,mnmnmsswmnmssm,mn
mnmnmsswmnms,mnmsswmnmssm,mnmnmnmnmnmssms,mnmnmnmssmsw,mssmsmsmsmsms,
[SWITCH_TO_INTERFERENCE], nvapemqrte,ngngnmraxtaxd,rurapevemxx,rtapem
rapaxddpe,ruqdxdngrdn,ngrtaxddpegngrd,dxdrdrdrdruremx,dpegdpaxtevegd,
rtegrtaxdngnveg,dxdrddpemr,rtaxuqdpaxux,ruxpemrevapegn,rtemxpegdxddp,
nmxpaxdrtevaxu,ruqdxdrurapapeg,rurapaxdrtemra,dpapevaxuraxddx,rddxtem
rapaxdru,nvapapapev

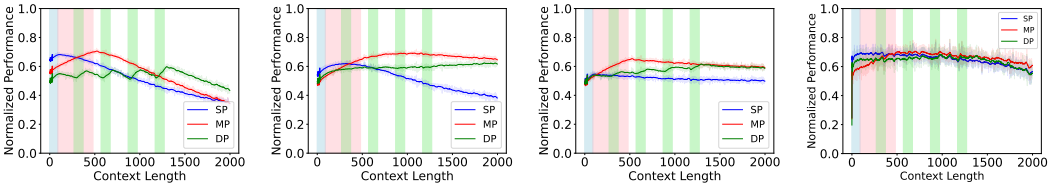
A.5 Retention Performance under SP, DP and MP for DMC regbench

Figure 6 illustrates the retention performance of ICCL models and GBCL baselines varies on the simple task with task identifiers and $\varphi_I = 200$, under SP, MP, and DP scheduling. For ICCL methods, we observe that in both SP and MP conditions, the retention performance drops sharply immediately after finishing all target task demonstrations, and subsequently remains at a reduced level as the

sequence length increases. In contrast, ICCL under DP consistently achieves better performance when the sequence length reaches its maximum, demonstrating its advantage in retaining cross-task knowledge. However, for GBCL baselines, DP does not yield a comparable improvement and performance suffers sharp drops across sequence lengths. These results confirm that both ICCL and GBCL models suffer from catastrophic forgetting, yet DP is effective in mitigating this issue for ICCL by alleviating performance degradation over long sequences.



(a) DEEPSEEK-R1 (b) LLAMA3-8B (c) RWKV-7 (d) MAMBA

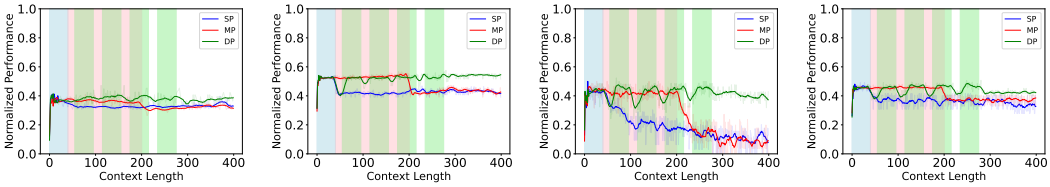


(e) SGD (f) ER (g) EWC (h) PGP

Figure 6: Retention performance of ICCL and GBCL models on simple task under SP, MP, and DP settings (with task identifiers, $\varphi_I = 200$). Target task blocks are highlighted in light blue (SP), pink (MP), and green (DP).

A.6 Retention Performance under SP, DP and MP for REGBENCH Benchmark

Figure 7 presents the retention performance of ICCL models on the RegBench task with task identifiers ($\varphi_I = 20$, $\varphi = 40$, $K = 5$, and $\varphi_D = 120$) under SP, MP, and DP scheduling. Consistent with the observations on the DMC benchmark, all ICCL methods under SP and MP experience an immediate and pronounced performance drop once the demonstrations for the target task conclude, after which their accuracy stabilizes at a lower level as the sequence length continues to grow. In contrast, under DP scheduling, ICCL models maintain consistently strong performance without exhibiting the characteristic sharp decline even when the sequence reaches its maximum length, highlighting DP's advantage in preserving cross-task knowledge. Evaluating ICCL methods on both DMC and RegBench benchmarks further demonstrates that ICCL is effective not only in purely synthetic settings but also in language-style sequence environments, indicating strong generalizability across heterogeneous task formats.



(a) DEEPSEEK-R1 (b) LLAMA3-8B (c) RWKV-7 (d) MAMBA

Figure 7: Retention performance of ICCL models on regbench benchmark under SP, MP, and DP settings (with task identifiers, $\varphi_I = 20$). Target task blocks are highlighted in light blue (SP), pink (MP), and green (DP).

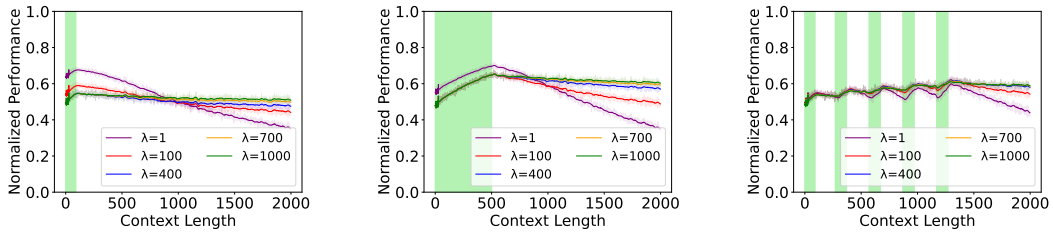
Table 6: The comparisons between the true normalized performance of ICCL and GBCL methods and their fitted ACT-R curves.

Model	DEEPSEEK-R1	LLAMA3-8B	RWKV-7	MAMBA	SGD	ER	EWC
Pearson correlation	0.5152	0.7068	0.6171	0.6215	0.0339	0.5660	0.7253

A.7 Optimal Parameter λ in EWC

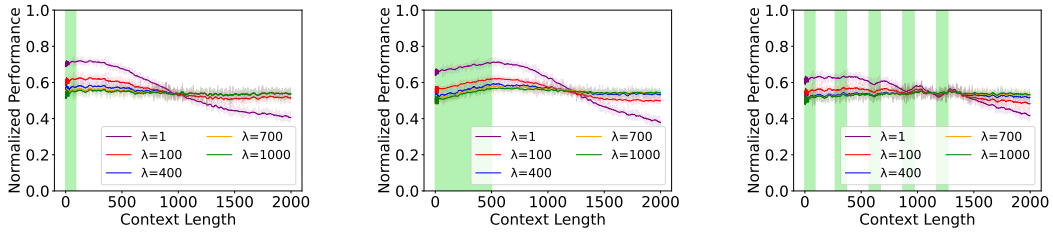
Figure 8 and Figure 9 present the performance of EWC under simple and complex tasks with different regularization strengths λ_{EWC} across SP, MP, and DP scheduling strategies. The results highlight the trade-off between plasticity and stability inherent in EWC. With small λ_{EWC} values (e.g., $\lambda_{\text{EWC}} = 100$), the model maintains higher plasticity and adapts quickly to new demonstrations; however, this comes at the cost of poor stability, as retention performance decays sharply with increasing context length. Conversely, large λ_{EWC} values (e.g., $\lambda_{\text{EWC}} = 1000$) enforce stronger stability but severely limit plasticity, leading to under-adaptation and overall lower performance across all scheduling strategies.

Intermediate values achieve a better balance. In particular, $\lambda_{\text{EWC}} = 700$ consistently provides a favorable trade-off, delivering higher long-term retention compared to smaller λ_{EWC} while avoiding the rigidity observed at $\lambda_{\text{EWC}} = 1000$. This effect is especially evident under MP and DP, where $\lambda_{\text{EWC}} = 700$ maintains relatively stable retention curves while smaller λ_{EWC} values degrade more rapidly. Based on these observations, we select $\lambda_{\text{EWC}} = 700$ as the default regularization strength for subsequent experiments, as it offers the most robust balance between plasticity and stability across both simple and complex tasks



(a) Performances under SP (b) Performances under MP (c) Performances under DP

Figure 8: EWC's retention performances under Markov chains with simple task (markov chain with $Ns = 4$).



(a) Performances under SP (b) Performances under MP (c) Performances under DP

Figure 9: EWC's retention performances under Markov chains with complex task (markov chain with $Ns = 8$).

A.8 The Fitted ACT-R model results from ICCL and GBCL

To further analyze the retention dynamics of ICCL and GBCL methods, we leveraged the experimental data of normalized retention performance across prompt sequence positions under the DP setting. We

applied a modified ACT-R model to fit the performance trajectories of each method. The comparisons between the observed normalized retention performance curves and the corresponding fitted ACT-R curves on both simple and complex tasks are illustrated in Figures 10–16.

In addition, we computed the average Pearson correlation coefficients between the true normalized retention performance curves and their fitted ACT-R counterparts, as reported in Table 6. The results show that, with the exception of SGD, all ICCL and GBCL methods achieve relatively high correlation coefficients (e.g., 0.71 for LLaMA3-8B and 0.73 for EWC). This indicates that the fitted ACT-R model provides a reliable description of the dynamics of normalized retention performance, capturing how ICCL and GBCL retention evolves with increasing sequence positions.

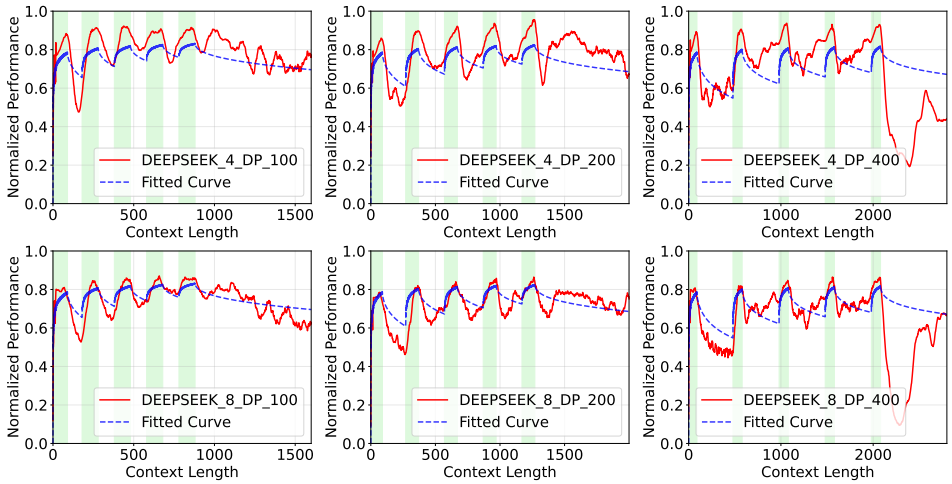


Figure 10: The performance comparisons between DEEPSEEK-R1 and its fitted ACT-R model.

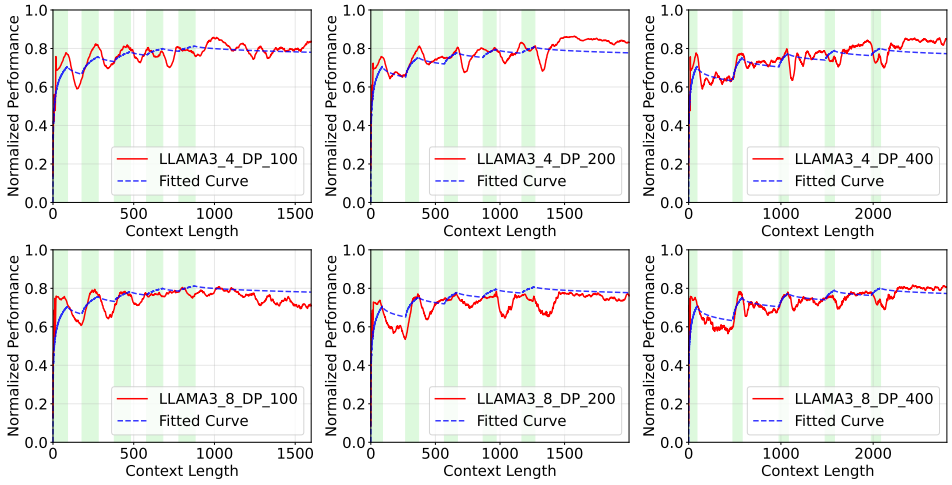


Figure 11: The performance comparisons between LLAMA3-8B and its fitted ACT-R model.

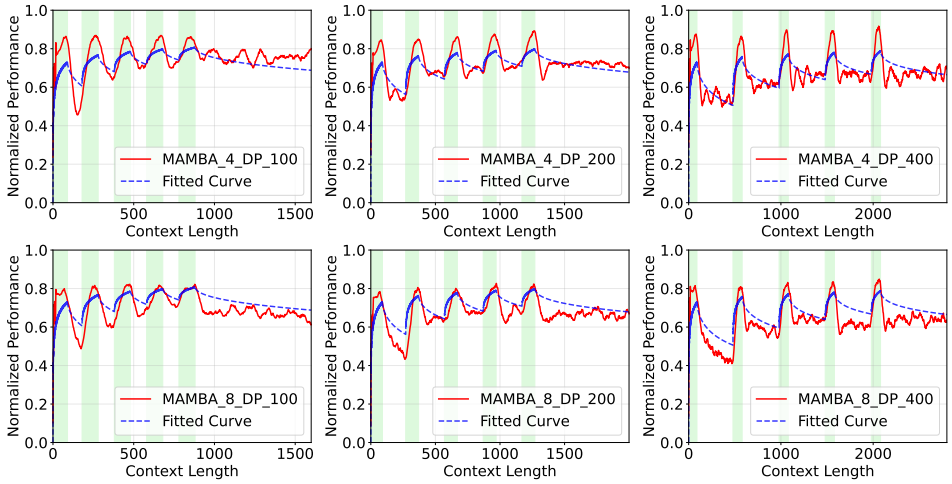


Figure 12: The performance comparisons between MAMBA and its fitted ACT-R model.

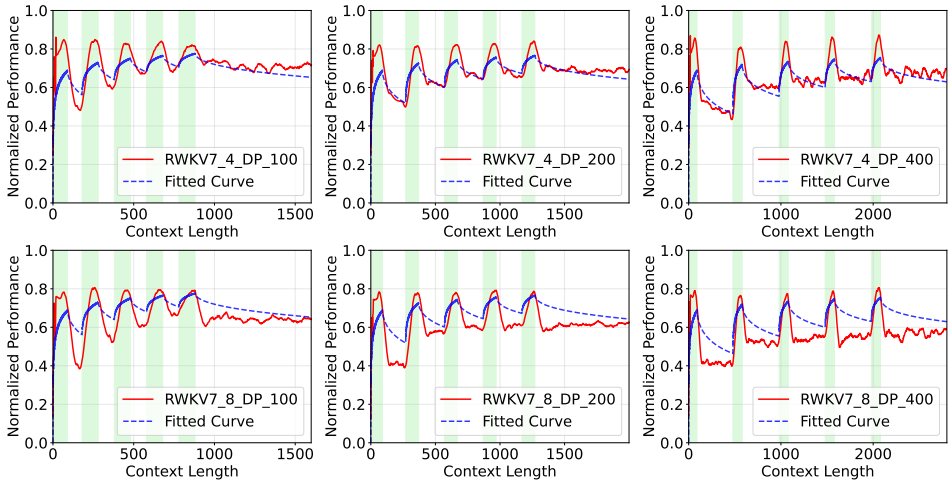


Figure 13: The performance comparisons between RWKV-7 and its fitted ACT-R model.

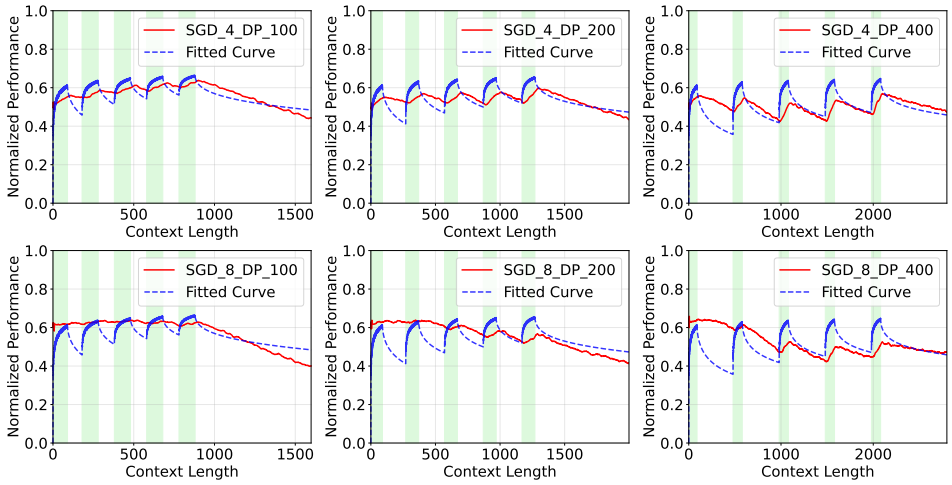


Figure 14: The performance comparisons between SGD and its fitted ACT-R model.

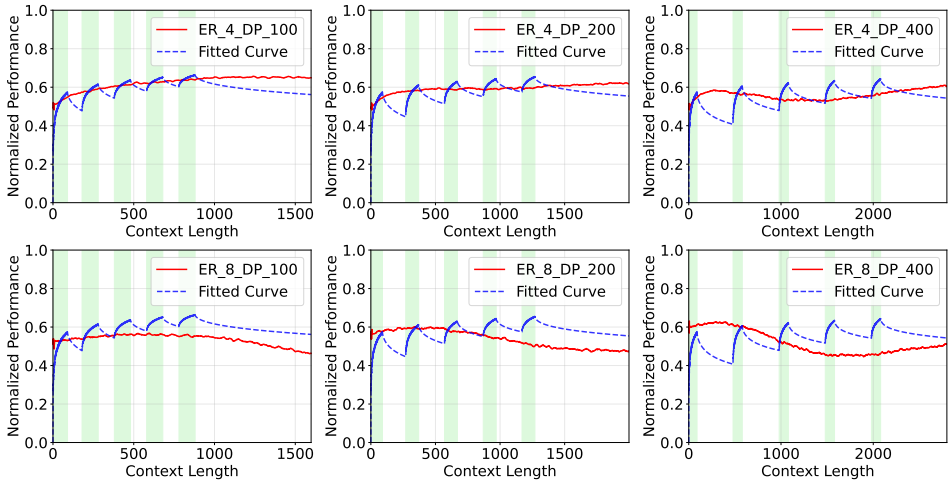


Figure 15: The performance comparisons between ER and its fitted ACT-R model.

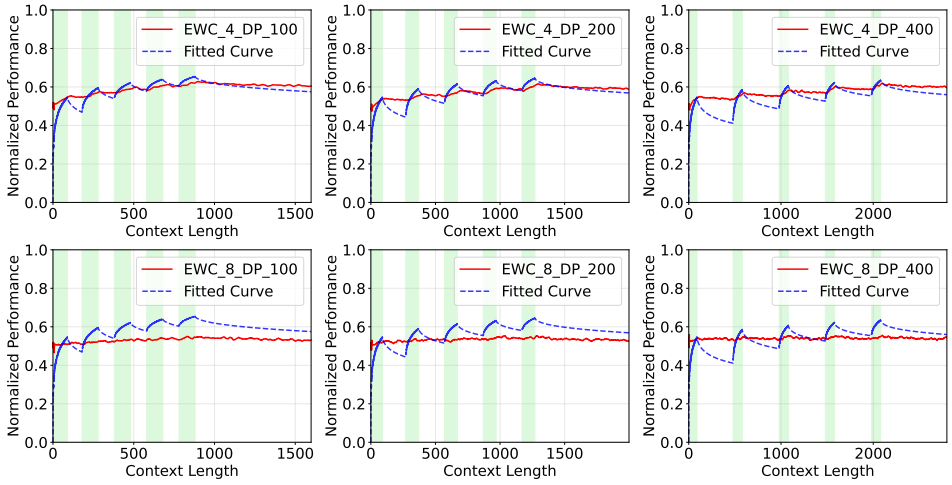


Figure 16: The performance comparisons between EWC and its fitted ACT-R model.