FOREVER: Forgetting Curve-Inspired Memory Replay for Language Model Continual Learning

Yujie Feng[1*], **Hao Wang**[2*], **Jian Li**[3], **Xu Chu**[4], **Zhaolu Kang**[4]
Yiran Liu[5], **Yasha Wang**[4], **Philip S. Yu**[6†], **Xiao-Ming Wu**[1†]
[1]The Hong Kong Polytechnic University [2]ShineMo Ltd., China
[3]Solar System of OVB, Tencent, China [4]Peking University
[5]University College London [6]University of Illinois Chicago
yujie.feng@connect.polyu.hk, xiao-ming.wu@polyu.edu.hk

Abstract

Continual learning (CL) for large language models (LLMs) aims to enable sequential knowledge acquisition without catastrophic forgetting. Memory replay methods are widely used for their practicality and effectiveness, but most rely on fixed, step-based heuristics that often misalign with the model's actual learning progress, since identical training steps can result in varying degrees of parameter change. Motivated by recent findings that LLM forgetting mirrors the Ebbinghaus human forgetting curve, we propose **FOREVER** (**FORgEtting curVe-inspired mEmory Replay**), a novel CL framework that aligns replay schedules with a model-centric notion of time. FOREVER defines model time using the magnitude of optimizer updates, allowing forgetting curve-inspired replay intervals to align with the model's internal evolution rather than raw training steps. Building on this approach, FOREVER incorporates a forgetting curve-based replay scheduler to determine *when* to replay and an intensity-aware regularization mechanism to adaptively control *how* to replay. Extensive experiments on three CL benchmarks and models ranging from 0.6B to 13B parameters demonstrate that FOREVER consistently mitigates catastrophic forgetting[1].

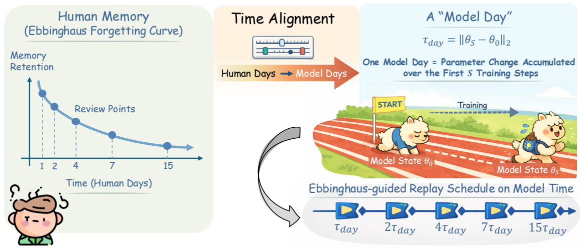


Figure 1: **Aligning human time and model time in FOREVER.** FOREVER aligns Ebbinghaus-inspired human replay intervals with a model-centric timeline defined by accumulated parameter update magnitude, enabling replay to be triggered based on the model's actual learning progress.

1 Introduction

Enabling large language models (LLMs) with continual learning (CL) capabilities is increasingly important in dynamic environments, where models must acquire new knowledge sequentially without costly retraining over all historical data (Yu et al., 2024; Chang et al., 2024; Eskandar et al., 2025). Despite its importance, effective CL for LLMs remains challenging: sequential updates induce distribution drift and interfere with previously learned representations, disrupting the stability–plasticity trade-off and resulting in catastrophic forgetting (CF) (McCloskey and Cohen, 1989).

Replay-based CL methods (Lu et al., 2025; Wan et al., 2025a; Chen and Zeng, 2025; Bai et al., 2025), which revisit a small buffer of past examples while training on new tasks, have emerged as a widely adopted and empirically effective strategy for mitigating forgetting. However, their effectiveness critically depends on two fundamental design questions: (i) **when to replay**, i.e., at which stages of training replay should be activated, and (ii) **how to replay**, i.e., how strongly past knowledge should be reinforced relative to current learning. Existing approaches typically rely on hand-crafted replay heuristics, such as uniform replay intervals or static replay weights, which remain largely decoupled from the model's actual learning dynamics.

Recent studies indicate that neural models exhibit forgetting behaviors similar to those observed in humans (Wu et al., 2025a; Kline, 2025; Zhang et al., 2025b; Wu et al., 2025b), motivating the application of classical cognitive theories—such as the Ebbinghaus forgetting curve (Murre and Dros, 2015)—to LLMs. The Ebbinghaus forgetting curve describes rapid memory loss soon after learning,

[*] Equal contribution.
[†] Corresponding author.
[1]The source code is available at `https://github.com/WoodScene/FOREVER`

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 24945–24965
July 2-7, 2026

followed by a slower rate of decay. Drawing on this analogy, recent work has introduced Ebbinghaus-style spaced replay schedules, where replay occurs more frequently shortly after learning and becomes increasingly spaced over time (e.g., 1, 2, 4, 7, 15 days in humans), to better match the temporal dynamics of memory decay (Zhong et al., 2024; Chen et al., 2025a; Kang et al., 2025a).

However, these approaches typically implement replay schedules based on training steps, implicitly equating step count with the passage of human time. This assumption is problematic, as the same number of training steps can result in varying degrees of model change depending on optimization settings such as learning rate or batch size. As a result, step-based replay schedules may trigger replay at inconsistent model states, leading to a misalignment between human time and model time.

This raises a central question:

How can "human days" be aligned with "model days" to enable more reliable replay scheduling?

To address this question, we propose a novel continual learning framework, **FOREVER** (**FOR**g**E**tting cur**V**e-inspired m**E**mory **R**eplay). FOREVER redefines replay scheduling by replacing step-based time with a model-centric notion of time, measured by **parameter update magnitude**. Unlike training steps, which are external and do not reflect the model's internal changes, parameter update magnitude directly quantifies the extent of model evolution during learning. By tracking this signal, FOREVER aligns human-inspired replay schedules with the model's actual update dynamics, calibrating Ebbinghaus-style intervals to the model's learning progress. This ensures that replay is triggered based on true model advancement rather than arbitrary iteration counts.

FOREVER consists of two integrated components: an **forgetting curve-inspired replay scheduler** and an **intensity-aware replay regularizer**. The scheduler determines **when to replay** by using accumulated parameter update magnitude to define a model-specific "day," triggering replay at Ebbinghaus-style intervals along this axis. This approach synchronizes replay events with comparable stages of model evolution, effectively mapping the forgetting curve onto the model's learning timeline.

To determine **how to replay**, FOREVER monitors recent update intensity and applies a replay-time regularization term with adaptively adjusted strength. Stronger regularization is used during periods of rapid change to stabilize learning, while gentler regularization is applied as updates slow, supporting new task adaptation. Both replay timing and regularization strength are derived from model update dynamics: accumulated updates indicate progress, and recent intensity reflects the rate of change. This unified approach provides a coherent strategy for both *when* and *how* to replay.

Extensive experiments on diverse CL benchmarks demonstrate that FOREVER consistently improves knowledge retention while maintaining strong adaptation to new tasks.

Our main contributions are:

- We introduce FOREVER, a novel CL framework that bridges cognitive forgetting theory and model training dynamics by defining time via parameter update magnitude, enabling forgetting curve-inspired replay scheduling beyond step-based heuristics.
- We develop two complementary techniques: an forgetting curve-inspired memory replay scheduler for determining *when* to replay, and an intensity-aware replay regularizer for adaptively controlling *how* to replay.
- Extensive evaluation on three CL benchmarks and four model backbones (0.6B to 13B parameters) demonstrates the effectiveness of FOREVER in mitigating catastrophic forgetting.

2 Proposed Method: FOREVER

Problem Formulation Continual learning aims to progressively accumulate knowledge from a sequence of tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$ over time. Each task $\mathcal{T}_k$ is associated with a dataset $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ of size N_k, where $x_i^k \in \mathcal{X}_k$ denotes the input and $y_i^k \in \mathcal{Y}_k$ denotes the corresponding target.

A model parameterized by Θ is trained on these tasks sequentially, without access to the full data from previous tasks once they have been observed. The objective is to minimize the expected negative log-likelihood over the data encountered so far:

$$\mathcal{L} = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\leq k}} \left[-\log p_\Theta(y \mid x) \right], \qquad (1)$$

Following common practice in replay-based CL, we assume access to a memory buffer that stores a limited number of samples from previous tasks. For each past task $\mathcal{T}_i$ $(i < k)$, at most $|\mathcal{M}|$ examples are retained in a task-specific buffer $\mathcal{M}_i$, yielding a global memory $\mathcal{M}_{<k} = \bigcup_{i<k} \mathcal{M}_i$. When learning task $\mathcal{T}_k$, the model is optimized jointly on the current task data $\mathcal{D}_k$ and the replay memory $\mathcal{M}_{<k}$.

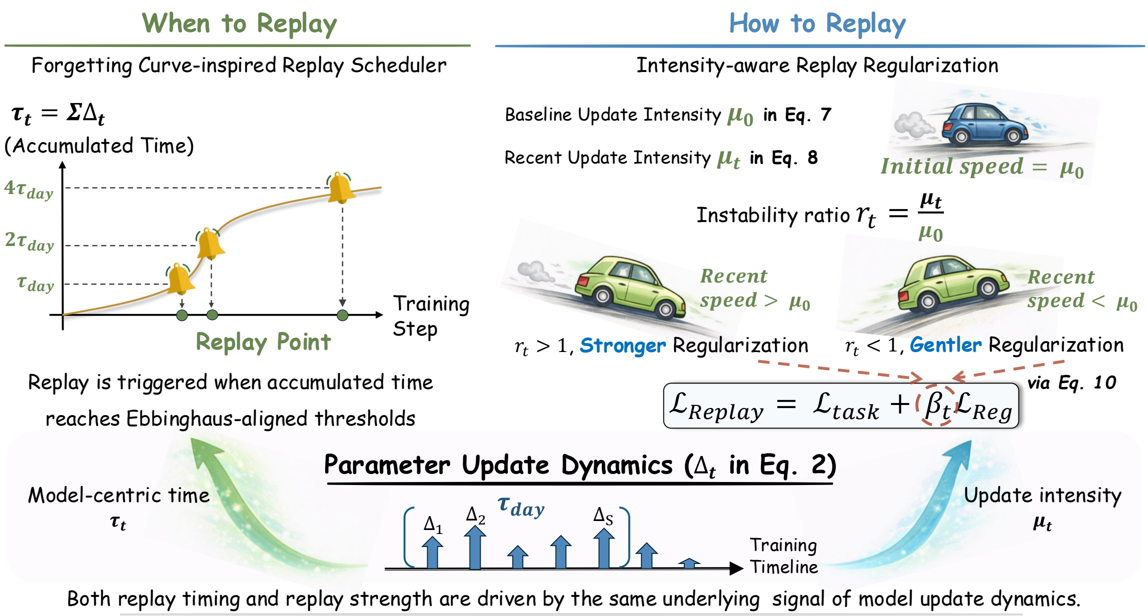


Figure 2: **Overview of FOREVER.** FOREVER decomposes replay into two coupled decisions—*when to replay* and *how to replay*—both grounded in model update dynamics. Parameter update magnitudes Δ_t track model evolution over training steps, whose accumulation defines a model-centric notion of time (virtual "model days"). **When to replay** (Left): accumulated model time τ_t measures how far the model has progressed in parameter space and triggers replay when Ebbinghaus-guided time thresholds are reached. **How to replay** (Right): recent update intensity μ_t, relative to a baseline μ_0, modulates replay regularization strength—stronger under rapid model changes and gentler when updates are stable. By unifying replay timing and replay strength under the same update-dynamics signal, FOREVER enables a coherent and model-centric replay strategy.

Overview As illustrated in Figure 2, FOREVER decomposes replay into two tightly coupled components: (i) an ***Forgetting Curve-inspired Replay Scheduler***, which determines *when to replay* by first calibrating a model-centric notion of time via parameter update dynamics and then mapping human Ebbinghaus intervals onto this model-time axis to trigger replay at the corresponding thresholds; and (ii) an ***Intensity-aware Replay Regularization*** mechanism, which regulates *how to replay* by adaptively adjusting replay strength according to recent update intensity. Together, these components unify replay timing and replay strength under a single dynamics-aware framework, grounding replay decisions in how much and how fast the model evolves, rather than in fixed step-based heuristics.

2.1 When to Replay: Forgetting Curve-inspired Replay Scheduler

Model-centric Time Calibration. A key challenge in applying Ebbinghaus-style replay schedules to CL lies in defining a meaningful notion of time for LLMs. While human memory decay is naturally measured in days, training steps are a poor proxy for model progress, as their effects vary substantially with optimization settings such as batch sizes. To address this mismatch, we introduce a model-centric notion of time grounded in parameter update dynamics, which enables the calibration of *virtual Ebbinghaus days* for LLMs.

Parameter Update Dynamics. Let $\Theta_t \in \mathbb{R}^N$ denote the model parameters after the t-th training update, where N is the total number of parameters. We quantify the magnitude of model change at step t using the parameter update norm

$$\Delta_t = \|\Theta_t - \Theta_{t-1}\|_2, \tag{2}$$

which measures how much the model evolves during a single optimization step. In practice, Δ_t is computed over trainable parameters only (i.e., the LoRA (Hu et al., 2021) weights) and can be directly obtained from the optimizer's applied parameter updates, without requiring additional forward or backward passes. Accumulating these updates yields a model-centric notion of time,

$$\tau_t = \sum_{i=1}^{t} \Delta_i, \tag{3}$$

which represents the total distance the model has traveled in parameter space. In our implementation, τ_t is reset at the beginning of each new task and measures the elapsed model-centric time since learning on the current task started. Unlike raw step counts, τ_t directly reflects model evolution and is substantially less sensitive to optimization hyperparameters.

Calibrating a Virtual Model Day. Based on the accumulated update magnitude, we define a virtual model "day" as the amount of parameter change observed over an initial window of S training steps:

$$\tau_{\text{day}} = \sum_{i=1}^{S} \Delta_i. \tag{4}$$

This quantity serves as a model-specific unit of time, anchoring human-defined temporal intervals to the model's own learning dynamics.

Virtual Ebbinghaus Days. Given a sequence of Ebbinghaus-style replay intervals expressed in human days, $\mathcal{D}_{\text{human}} = \{d_1, d_2, \ldots\}$ (e.g., $\{1, 2, 4, 7, 15, 30, \ldots\}$), we map them onto the model-centric time axis as

$$\mathcal{D}_{\text{model}} = \{d \cdot \tau_{\text{day}} \mid d \in \mathcal{D}_{\text{human}}\}. \tag{5}$$

Each element in $\mathcal{D}_{\text{model}}$ specifies a target amount of accumulated model change at which replay is triggered, aligning human-inspired replay intervals with comparable stages of model evolution.

The classical Ebbinghaus forgetting curve describes rapid memory decay at early stages, followed by slower decay, motivating replay schedules with increasing intervals. Leveraging the calibrated model-centric notion of time and replay thresholds $\mathcal{D}_{\text{model}}$, we describe how FOREVER determines *when* replay should be triggered.

Replay Triggering Criterion. During training, we continuously track the accumulated model-centric time τ_t. Replay is triggered whenever τ_t reaches the next threshold in $\mathcal{D}_{\text{model}}$. Formally, let j denote the index of the next scheduled replay. A replay event is initiated at training step t if

$$\tau_t \geq \mathcal{D}_{\text{model}}^{(j)}. \tag{6}$$

After replay is performed, the scheduler advances to the subsequent threshold, and training resumes on the current task.

2.2 How to Replay: Intensity-aware Replay Regularization

Once replay is triggered, FOREVER determines how strongly replay should be applied. The key idea is to adapt replay strength to the model's current training dynamics: replay should impose stronger constraints when the model is changing rapidly and remain gentle once learning stabilizes. To this end, FOREVER modulates replay regularization based on the ratio between recent and baseline update intensity over training.

Update Intensity and Instability Ratio. Using the same warm-up window of S steps, we define the baseline update intensity as

$$\mu_0 = \frac{1}{S} \sum_{t=1}^{S} \Delta_t, \tag{7}$$

which provides a reference scale for subsequent training dynamics.

Thereafter, the recent update intensity is tracked using an exponential moving average,

$$\mu_t = (1 - \lambda)\mu_{t-1} + \lambda \Delta_t, \tag{8}$$

where λ controls the degree of temporal smoothing. We then compute an instability ratio

$$r_t = \frac{\mu_t}{\mu_0}, \tag{9}$$

which indicates whether the model is currently updating more aggressively ($r_t > 1$) or more conservatively ($r_t < 1$) than at the beginning of training.

The replay regularization strength is scaled according to the instability ratio r_t:

$$\beta_t = \beta_{\text{base}} \cdot \text{clip}(1 + \gamma(r_t - 1),\ g_{\min},\ g_{\max}), \tag{10}$$

where β_{base} balances the magnitude between the replay regularization loss and the task loss, γ controls sensitivity to update intensity, and $g_{\min}$ and $g_{\max}$ denote the lower and upper bounds of the clipping operation for numerical stability.

Replay Objective. At replay time, the model is optimized on samples from the memory buffer while applying a parameter-level regularization anchored at a reference snapshot $\Theta^{\star}$, taken at the end of the previous task. The replay loss is defined as

$$\mathcal{L}_{\text{replay}} = \mathcal{L}_{\text{task}}^{(\text{old})} + \beta_t \sum_j \|\Theta_j - \Theta_j^{\star}\|_2^2, \tag{11}$$

where θ_j denotes an individual model parameter.

By scaling replay regularization with the instability ratio r_t, FOREVER applies stronger constraints when the model undergoes rapid changes and relaxes replay as learning stabilizes. This design enables adaptive replay strength driven solely by parameter update dynamics, without relying on fixed replay weights or hand-crafted schedules.

Detailed implementation of FOREVER algorithm is provided in the Appendix (Algorithm 1).

3 Experiments and Analysis

Datasets Following the experimental protocol of Du et al. (2024), we evaluate on three representative CL benchmarks for NLP: (i) the **Standard CL Benchmark**, which comprises five text classification tasks from Zhang et al. (2015); (ii) the **Long Sequence Benchmark** (Razdaibiedina et al., 2023), a more challenging setting consisting of 15 sequential tasks designed to assess long-horizon knowledge accumulation and retention; and (iii) the **SuperNI Benchmark** (Wang et al., 2022), a large-scale instruction-following benchmark containing 15 diverse NLP generation tasks. Following Wang et al. (2023), we sample 1000 training instances per task and reserve 500 instances per class for evaluation. For each benchmark, we evaluate multiple task orders, with detailed dataset statistics and task sequences provided in Appendix G.

Metrics Let $a_{i,j}$ denote the testing performance on task $\mathcal{T}_i$ after training on task $\mathcal{T}_j$, and let K denote the total number of tasks. We evaluate the overall performance (OP) (Chaudhry et al., 2018) and backward transfer (BWT) (Ke and Liu, 2022) after training on the final task:

$$\textbf{OP} = \frac{1}{K}\sum_{i=1}^{K} a_{i,K} \tag{12}$$

$$\textbf{BWT} = \frac{1}{K-1}\sum_{i=1}^{K-1} (a_{i,K} - a_{i,i}) \tag{13}$$

Baselines We compare FOREVER with a comprehensive set of CL baselines, with a particular emphasis on replay-based methods that adopt different replay scheduling strategies. For fairness, all methods are implemented under the same LoRA-based framework and trained for the same number of epochs. The evaluated baselines include MixReplay, Fixed-interval Replay, EWC (Kirkpatrick et al., 2017), O-LoRA (Wang et al., 2023), MoELoRA (Luo et al., 2024), SAPT (Zhao et al., 2024), MIGU (Du et al., 2024), SSR (Huang et al., 2024), Recurrent-KIF (Feng et al., 2025b), AIM-Merging (Feng et al., 2025a), and VBM (Kang et al., 2025a). Finally, multi-task learning (MTL), which jointly trains on all tasks, serves as an upper-bound reference. Detailed descriptions of all baselines are provided in Appendix F.

Training Details We evaluate FOREVER across multiple backbone models, including Qwen3-0.6B and Qwen3-4B (Yang et al., 2025), as well as LLaMA3.1-8B (Touvron et al., 2023) and LLaMA2-13B. Following Feng et al. (2025b), we store 2% of the original training data from each task in a memory buffer for replay. For FOREVER, the warm-up window size S is set to 24, which is used to calibrate both the virtual model day and the baseline update intensity. The smoothing coefficient λ in Eq. (8) is fixed to 0.05. The base regularization coefficient β_{base} in Eq. (10) is set to 10^{-3}, with clipping bounds $g_{\min} = 0.5$ and $g_{\max} = 3.0$. All experiments are averaged over three independent runs. More details are provided in Appendix H.

3.1 Main Results

The overall CL results using the same Qwen3-0.6B backbone are summarized in Table 1.

FOREVER Effectively Mitigates the Challenge of Catastrophic Forgetting. FOREVER consistently outperforms representative CL baselines, including regularization-based methods (e.g., EWC) and replay-based approaches (e.g., SSR and Recurrent-KIF). In particular, FOREVER achieves a notable improvement in OP, increasing the average OP from 58.7% to 61.5% compared to the strongest replay-based baseline, SSR. Meanwhile, FOREVER exhibits stronger backward transfer, improving BWT from -4.9% to -4.2% relative to AIM-Merging. These results indicate that FOREVER more effectively preserves knowledge from earlier tasks while maintaining competitive performance on newly learned tasks.

Moreover, FOREVER consistently surpasses the Ebbinghaus-inspired VBM baseline, yielding gains of 1.2% in OP and 0.9% in BWT. This demonstrates the advantage of aligning replay decisions with model-centric learning dynamics, enabling more robust knowledge retention than step-based replay or static regularization strategies.

Method	Memory-Based	Standard CL		Long Sequence		SuperNI	
		OP↑	BWT↑	OP↑	BWT↑	OP↑	BWT↑
Fine-tuning	✗	47.2	-12.6	36.0	-17.5	8.2	-27.4
EWC (Kirkpatrick et al., 2017)	✗	51.0	-10.3	44.8	-13.8	32.9	-18.6
O-LoRA (Wang et al., 2023)	✗	59.4	-7.9	54.1	-12.4	23.7	-17.5
MoELoRA (Luo et al., 2024)	✗	55.3	-8.2	35.3	-13.6	25.7	-11.3
MixReplay	✓	65.8	-8.0	65.1	-11.4	34.6	-14.1
Fixed-interval Replay	✓	65.1	-9.2	64.5	-10.9	34.7	-14.5
SAPT (Zhao et al., 2024)	✓	68.8	-6.9	67.2	-8.8	38.5	-6.2
MIGU (Du et al., 2024)	✓	69.9	-7.5	66.3	-9.2	35.0	-8.1
SSR (Huang et al., 2024)	✓	68.4	-7.1	67.5	-9.0	40.1	-5.4
Recurrent-KIF (Feng et al., 2025b)	✓	70.6	-6.5	67.7	-8.4	39.7	-5.0
AIMMerging (Feng et al., 2025a)	✓	71.9	-5.0	67.9	-6.3	41.0	-3.4
VBM (Kang et al., 2025a)	✓	71.5	-5.2	68.1	-6.1	41.3	-3.7
FOREVER (ours)	✓	**72.9**	**-4.7**	**69.4**	**-5.0**	**42.1**	**-2.9**
MTL		77.4	-	77.8	-	48.2	-

Table 1: Overall CL results on three benchmarks using the Qwen3-0.6B backbone. We report Overall Performance (OP) and Backward Transfer (BWT) after training on the final task. All results are averaged over different task orders. The last row corresponds to the multi-task learning (MTL) upper bound. Our method, FOREVER, outperforms the previous best method, VBM, with an average improvement of 1.2% in OP and a 0.9% increase in BWT.

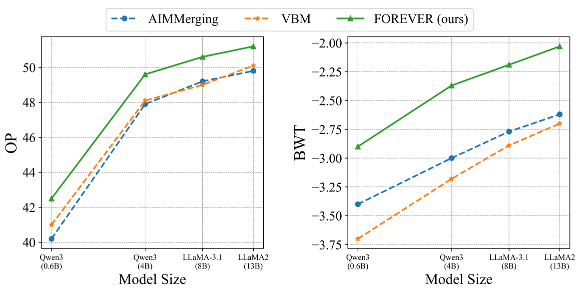


Figure 3: Performance of FOREVER with different backbones on the SuperNI Benchmark.

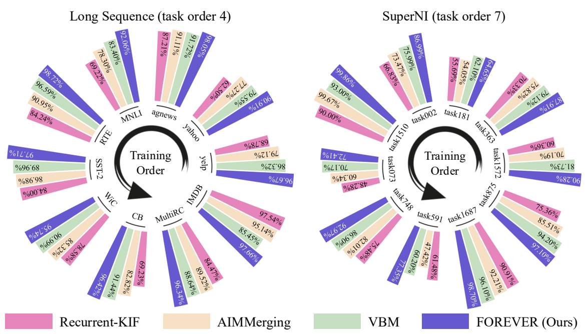


Figure 4: **Impact of Catastrophic Forgetting in Continual Learning.** Performance is reported as a percentage of each task's upper bound. After fine-tuning on the final task, FOREVER shows superior resistance to performance decline on previously learned tasks.

FOREVER Generalizes Consistently Across Model Scales. We further evaluate FOREVER across backbone models ranging from 0.6B to 13B parameters, as shown in Figure 3. Across all model scales, FOREVER consistently outperforms the corresponding baselines. For example, with the LLaMA3.1-8B backbone, FOREVER improves OP from 49.0% to 50.6% and reduces BWT from -2.9% to -2.1% compared to VBM. These results demonstrate that the benefits of model-centric replay scheduling generalize across backbone sizes.

Forgetting Curve-inspired Replay Enables Knowledge Retention. Figure 4 presents a task-wise comparison after training on the final task. FOREVER consistently retains higher performance on previously learned tasks than all baselines. Notably, on RTE and task1687, FOREVER achieves performance comparable to the multi-task learning upper bound, indicating strong resistance to catastrophic forgetting. Overall, these results suggest that FOREVER balances prior knowledge retention and new task adaptation in continual learning.

3.2 Ablation Study

We conduct ablation studies to analyze the contribution of individual components in FOREVER. Results for task order 7 on the SuperNI are reported in Table 2. Additional analysis, including time complexity, memory size sensitivity, and hyperparameter robustness, are provided in Appendix B.

3.2.1 Effect of the Forgetting Curve-inspired Replay Scheduler

We first examine how different replay schedules affect CL performance to validate the Ebbinghaus-style design. Specifically, we replace the human-derived schedule $\mathcal{D}_{\text{human}}$ with alternative strategies to construct model-driven replay sequences $\mathcal{D}_{\text{model}}$: (i) *Fixed-interval replay* ("+FIR"), which triggers replay at uniform intervals (e.g., $\{24, 48, 96, \ldots\}$); (ii) *Reversed replay* ("+RR"), which applies replay intervals in descending order (e.g., $\mathcal{D}_{\text{human}} = \{30, 15, 7, 4, 2, 1\}$); and (iii) *End-only replay* ("+ER"), which performs replay only after completing training on the new task.

Across all variants, the forgetting curve-inspired schedule consistently yields the strongest performance. This result indicates that replaying more frequently during early training—when parameter updates are large—and gradually reducing replay frequency as learning stabilizes is critical for mitigating catastrophic forgetting.

3.2.2 Effect of Model-Centric Time Calibration

We next investigate the impact of aligning human time with model-centric time on replay effectiveness. We compare our update-dynamics-based calibration with a *step-based calibration* baseline ("+STC"), which maps human-defined days to fixed training steps.

Model-centric calibration consistently outperforms step-based calibration, yielding an average improvement of 1.2% in OP and 1.1% in BWT. This gap arises because step-based schedules trigger replay at fixed iteration counts across tasks, ignoring task-specific learning dynamics. In contrast, our approach adapts replay timing to the model's intrinsic evolution, enabling more accurate alignment between replay events and learning stages.

3.2.3 Effect of Intensity-aware Replay Regularization

Finally, we evaluate the role of intensity-aware replay regularization in FOREVER. We compare our approach with three variants: (i) removing the regularization term ("−IAR"); (ii) replacing it with parameter-importance regularization following EWC ("+PIR"); and (iii) combining importance with intensity-aware scaling ("+IAR & PIR").

Removing IAR leads to clear performance drops, confirming its necessity. Compared with PIR, FOREVER achieves comparable results while avoiding the additional cost of estimating and storing parameter importance scores. Combining IAR and PIR yields only marginal gains, suggesting that both mechanisms address similar sources of instability during training. Overall, these results demonstrate that parameter update dynamics provide an efficient and effective signal for modulating replay strength.

Category	Method	OP ↑	BWT ↑
Full Model	FOREVER	42.5	-2.8
Replay Scheduler (§3.2.1)	+ FIR	40.1	-5.2
	+ RR	37.2	-7.8
	+ ER	40.9	-6.9
Time Calibration (§3.2.2)	+ STC	41.3	-3.9
Regularization (§3.2.3)	− IAR	39.9	-4.4
	+ PIR	42.7	-3.0
	+ IAR & PIR	42.8	-2.6

Table 2: **Ablation study.** We evaluate the contribution of each component by selectively removing ("−") or replacing ("+") it with alternative designs.

3.3 Visualization of Replay Dynamics

We visualize the replay dynamics induced by the proposed model-centric scheduling mechanism. Specifically, we analyze the step-wise parameter update magnitude Δ_t, the evolution of accumulated model-centric time τ_t, and the resulting replay trigger points throughout training. Figure 5 shows results for task order 7 on the SuperNI benchmark, while visualizations for all eight task orders are provided in Appendix E (Figure 7).

As shown in the left panel, the update magnitude Δ_t exhibits pronounced non-uniformity over training. Large updates occur at the early stages of each task, followed by progressively smaller updates as optimization stabilizes. Correspondingly, the accumulated model-centric time τ_t (right panel) grows rapidly during periods of substantial parameter change and increases more slowly once training enters a stable regime.

Importantly, under FOREVER, replay events are not tied to fixed training steps. Instead, replay is triggered whenever τ_t reaches the next Ebbinghaus-calibrated threshold. As a result, the training steps at which replay occurs vary substantially across tasks. For example, the replay corresponding to the "7-day" threshold spans a wide range of training steps (approximately steps 140–180) across different tasks. This behavior illustrates that replay in

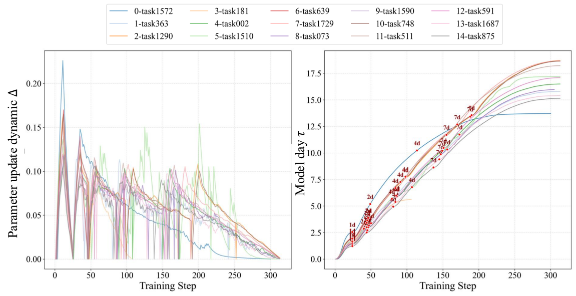


Figure 5: **Visualization of model-centric replay dynamics during training. Left:** step-wise parameter update magnitude Δ_t across training steps. **Right:** accumulated model-centric time τ_t with replay trigger points annotated. Under the proposed model-centric time definition, replay is triggered at different training steps for different tasks, reflecting task-dependent learning dynamics rather than fixed step-based schedules.

FOREVER is governed by the model's intrinsic learning progress rather than by predefined step-based schedules, providing a direct explanation for its improved robustness to catastrophic forgetting.

3.4 Effect of Increasing-Spacing Replay Scheduling

We further analyze the effect of replay interval scheduling in continual learning. The key idea behind our design is not a specific handcrafted interval sequence, but a structural principle: forgetting typically follows a fast-then-slow decay pattern, which motivates a dense-to-sparse (increasing-spacing) replay schedule. In early training stages, frequent reinforcement is required, while later stages allow progressively wider intervals.

To isolate the effect of interval structure, we conduct controlled comparisons under identical replay budgets and cumulative replay data volumes. Our original setting adopts an Ebbinghaus-inspired increasing-spacing schedule as the default configuration. We additionally consider several alternative scheduling strategies:

- **Increasing-spacing**: including multiple parameterizations, such as exponential (geometric progression) and polynomial (square growth);
- **Uniform-spacing**: equal interval replay;
- **Decreasing-spacing**: reversed scheduling (sparse-to-dense).

Results on SuperNI (Qwen3-0.6B, task order 7) are shown in Table 3. We observe that increasing-spacing schedules consistently outperform uniform and decreasing-spacing strategies. This aligns with the intuition that replay frequency should match the evolving forgetting dynamics: frequent replay is needed when forgetting is rapid, while wider intervals suffice when forgetting slows down.

Among different increasing-spacing variants, the standard Ebbinghaus-style instantiation achieves the best performance. Notably, this schedule is not an arbitrary functional choice, but is grounded in long-established empirical studies of human memory. Compared with purely parametric forms (e.g., exponential or polynomial growth), it reflects a more refined spacing pattern.

Overall, these results suggest that performance gains stem from alignment with the underlying forgetting behavior, rather than any specific handcrafted interval sequence.

4 Related Work

4.1 Continual Learning for LLMs

Continual learning (CL) aims to enable models to acquire new knowledge while mitigating catastrophic forgetting of previously learned information (Ke et al., 2021; Zhou et al., 2024; Jiang et al., 2025). Extending CL to LLMs has recently gained traction, as LLMs are expected to adapt to evolving tasks without full retraining (Pham et al., 2023; Zeng et al., 2025b; Wang et al., 2025b).

Existing CL approaches for LLMs fall into three major categories: (1) Regularization-based methods constrain parameter updates based on estimated importance scores (Li et al., 2024; Zhang et al., 2025a; Cheng et al., 2025; Momeni et al., 2025), but their scalability is challenged by the prohibitive cost of importance estimation in LLMs (Tang et al., 2025; Ong et al., 2025). (2) Architecture-based methods allocate task-specific components such as adapters, orthogonal subspaces, or MoE modules (Ke et al., 2023; Wan et al., 2024; He et al., 2024; Wang et al., 2024; Feng et al., 2024b), which reduce interference but often introduce additional memory or require task identifiers, limiting scalability. (3) Replay-based methods revisit selected samples from previous tasks (Huang et al., 2024; Alexandrov et al., 2024), and have shown strong empirical performance when combined with PEFT techniques like LoRA (Huang et al., 2025).

However, most replay-based methods rely on heuristic schedules—such as fixed intervals or mixing ratios—that are decoupled from the model's learning dynamics (Wan et al., 2025b). This raises a central question: *when and how should replay be triggered to align with the model's internal state?*

Replay Strategy	D_{human}	OP ↑	BWT ↑
Increasing-spacing (standard)	$\{1, 2, 4, 7, 15, \dots\}$	42.5	-2.8
Increasing-spacing (exponential)	$\{1, 2, 4, 8, 16, \dots\}$	42.3	-2.6
Increasing-spacing (polynomial)	$\{1, 4, 9, 16, \dots\}$	41.5	-3.2
Uniform-spacing	$\{2, 4, 6, 8, 10, \dots\}$	40.9	-3.6
Decreasing-spacing	$\{15, 7, 4, 2, 1\}$	37.2	-7.8

Table 3: Comparison of different replay interval schedules under the same replay budget.

Our work addresses this gap by grounding replay decisions in the model's intrinsic learning dynamics, moving beyond step-based or static schedules.

4.2 Forgetting Dynamics in LLMs

The forgetting behavior of LLMs has recently gained attention from perspectives such as representation drift, knowledge loss, and long-term memory degradation (Jiang et al., 2024; Dou et al., 2024; Ren et al., 2024; Li et al., 2025; Liu et al., 2025). Beyond CL, forgetting is now studied as a general property of LLMs under sequential or prolonged training (Zeng et al., 2025a; Wang et al., 2025a).

Recent empirical studies show that forgetting in LLMs exhibits structured temporal patterns resembling human memory decay, with rapid early performance drops followed by slower degradation (Wu et al., 2025a; Kline, 2025). This has motivated the adoption of cognitive principles such as the Ebbinghaus forgetting curve in CL, where replay intervals increase over training (Deng et al., 2025). However, existing approaches typically measure "time" using training steps, implicitly assuming that iteration counts reflect comparable model changes—an assumption that often breaks down under the highly variable optimization dynamics of LLMs. In contrast, our work grounds Ebbinghaus-style replay in a model-centric notion of time defined by parameter update dynamics, enabling replay decisions aligned with the model's intrinsic learning process.

5 Conclusion

In this paper, we propose FOREVER, a replay-based continual learning framework for large language models that aligns human-inspired replay schedules with a model-centric notion of time. By grounding replay decisions in parameter update dynamics, FOREVER jointly determines when to replay and how strongly to regularize past knowledge. Extensive experiments across multiple benchmarks and model scales show that FOREVER effectively mitigates catastrophic forgetting and consistently outperforms existing continual learning methods.

Limitations

We acknowledge two primary limitations of our work, each of which highlights promising avenues for future research.

First, FOREVER utilizes parameter update dynamics as an indirect proxy for model evolution and forgetting. While metrics such as accumulated update magnitude and recent update intensity offer a principled, model-centric notion of time, they do not directly reflect task-level performance degradation or semantic forgetting. Incorporating more explicit indicators of forgetting—such as performance-based metrics or task-specific diagnostics—could further enhance the effectiveness of replay scheduling.

Second, FOREVER employs a predefined Ebbinghaus-inspired replay interval pattern, drawing from cognitive theory. Although this approach provides an interpretable and effective prior for replay, it may not be universally optimal across diverse tasks or training settings. Learning replay interval patterns from data, or dynamically adapting them to specific task characteristics, represents a valuable direction for extending beyond fixed, human-inspired schedules.

Acknowledgments

The authors thank the anonymous reviewers for their helpful feedback. This work was partially supported by the Hong Kong General Research Fund (15204225), and Project P0056021 (Parent Project P0050643) of the Otto Poon Charitable Foundation Smart Cities Research Institute, The Hong Kong Polytechnic University.

References

Anton Alexandrov, Veselin Raychev, Mark Mueller, Ce Zhang, Martin Vechev, and Kristina Toutanova. 2024. Mitigating catastrophic forgetting in language transfer via model merging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 17167–17186.

Andrew Bai, Chih-Kuan Yeh, Cho-Jui Hsieh, and Ankur Taly. 2025. An efficient rehearsal scheme for catastrophic forgetting mitigation during multi-stage fine-tuning. *Preprint*, arXiv:2402.08096.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547.

Hongqi Chen, Zhiyong Feng, Shizhan Chen, Hongyue Wu, Yingchao Sun, Jingyu Li, Qinghang Gao, Lu Zhang, and Xiao Xue. 2025a. Incorporating forgetting curve and memory replay for evolving socially-aware recommendation. *Information Processing & Management*, 62(3):104070.

Shengyuan Chen, Qinggang Zhang, Junnan Dong, Wen Hua, Qing Li, and Xiao Huang. 2024. Entity alignment with noisy annotations from large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. 2025b. You don't need pre-built graphs for RAG: Retrieval augmented generation with adaptive reasoning structures. In *The Fortieth AAAI Conference on Artificial Intelligence*.

Xi Chen and Min Zeng. 2025. Prototype conditioned generative replay for continual learning in nlp. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12754–12770.

Quan Cheng, Yuanyu Wan, Lingyu Wu, Chenping Hou, and Lijun Zhang. 2025. Continuous subspace optimization for continual learning. *arXiv preprint arXiv:2505.11816*.

Xu Chu, Xinke Jiang, Rihong Qiu, Jiaran Gao, and Junfeng Zhao. 2025a. Model shapley: Find your ideal parameter player via one gradient backpropagation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xu Chu, Zhixin Zhang, Tianyu Jia, and Yujie Jin. 2025b. Stackelberg self-annotation: A robust approach to data-efficient llm alignment. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Junze Deng, Qinhang Wu, Peizhong Ju, Sen Lin, Yingbin Liang, and Ness Shroff. 2025. Unlocking the power of rehearsal in continual learning: A theoretical perspective. *Preprint*, arXiv:2506.00205.

Xiaoyu Dong, Yujie Feng, Zexin Lu, Guangyuan Shi, and Xiao-Ming Wu. 2024. Zero-shot cross-domain dialogue state tracking via context-aware auto-prompting and instruction-following contrastive decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8527–8540.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945, Bangkok, Thailand. Association for Computational Linguistics.

Wenyu Du, Shuang Cheng, Tongxu Luo, Zihan Qiu, Zeyu Huang, Ka Chun Cheung, Reynold Cheng, and Jie Fu. 2024. Unlocking continual learning abilities in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6503–6522.

Masih Eskandar, Tooba Imtiaz, Davin Hill, Zifeng Wang, and Jennifer Dy. 2025. Star: Stability-inducing weight perturbation for continual learning. *arXiv preprint arXiv:2503.01595*.

Yujie Feng, Xu Chu, Yongxin Xu, Zexin Lu, Bo Liu, Philip S Yu, and Xiao-Ming Wu. 2024a. Kif: Knowledge identification and fusion for language model continual learning. *arXiv preprint arXiv:2408.05200*.

Yujie Feng, Xu Chu, Yongxin Xu, Guangyuan Shi, Bo Liu, and Xiao-Ming Wu. 2024b. Tasl: Continual dialog state tracking via task skill localization and consolidation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1266–1279.

Yujie Feng, Jian Li, Xiaoyu Dong, Pengfei Xu, Xiaohui Zhou, Yujia Zhang, Zexin Lu, Yasha Wang, Alan Zhao, Xu Chu, and 1 others. 2025a. Aimmerging: Adaptive iterative model merging using training trajectories for language model continual learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13431–13448.

Yujie Feng, Bo Liu, Xiaoyu Dong, Zexin Lu, Li-Ming Zhan, Xiao-Ming Wu, and Albert Lam. 2024c. Continual dialogue state tracking via reason-of-select distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7075–7087.

Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiao-Ming Wu. 2023. Towards llm-driven dialogue state tracking. *arXiv preprint arXiv:2310.14970*.

Yujie Feng, Xujia Wang, Zexin Lu, Shenghong Fu, Guangyuan Shi, Yongxin Xu, Yasha Wang, Philip S

Yu, Xu Chu, and Xiao-Ming Wu. 2025b. Recurrent knowledge identification and fusion for language model continual learning. *arXiv preprint arXiv:2502.17510*.

Jinghan He, Haiyun Guo, Kuan Zhu, Zihan Zhao, Ming Tang, and Jinqiao Wang. 2024. Seekr: Selective attention-guided knowledge retention for continual learning of large language models. *arXiv preprint arXiv:2411.06171*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Zhiyuan Hu, Yuliang Liu, Jinman Zhao, Suyuchen Wang, WangYan WangYan, Wei Shen, Qing Gu, Anh Tuan Luu, See-Kiong Ng, Zhiwei Jiang, and Bryan Hooi. 2025. LongRecipe: Recipe for efficient long context generalization in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11857–11870, Vienna, Austria. Association for Computational Linguistics.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. *arXiv preprint arXiv:2403.01244*.

Wei Huang, Anda Cheng, and Yinggui Wang. 2025. Mitigating catastrophic forgetting in large language models with forgetting-aware pruning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21853–21867.

Gangwei Jiang, Caigao Jiang, Zhaoyi Li, Siqiao Xue, Jun Zhou, Linqi Song, Defu Lian, and Ying Wei. 2024. Interpretable catastrophic forgetting of large language model fine-tuning via instruction vector. *arXiv preprint arXiv:2406.12227*.

Gangwei Jiang, Caigao Jiang, Zhaoyi Li, Siqiao Xue, Jun Zhou, Linqi Song, Defu Lian, and Ying Wei. 2025. Unlocking the power of function vectors for characterizing and mitigating catastrophic forgetting in continual instruction tuning. *arXiv preprint arXiv:2502.11019*.

Hankyul Kang, Gregor Seifer, Donghyun Lee, and Jongbin Ryu. 2025a. Do your best and get enough rest for continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10077–10086.

Zhaolu Kang, Junhao Gong, Jiaxu Yan, Wanke Xia, Yian Wang, Ziwen Wang, Huaxuan Ding, Zhuo Cheng, Wenhao Cao, Zhiyuan Feng, Siqi He, Shannan Yan, Junzhe Chen, Xiaomin He, Chaoya Jiang, Wei Ye, Kaidong Yu, and Xuelong Li. 2025b. Hss-bench: Benchmarking humanities and social sciences ability for multimodal large language models. *Preprint*, arXiv:2506.03922.

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

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*, 34:22443–22456.

Zixuan Ke, Bing Liu, Wenhan Xiong, Asli Celikyilmaz, and Haoran Li. 2023. Sub-network discovery and soft-masking for continual learning of mixed tasks. *arXiv preprint arXiv:2310.09436*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Dylan Kline. 2025. Human-like forgetting curves in deep neural networks. *arXiv preprint arXiv:2506.12034*.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting catastrophic forgetting in large language model tuning. *arXiv preprint arXiv:2406.04836*.

Junzhuo Li, Bo Wang, Xiuze Zhou, and Xuming Hu. 2025. Dynamic expert specialization: Towards catastrophic forgetting-free multi-domain moe adaptation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18489–18504.

Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, and 1 others. 2025. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*.

Aojun Lu, Hangjie Yuan, Tao Feng, and Yanan Sun. 2025. Rethinking the stability-plasticity trade-off in continual learning from an architectural perspective. *Preprint*, arXiv:2506.03951.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. 2024. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Saleh Momeni, Changnan Xiao, and Bing Liu. 2025. Anacp: Toward upper-bound continual learning via analytic contrastive projection. *arXiv preprint arXiv:2511.13880*.

Jaap MJ Murre and Joeri Dros. 2015. Replication and analysis of ebbinghaus' forgetting curve. *PloS one*, 10(7):e0120644.

Kai Tzu-iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seung-won Hwang, Dongha Lee, and Jinyoung Yeo. 2025. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8631–8661.

Quang Pham, Chenghao Liu, and Steven CH Hoi. 2023. Continual learning, fast and slow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Anastasia Razdai, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2022. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations*.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*.

Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. 2024. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*.

Guangyuan Shi, Zexin Lu, Xiaoyu Dong, Wenlong Zhang, Xuanyu Zhang, Yujie Feng, and Xiao-Ming Wu. 2024. Understanding layer significance in llm alignment. *arXiv preprint arXiv:2410.17875*.

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. 2025. Merging models on the fly without retraining: A sequential approach to scalable continual model merging. *Preprint*, arXiv:2501.09522.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge fusion of large language models. *arXiv preprint arXiv:2401.10491*.

Hongyi Wan, Shiyuan Ren, Wei Huang, Miao Zhang, Xiang Deng, Yixin Bao, and Liqiang Nie. 2025a. Understanding the forgetting of (Replay-based) continual learning via feature learning: Angle matters. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 61956–62019. PMLR.

Qian Wan, Wangzi Shi, Jintian Feng, Shengyingjie Liu, Luona Wei, Zhicheng Dai, and Jianwen Sun. 2025b. Empowering math problem generation and reasoning for large language model via synthetic data based continual learning framework. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23983–24002.

Bowen Wang, Haiyuan Wan, Liwen Shi, Chen Yang, Peng He, Yue Ma, Haochen Han, Wenhao Li, Tiao Tan, Yongjian Li, and 1 others. 2025a. Recall: Representation-aligned catastrophic-forgetting alleviation via hierarchical model merging. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16392–16406.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. 2024. Self-expansion of pre-trained models with mixture of adapters for continual learning. *arXiv preprint arXiv:2403.18886*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, and 1 others. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Zhilin Wang, Yafu Li, Xiaoye Qu, and Yu Cheng. 2025b. See: Continual fine-tuning with sequential ensemble of experts. *arXiv preprint arXiv:2504.06664*.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025a. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *Preprint*, arXiv:2504.15965.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025b. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *arXiv preprint arXiv:2504.15965*.

Yongxin Xu, Ruizhe Zhang, Xinke Jiang, Yujie Feng, Yuzhen Xiao, Xinyu Ma, Runchuan Zhu, Xu Chu, Junfeng Zhao, and Yasha Wang. 2025. Parenting: Optimizing knowledge selection of retrieval-augmented language models with parameter decoupling and tailored tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11643–11662.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Dianzhi Yu, Xinni Zhang, Yankai Chen, Aiwei Liu, Yifei Zhang, Philip S Yu, and Irwin King. 2024. Recent advances of multimodal continual learning: A comprehensive survey. *arXiv preprint arXiv:2410.05352*.

Fanhu Zeng, Fei Zhu, Haiyang Guo, Xu-Yao Zhang, and Cheng-Lin Liu. 2025a. Modalprompt: Towards efficient multimodal continual instruction tuning with dual-modality guided prompt. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12137–12152.

Min Zeng, Haiqin Yang, Xi Chen, and Yike Guo. 2025b. Task-wrapped continual learning in task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3173–3183, Albuquerque, New Mexico. Association for Computational Linguistics.

Dingkun Zhang, Shuhan Qi, Xinyu Xiao, Kehai Chen, and Xuan Wang. 2025a. Merge then realign: Simple and effective modality-incremental continual learning for multimodal llms. *arXiv preprint arXiv:2503.07663*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025b. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47.

Jinman Zhao and Xueyan Zhang. 2024. Large language model is not a (multilingual) compositional relation reasoner. In *First Conference on Language Modeling*.

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

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

Method	OP ↑	BWT ↑
Recurrent-KIF	46.0	-4.1
AIMMerging	46.4	-3.0
VBM	46.4	-3.1
FOREVER	**47.5**	**-2.2**

Table 4: Results under full-parameter fine-tuning.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. 2024. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*.

Yixiao Zhou, Ziyu Zhao, Dongzhou Cheng, Zhiliang Wu, Jie Gui, Yi Yang, Fei Wu, Yu Cheng, and Hehe Fan. 2025. Dropping experts, recombining neurons: Retraining-free pruning for sparse mixture-of-experts llms. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15169–15186.

A Usage of Language Models

We used a large language model (LLM) for editorial tasks such as proofreading, grammar correction, and enhancing clarity and readability.

B Additional Results

B.1 Generalization to Full-Parameter Fine-Tuning

We further evaluate whether our method generalizes beyond parameter-efficient fine-tuning. While the main experiments are conducted under LoRA-based adaptation, the proposed model-centric time formulation is not tied to any specific parameterization.

To validate this, we conduct additional experiments under full-parameter fine-tuning on SuperNI (Qwen3-0.6B, task order 7). Results are summarized in Table 4.

FOREVER achieves the best performance on both OP and BWT. Compared with the strongest baseline, it improves OP by +1.1 and reduces forgetting (BWT) by +0.9. These results demonstrate that our approach remains effective when all model parameters are updated.

Overall, this suggests that the proposed model-centric time calibration is a general mechanism for continual learning, rather than being specific to LoRA-based adaptation.

B.2 Effect of the Memory Size

We examine the effect of varying memory size on the performance of VBM and FOREVER. Specifically, we adjust the memory size per task $|M|$ to $\{2\%, 5\%, 10\%, 50\%\}$, and report the results on SuperNI benchmark in Table 5.

As expected, increasing the memory size consistently improves performance for both methods, since more historical samples provide stronger supervision for mitigating forgetting. However, under the same memory budget, FOREVER consistently outperforms VBM across all memory sizes. This advantage stems from FOREVER's dynamics-aware replay strategy, which selectively schedules replay based on model-centric time and adaptively controls replay strength.

Notably, the performance gap between FOREVER and MixReplay is more pronounced in low-memory regimes (e.g., $|M| = 2\%$ and 5%), indicating that FOREVER makes more efficient use of limited memory resources. These results suggest that aligning replay with model update dynamics is particularly beneficial when memory capacity is constrained, a setting that is common in practical continual learning scenarios.

	Memory Size			
	2%	5%	10%	50%
VBM	41.2	42.4	43.0	45.4
FOREVER	42.5	43.5	43.9	46.4

Table 5: Ablation study on memory size, using Qwen3-0.6B as the backbone.

B.3 Sensitivity Analysis of Hyperparameters

We analyze the sensitivity of FOREVER to the warm-up length S, which is used to calibrate the virtual model day and the baseline update intensity. Intuitively, S controls the scale of model-centric time but should not critically affect replay behavior as long as it provides a stable estimate of early training dynamics. We vary S from $\{12, 24, 48\}$ steps, where $S = 24$ is the default choice used in all main experiments. The results on the SuperNI benchmark (task order 4) are reported in Table 6.

Overall, FOREVER exhibits robust performance across different values of S. While extremely small warm-up windows may introduce minor variance due to noisier update estimates, performance remains stable once S is sufficiently large. Notably, $S = 24$ consistently achieves strong results across all metrics, striking a good balance between calibration stability and computational overhead.

These results indicate that FOREVER is not sensitive to the precise choice of S, confirming that the proposed model-centric time calibration serves as a coarse-grained normalization of learning dynamics rather than a finely tuned hyperparameter.

Warm-up S	OP (↑)	BWT (↑)
12	42.0	-3.4
24	**42.5**	**-2.8**
48	42.3	-3.0

Table 6: **Sensitivity analysis of the warm-up length S.** S controls the calibration of virtual model days and baseline update intensity. Results are reported on the SuperNI benchmark (task order 7).

B.4 Sensitivity Analysis of Memory Epochs

We further analyze the sensitivity of FOREVER to the number of replay epochs performed at each replay event. Specifically, when replay is triggered, we vary the number of training epochs on the memory buffer, denoted as $E_{\text{mem}} \in \{1, 2, 4\}$, while keeping all other settings fixed. $E_{\text{mem}} = 2$ is the default choice used in all main experiments.

Table 7 reports the results. Overall, FOREVER demonstrates stable performance across different choices of E_{mem}. Increasing the number of memory epochs generally leads to slightly better retention of previous tasks, reflected by modest improvements in OP, but the gains quickly saturate. In contrast, using too many replay epochs does not yield proportional benefits and may introduce unnecessary computational overhead.

Based on this observation, we set $E_{\text{mem}} = 2$ as the default choice in all experiments, which achieves a good balance between performance and efficiency. This result indicates that FOREVER does not rely on aggressive replay, and that its effectiveness mainly stems from *when* replay is triggered and *how* replay strength is adaptively controlled, rather than from repeatedly revisiting memory samples.

E_{mem}	OP	BWT
1	42.2	-3.0
2	42.5	-2.8
4	42.6	-2.9

Table 7: Sensitivity to the number of replay epochs per replay event on the SuperNI benchmark.

B.5 Time Complexity Analysis

We analyze the computational overhead of FOREVER by comparing its training time with representative baseline methods across different model scales, as reported in Table 8.

Overall, FOREVER incurs only a modest increase in training time compared to standard replay-based baselines such as MixReplay, while remaining consistently more efficient than step-based Ebbinghaus replay methods (VBM). This efficiency arises from the design of FOREVER, which relies solely on lightweight parameter update statistics—specifically, the update magnitude and its exponential moving average—to guide both replay timing and replay strength.

In contrast to model merging–based approaches such as AIMMerging, which require additional bookkeeping or repeated parameter fusion, FOREVER avoids expensive auxiliary optimization and task-wise parameter importance estimation. As a result, its training time scales smoothly with model size, making it well suited for large-scale LLM continual learning settings.

Compared to VBM, which triggers replay at fixed step-based intervals and requires additional computation to optimize recall schedules, FOREVER schedules replay based on accumulated model evolution. Consequently, replay events are triggered more selectively, leading to fewer unnecessary replay phases and improved runtime efficiency, particularly for larger models.

Training Time (Min/Epoch)	Qwen3-0.6B	Qwen3-4B	LLaMA3.1-8B	LLaMA2-13B
MixReplay	1.3	3.5	5.6	6.8
AIMMerging	1.5	4.9	8.0	9.9
VBM	1.4	3.9	7.2	9.1
FOREVER	1.4	3.8	6.9	8.5

Table 8: Training time comparison across backbones.

B.6 Empirical Analysis of Forgetting Dynamics

We further investigate whether the fast-then-slow forgetting pattern holds in our continual learning setting. Instead of assuming the applicability of cognitive theories, we directly measure forgetting dynamics during training.

Under a representative configuration (Qwen3-0.6B on SuperNI, task order 7), we periodically evaluate the loss on previously learned tasks while training on new tasks, without updating their gradients. We analyze 10 consecutive tasks (task IDs 6–15) and report the average behavior across runs.

Training Phase	Step Range	Avg Forgetting Rate
1st	0–60	0.0023
2nd	60–120	0.0018
3rd	120–180	0.0010
4th	180–240	0.0005
5th	240–300	0.0003

Table 9: Average forgetting rate (ΔLoss/N_{step}) across training phases.

To quantify forgetting speed, we divide the training trajectory into consecutive intervals (e.g., 0–60, 60–120 steps). For each interval, we compute the loss increase ΔLoss between the end and the beginning of the interval, and normalize it by the number of steps N_{step}. This gives the average forgetting rate within each phase. Results are shown in Table 9.

We observe that forgetting is substantially faster in early stages and gradually slows down over time, exhibiting a clear fast-then-slow decay pattern. The early-stage forgetting rate is approximately 7$\times$ higher than that in later stages, indicating strongly non-linear dynamics. This empirical observation supports the structural motivation of increasing-spacing replay. Importantly, we do not assume biological equivalence between human cognition and model training. Instead, our findings suggest that the fast-then-slow decay pattern underlying spaced replay also emerges in LLM continual learning, providing empirical justification for our design.

B.7 Effect of Initial Window Size S

We further analyze the effect of the initial window size S, which determines the scale of model-time in our framework. Specifically, S defines one virtual "model day" as $\tau_{\text{day}} = \sum_{i=1}^{S} \Delta\theta_i$, i.e., the accumulated parameter change corresponding to one unit of model-time. Thus, S rescales the model-time axis but does not alter the underlying increasing-spacing replay structure.

In the main paper, we reported ablation results for $S \in \{12, 24, 48\}$. Here, we extend the range to both smaller and larger values to more comprehensively evaluate robustness.

Results on SuperNI (Qwen3-0.6B, task order 7) are shown in Table 10. We observe that performance remains stable across a broad intermediate range (approximately 6–48), forming a clear plateau. Noticeable degradation occurs only at extreme values (e.g., $S = 3$ or $S = 96$).

This behavior is consistent with the role of S.

Initial Window Size S	OP ↑	BWT ↑
3	41.8	-3.8
6	42.1	-3.7
12	42.0	-3.4
24	**42.5**	**-2.8**
48	42.3	-3.0
96	41.2	-4.4

Table 10: Effect of the initial window size S on performance.

When S is too small, a model-day becomes short, leading to overly frequent replay, which increases computational overhead and may over-regularize adaptation. Conversely, when S is too large, replay becomes too sparse, allowing more forgetting to accumulate.

Overall, these results indicate that S primarily controls the temporal scale of replay rather than its structural behavior. As long as S lies within a reasonable range, performance remains robust, suggesting that it is not a highly sensitive hyperparameter. We adopt $S = 24$ in the main experiments, as it lies within the stable region and consistently achieves strong performance.

B.8 Scalability to Longer Task Sequences

We further evaluate the scalability of our method under longer continual learning horizons. While the main experiments consider up to 15 tasks, real-world scenarios often involve substantially longer task sequences. To this end, we construct a 30-task sequence by combining the Long Sequence benchmark (15 tasks) and SuperNI (15 tasks), followed by random shuffling to form a unified continual learning setting. Results on Qwen3-0.6B are reported in Table 11.

As the number of tasks increases, all methods experience performance degradation due to accumulated forgetting. Nevertheless, FOREVER consistently achieves the strongest performance, improving OP by approximately +1.2 over the strongest baseline, while also obtaining better BWT.

These results demonstrate that our model-centric time calibration scales effectively to longer task sequences and provides stronger robustness against cumulative forgetting.

C Task-Adaptive Behavior of Model-Time τ_{day}

We further analyze whether the variability of replay trigger timing across tasks reflects meaningful task-dependent dynamics. In our framework, replay is triggered based on the model-time τ_{day}, which is determined by accumulated parameter updates. As a result, different tasks may reach the same replay threshold at different training steps. We examine whether this variability is consistent, interpretable, and correlated with task properties.

Method	OP ↑	BWT ↑
Recurrent-KIF	24.7	-15.0
AIMMerging	26.1	-11.6
VBM	26.7	-12.8
FOREVER	**27.9**	**-10.8**

Table 11: Results on a 30-task continual learning setting.

Stability Across Task Orders. We first evaluate whether τ_{day} is stable under different task orders. We compute τ_{day} for all 15 tasks in SuperNI under two distinct task sequences (order 7 and order 8). The resulting Pearson correlation is $r = 0.7379$ with $p = 0.0017$, indicating a strong and statistically significant correlation.

This result suggests that τ_{day} is largely invariant to task order and primarily reflects task-intrinsic learning dynamics, rather than sequence-dependent effects.

Correlation with Final Forgetting (BWT). We next examine the relationship between τ_{day} and the final backward transfer (BWT) of each task. We observe no significant correlation ($r = 0.2182$, $p = 0.4536$).

This is expected, as BWT is influenced by global cross-task interference and order-dependent interactions, whereas τ_{day} captures local parameter drift during the learning of individual tasks. Therefore, τ_{day} and final BWT measure fundamentally different aspects of the continual learning process.

Correlation with Task Difficulty. Finally, we analyze whether τ_{day} correlates with task difficulty. We use multi-task learning (MTL) performance as a proxy for task difficulty, where lower performance indicates harder tasks.

We observe a statistically significant negative correlation between τ_{day} and MTL performance ($r = -0.5625$, $p = 0.0291$). This indicates that harder tasks tend to induce larger accumulated parameter updates, resulting in larger τ_{day} values.

Overall, these results suggest that τ_{day} captures task-dependent parameter dynamics in a consistent and interpretable manner. The variability of replay

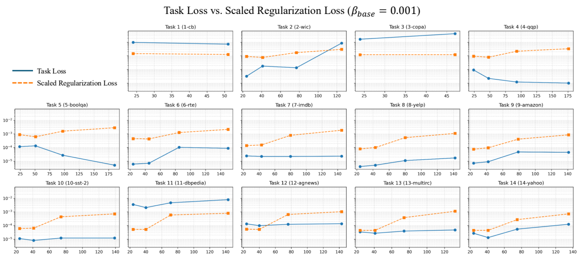


Figure 6: Visualization of task loss ($\mathcal{L}_{\text{task}}^{old}$) on memory samples and scaled replay regularization loss ($\beta_{\text{base}} \cdot \mathcal{L}_{\text{reg}}$) at replay stages.

trigger timing across tasks is therefore not incidental, but reflects adaptive behavior aligned with task difficulty and learning dynamics.

D Visualization of Task and Replay Regularization Losses

To better understand the role of the replay regularization coefficient and the design choice of β_{base}, we visualize the loss components involved during replay events. Specifically, for each replay stage, we record (i) the task loss on memory samples at the end of replay, and (ii) the scaled regularization loss applied during replay.

Figure 6 shows that after applying the base scaling factor $\beta_{\text{base}} = 0.001$, the magnitudes of the task loss and the replay regularization loss are brought onto a comparable numerical scale. This calibration is critical: without proper scaling, the replay objective would be dominated by either the task loss or the regularization term, preventing effective control of replay strength.

Importantly, this balanced scaling enables the proposed intensity-aware modulation mechanism to operate meaningfully. Once the two loss components are aligned in magnitude, the adaptive factor derived from update intensity can adjust replay strength without causing instability or overwhelming the task objective. As a result, replay regularization contributes to stabilizing previously learned knowledge while still allowing sufficient flexibility for new task adaptation. Overall, this visualization illustrates that β_{base} serves as a necessary normalization factor rather than a tuning heuristic, providing a well-conditioned foundation upon which intensity-aware replay regularization can be effectively applied.

E Visualization of Replay Dynamics

This appendix provides additional visualizations of replay dynamics for all task orders in the three benchmark (Fig. 7). For each task order, we plot the step-wise parameter update magnitude Δ_t, the accumulated model-centric time τ_t, and the corresponding replay trigger points induced by FOREVER.

These visualizations further illustrate that parameter update dynamics vary substantially across tasks and training stages, leading to non-uniform growth of τ_t. As a result, replay events are triggered at different training steps for different tasks, even when following the same Ebbinghaus-inspired schedule. This behavior highlights the adaptive nature of the proposed model-centric replay mechanism and contrasts with step-based replay strategies, which trigger replay at fixed iteration indices regardless of model state.

F Baselines

We compare FOREVER with a comprehensive set of CL baselines, with a particular emphasis on replay–based methods that vary in their replay scheduling strategies. For fairness, all methods are implemented using the LoRA framework. (1) ***MixReplay***: Mixes samples from the current task and the memory buffer, and trains on the combined data at each step. (2) ***Fixed-interval Replay***: Triggers replay at fixed, step-based intervals during training. (3) ***EWC*** (Kirkpatrick et al., 2017): Applies a regularization term to prevent interference with previously learned tasks. (4) ***O-LoRA*** (Wang et al., 2023): Learns task-specific LoRA adapters in orthogonal subspaces. (5) ***MoELoRA*** (Luo et al., 2024): Adopts the mixture-of-experts architecture for continual learning. (6) ***SAPT*** (Zhao et al., 2024): Uses a shared-attention mechanism to align LoRA block selection across tasks. (7) ***MIGU*** (Du et al., 2024): Updates task-relevant parameters based on gradient magnitude to mitigate forgetting. (8) ***SSR*** (Huang et al., 2024): A self-synthesized rehearsal method that generates synthetic instances using the LLM for rehearsal. (9) ***Recurrent-KIF*** (Feng et al., 2025b): Performs model merging with dynamic parameter importance estimation. (10) ***AIMMerging*** (Feng et al., 2025a): Applies adaptive iterative model merging using training trajectories. (11) ***VBM*** (Kang et al., 2025a): A training step-based replay method inspired by the Ebbinghaus forgetting curve. To ensure a fair comparison, all methods are trained for the same number of epochs. Finally, ***MTL*** (multi-task learning), which jointly trains on all tasks, serves as an upper-bound reference.

G Dataset Statistics

We adopt the experimental setup from Feng et al. (2024a), using three CL benchmark datasets: (i) **Standard CL Benchmark**, which consists of five text classification tasks from Zhang et al. (2015): AG News, Amazon Reviews, Yelp Reviews, DBpedia, and Yahoo Answers. (ii) **Long Sequence Benchmark**, a more challenging evaluation scenario comprising 15 tasks (Razdaibiedina et al., 2023): five from the Standard CL Benchmark, four from the GLUE benchmark, five from SuperGLUE, and the IMDB Movie Reviews dataset. (iii) **SuperNI Benchmark** (Wang et al., 2022), a comprehensive benchmark designed to evaluate a wide range of NLP tasks, includes tasks in dialogue generation (Dong et al., 2024), information extraction (Chen et al., 2025b), question answering (Zhao and Zhang, 2024), summarization (Feng et al., 2024c), and sentiment analysis (Chen et al., 2024).

Table 12 & 13 show details of the datasets we used for our experiments, along with their evaluation metrics. Overall, in SuperNI, we choose 3 tasks from dialogue generation (Dialog) (Chu et al., 2025a; Feng et al., 2023), information extraction (IE) (Xu et al., 2025), question answering (QA), summarization (Sum) (Shi et al., 2024) and sentiment analysis (SA) (Zhou et al., 2025), respectively. For the Long Sequence benchmark, this includes five tasks from the standard CL benchmark (AG News, Amazon reviews, Yelp reviews, DBpedia and Yahoo Answers), four from GLUE benchmark (MNLI, QQP, RTE, SST2), five from SuperGLUE benchmark (WiC, CB, COPA, MultiRC, BoolQ), and the IMDB movie reviews dataset (Chu et al., 2025b; Hu et al., 2025; Kang et al., 2025b).

We report 8 different task orders used for our experiments in Table 14.

H Implementation Details

All experiments are implemented using PyTorch and the Transformers library, and conducted on 8 NVIDIA H20 GPUs. We use a learning rate of 3×10^{-4} and a batch size of 8. Each new task is trained for 10 epochs, and when replay is triggered, previously learned tasks are trained for two additional epochs.

For LoRA, we set the rank $r = 8$, scaling factor $\alpha = 32$, and dropout rate to 0.05, and apply LoRA to the `q_proj` and `v_proj` modules. During inference, we use a temperature of 0.02, with top-p sampling disabled, top-k=1, beam size set to 1, and a maximum of 128 newly generated tokens.

It is worth noting that we use the same hyperparameters across different datasets and backbones, demonstrating the generalizability of our method without requiring extensive hyperparameter tuning for each specific setting. And all baselines are evaluated using identical model architectures, training configurations, and memory buffer sizes.

The detailed implementation of FOREVER is provided in Algorithm 1.

Dataset name	Task	Metric
1. task639_multi_woz_user_utterance_generation	dialogue generation	Rouge-L
2. task1590_diplomacy_text_generation	dialogue generation	Rouge-L
3. task1729_personachat_generate_next	dialogue generation	Rouge-L
4. task181_outcome_extraction	information extraction	Rouge-L
5. task748_glucose_reverse_cause_event_detection	information extraction	Rouge-L
6. task1510_evalution_relation_extraction	information extraction	Rouge-L
7. task002_quoref_answer_generation	question answering	Rouge-L
8. task073_commonsenseqa_answer_generation	question answering	Rouge-L
9. task591_sciq_answer_generation	question answering	Rouge-L
10. task511_reddit_tifu_long_text_summarization	summarization	Rouge-L
11. task1290_xsum_summarization	summarization	Rouge-L
12. task1572_samsum_summary	summarization	Rouge-L
13. task363_sst2_polarity_classification	sentiment analysis	accuracy
14. task875_emotion_classification	sentiment analysis	accuracy
15. task1687_sentiment140_classification	sentiment analysis	accuracy

Table 12: The details of 15 datasets in the SuperNI Benchmark (Wang et al., 2022).

Dataset name	Category	Task	Domain	Metric
1. Yelp	CL Benchmark	sentiment analysis	Yelp reviews	accuracy
2. Amazon	CL Benchmark	sentiment analysis	Amazon reviews	accuracy
3. DBpedia	CL Benchmark	topic classification	Wikipedia	accuracy
4. Yahoo	CL Benchmark	topic classification	Yahoo Q&A	accuracy
5. AG News	CL Benchmark	topic classification	news	accuracy
6. MNLI	GLUE	natural language inference	various	accuracy
7. QQP	GLUE	paragraph detection	Quora	accuracy
8. RTE	GLUE	natural language inference	news, Wikipedia	accuracy
9. SST-2	GLUE	sentiment analysis	movie reviews	accuracy
10. WiC	SuperGLUE	word sense disambiguation	lexical databases	accuracy
11. CB	SuperGLUE	natural language inference	various	accuracy
12. COPA	SuperGLUE	question and answering	blogs, encyclopedia	accuracy
13. BoolQA	SuperGLUE	boolean question and answering	Wikipedia	accuracy
14. MultiRC	SuperGLUE	question and answering	various	accuracy
15. IMDB	SuperGLUE	sentiment analysis	movie reviews	accuracy

Table 13: The details of 15 classification datasets in the Long Sequence Benchmark (Razdai et al., 2022). First five tasks correspond to the standard CL benchmark (Zhang et al., 2015).

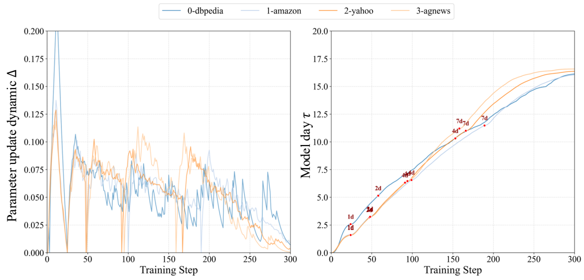
Order	Benchmark	Task Sequence
1		dbpedia → amazon → yahoo → ag
2	Standard CL	dbpedia → amazon → ag → yahoo
3		yahoo → amazon → ag → dbpedia
4		mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
5	Long Sequence	multirc → boolqa → wic → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yelp → amazon → yahoo
6		yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic
7	SuperNI	task1572 → task363 → task1290 → task181 → task002 → task1510 → task639 → task1729 → task073 → task1590 → task748 → task511 → task591 → task1687 → task875
8		task748 → task073 → task1590 → task639 → task1572 → task1687 → task591 → task363 → task1510 → task1729 → task181 → task511 → task002 → task1290 → task875

Table 14: Eight different orders of task sequences used for continual learning experiments. Orders 1-3 correspond to the standard CL becnhmark adopted by prior works. Orders 4-6 are long-sequence orders spanning 15 tasks, and orders 7-8 are superni spanning 15 tasks following (Razdaibiedina et al., 2023).

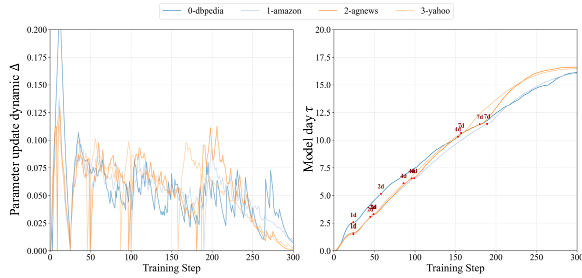
Algorithm 1 FOREVER: **FOR**g**E**tting cur**V**e-inspired m**E**mory **R**eplay

Require: Current task dataset $\mathcal{D}_k$; memory buffer $\mathcal{M}_{<k}$; model weights after training on task $k-1$, Θ^{k-1}; human replay intervals $\mathcal{D}_{\text{human}}$; warm-up length S; EMA coefficient λ; scaling factor γ; clipping bounds $g_{\min}, g_{\max}$; base coefficient β_{base}.
Ensure: Final parameters Θ^k.

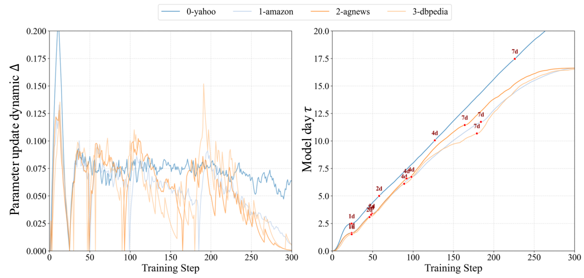
$\Theta^\star \leftarrow \Theta^{k-1}$ *# anchor from previous task*
Initialize $\tau \leftarrow 0$, $\mu \leftarrow 0$, $j \leftarrow 1$
 # Warm-up calibration
Compute $\{\Delta_t\}_{t=1}^{S}$ on $\mathcal{D}_k$
$\tau_{\text{day}} \leftarrow \sum_{t=1}^{S} \Delta_t$
$\mu_0 \leftarrow \frac{1}{S}\sum_{t=1}^{S} \Delta_t$, $\mu \leftarrow \mu_0$
$\mathcal{D}_{\text{model}} \leftarrow \{d \cdot \tau_{\text{day}} \mid d \in \mathcal{D}_{\text{human}}\}$
 # Training with scheduled replay
while training on $\mathcal{D}_k$ **do**
  Update Θ on current task data
  Measure update Δ_t; update $\tau \leftarrow \tau + \Delta_t$
  Update $\mu \leftarrow (1-\lambda)\mu + \lambda\Delta_t$
  if $j \le |\mathcal{D}_{\text{model}}|$ **and** $\tau \ge \mathcal{D}_{\text{model}}[j]$ **and** $k > 1$ **then**
    # Replay begin
    $r \leftarrow \mu/(\mu_0 + \epsilon)$
    $s \leftarrow \text{clip}(1 + \gamma(r-1), g_{\min}, g_{\max})$
    $\beta \leftarrow \beta_{\text{base}} \cdot s$
    Optimize replay loss on $\mathcal{M}$ for two replay epochs
    $j \leftarrow j + 1$
  end if
end while
 # End-of-task consolidation
Optimize replay loss on $\mathcal{M}$ for one short epoch via Eq. (11)
Update memory buffer $\mathcal{M}_{<k+1}$
return Θ^k



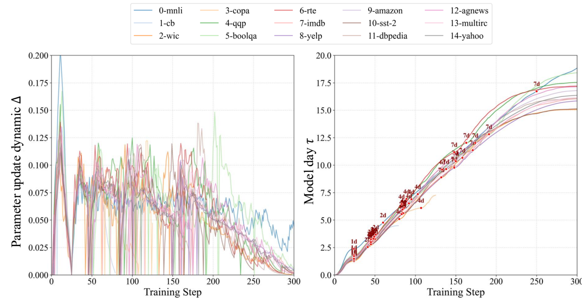
(a) Standard CL Benchmark, task order 1



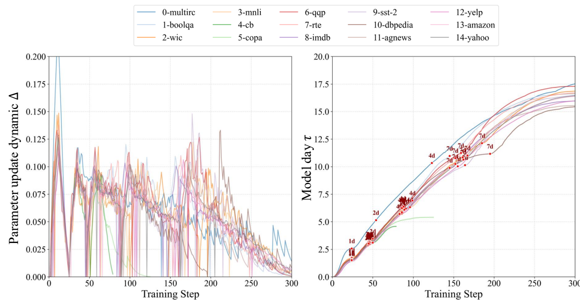
(b) Standard CL Benchmark, task order 2



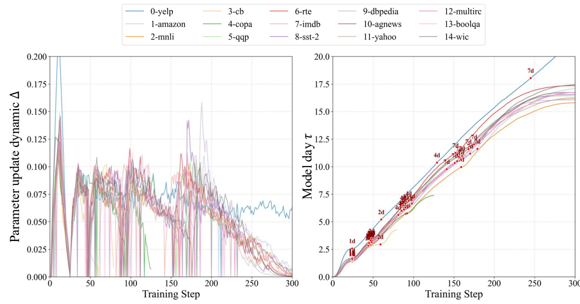
(c) Standard CL Benchmark, task order 3



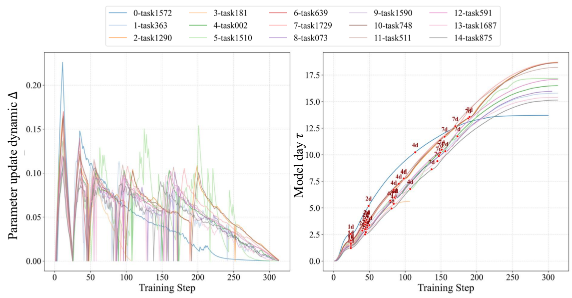
(d) Long Sequence Benchmark, task order 4



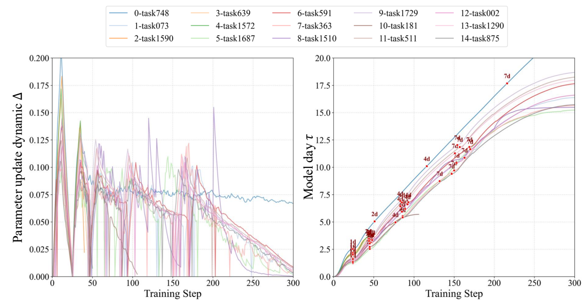
(e) Long Sequence Benchmark, task order 5



(f) Long Sequence Benchmark, task order 6



(g) SuperNI Benchmark, task order 7



(h) SuperNI Benchmark, task order 8

Figure 7: **Replay dynamics across datasets and task orders.** Each subfigure shows Δ_t, accumulated model time τ_t, and replay trigger points for one task order, illustrating adaptive replay scheduling based on model-centric time.