UNDERSTANDING FORGETTING DYNAMICS IN FULL FINE-TUNING OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Full fine-tuning can cause large language models to forget previously mastered knowledge. Understanding these forgetting dynamics is essential for balancing adaptation and retention. Existing analyses based on neural tangent kernel (NTK) theory commonly assume a fixed gradient kernel. Our study shows that during full fine-tuning, the gradient kernel evolves primarily through a rapid decay in norm while retaining a moderately stable relative structure. Kernel evolution affects predictions under the NTK framework by substantially altering the response magnitude while having a smaller effect on its direction. To investigate how the direction-aligned but magnitude-biased NTK-based approximation affects parameter optimization, we introduce and compare two Pontryagin's maximum principle (PMP)-based parameter optimization methods, PG-NoDe and PG-De. Results show that, in our setting, a direction-aligned but magnitude-biased approximation can guide replay-weight optimization toward effective schedules. Calibrating the absolute prediction magnitudes provides limited additional benefit.

1 INTRODUCTION

Large language models (LLMs) exhibit broad knowledge and reasoning capabilities acquired through large-scale pretraining (Yang et al., 2026a). Full fine-tuning further adapts them to new knowledge and tasks (You et al., 2025), but updating all parameters can damage previously correct answers and cause catastrophic forgetting (Li et al., 2024; Jin & Ren, 2024). Mitigation methods broadly use gradient projection (Wang et al., 2025; Yang et al., 2026b), regularization (Kirkpatrick et al., 2016; Li et al., 2024), parameter restriction (Wang et al., 2023; Zheng et al., 2025), or replay (Jin & Ren, 2024; Wang et al., 2024).

Dynamic control of training parameters can better balance learning new knowledge against retaining mastered knowledge. Existing approaches broadly follow two strategies. Feedback-based methods adjust loss weights, sampling ratios, or related controls from current or recent training signals and therefore require no advance prediction of the trajectory, but their decisions primarily reflect local progress (Liu et al., 2019; Chen et al., 2026a). Trajectory-based methods instead optimize a finite-horizon objective, but must anticipate how candidate controls affect the training process. Existing methods commonly approximate these effects using target or proxy training trajectories (Gu et al., 2025; Chen et al., 2026b). Accurate modeling of the training process is therefore central to global training-parameter optimization.

Modeling LLM full-fine-tuning dynamics is challenging because both prediction residuals and gradient kernels evolve during training. NTK theory treats the kernel as approximately constant, providing a theoretical basis for approximating these dynamics (Jacot et al., 2018). This foundation has motivated a range of NTK-based analyses of training behavior. NTK overlap has been used to analyze interference and gradient projection (Doan et al., 2021). Kernel views further explain how language models adapt during fine-tuning (Malladi et al., 2023; Liu et al., 2026). Most of these analyses use the fixed-kernel approximation justified in the infinite-width regime, whereas practical LLMs have finite width and their kernels can evolve during training.

This gap leads to three connected questions, which organize our contributions:

• **Does the gradient kernel evolve during LLM full fine-tuning?** Yes, primarily in scale: its norm decays rapidly while its relative structure remains moderately stable.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

• **How does kernel evolution affect the NTK-based approximation framework?** It mainly affects the predicted response magnitude while changing the response direction much less. Fitting the decay of the gradient kernel can further reduce errors in the predicted response magnitude.

• **How does kernel evolution affect training-parameter optimization?** Its effect on replay parameter optimization is smaller than its effect on the predicted response magnitude. In our setting, a trend-aligned but magnitude-biased approximation can effectively guide PMP optimization, while magnitude calibration offers limited additional benefit.

2 Related Work

Catastrophic forgetting in LLM full fine-tuning. Updating all parameters of an LLM on new data can degrade capabilities acquired before fine-tuning (Li et al., 2024; Jin & Ren, 2024). Mitigation methods change different parts of training: projection restricts interfering update directions (Wang et al., 2025; Yang et al., 2026b); regularization constrains important parameters (Kirkpatrick et al., 2016; Li et al., 2024); parameter restriction isolates or freezes model components (Wang et al., 2023; Zheng et al., 2025); and replay mixes stored examples into new-task training (Jin & Ren, 2024; Wang et al., 2024).

NTK models of LLM instruction fine-tuning. NTK links parameter updates to output changes through inner products of output gradients (Jacot et al., 2018). Malladi et al. (2023) study when a kernel view describes fine-tuning pretrained language models, including prompt-formulated tasks, while Xia et al. (2024) use low-dimensional gradient similarity to select examples for targeted instruction tuning. These results make gradient geometry useful for analyzing LLM adaptation. We use it to examine how kernel evolution affects mastered-answer predictions during full fine-tuning.

Training-parameter optimization. Optimizing training-time parameters is important for steering the learning trajectory toward a desired objective, independent of any particular replay mechanism. Existing approaches broadly fall into two categories. Feedback-based methods update loss weights, sampling ratios, or related controls from current or recent training signals (Liu et al., 2019; Chen et al., 2026a). Trajectory-based methods instead optimize a finite-horizon objective using target- or proxy-training rollouts (Gu et al., 2025; Chen et al., 2026b). The former are inexpensive and adaptive but primarily reflect local progress, while the latter account for delayed effects at the cost of obtaining a usable rollout.

3 Method

3.1 Kernel response of forgetting

Hereafter, "fine-tuning" refers to full fine-tuning. Following prior studies of forgetting in fine-tuning (Jin & Ren, 2024; Zhou et al., 2025), let θ_0 be the initial LLM and partition a dataset by its initial answers. The *fine-tuning set* $\mathcal{F}$ contains examples θ_0 answers incorrectly; the *mastered set* $\mathcal{M}$ contains those it answers correctly.

Building on prior NTK-based analyses of forgetting (Doan et al., 2021; Zheng et al., 2026; Liu et al., 2026), we study how fine-tuning on $\mathcal{F}$ changes the correct-answer log-odds on the mastered set $\mathcal{M}$. Let $p_{\mathcal{M}}(\theta)$ collect the model's full-vocabulary probabilities of the correct answer tokens on $\mathcal{M}$, and define the log-odds vector componentwise as

$$f_{\mathcal{M}}(\theta) = \log \frac{p_{\mathcal{M}}(\theta)}{\mathbf{1} - p_{\mathcal{M}}(\theta)}. \tag{1}$$

For parameters θ_t after t epochs, $\Delta f_{\mathcal{M}}(t) = f_{\mathcal{M}}(\theta_t) - f_{\mathcal{M}}(\theta_0)$ measures the drift on the mastered set. We define $f_{\mathcal{F}}$ analogously for the fine-tuning set. Our goal is to model $\Delta f_{\mathcal{M}}(t)$ during LLM fine-tuning.

To describe forgetting while allowing the kernel of a finite-width LLM to evolve, for $X \in \{\mathcal{M}, \mathcal{F}\}$, write $f_X(t) = f_X(\theta_t)$ and let $J_X(t)$ be its parameter Jacobian evaluated at θ_t, with one score gradient per row. Since $p_{\mathcal{F}}(t) = \sigma(f_{\mathcal{F}}(t))$ componentwise, the full-batch training loss on $\mathcal{F}$ can be written in

log-odds space as

$$\mathcal{L}_{\mathcal{F}}(\theta_t) = -\frac{1}{|\mathcal{F}|}\mathbf{1}^\top \log p_{\mathcal{F}}(t) = \frac{1}{|\mathcal{F}|}\mathbf{1}^\top \log(\mathbf{1} + \exp(-f_{\mathcal{F}}(t)))\,. \tag{2}$$

The logarithm and exponential act componentwise. Differentiating this expression with respect to the log-odds, then using the Jacobian to apply the chain rule, gives

$$\begin{aligned} \nabla_{f_{\mathcal{F}}}\mathcal{L}_{\mathcal{F}}(\theta_t) &= -\frac{1}{|\mathcal{F}|}\frac{\exp(-f_{\mathcal{F}}(t))}{\mathbf{1}+\exp(-f_{\mathcal{F}}(t))} = -\frac{1}{|\mathcal{F}|}(\mathbf{1} - p_{\mathcal{F}}(t)), \\ \nabla_\theta \mathcal{L}_{\mathcal{F}}(\theta_t) &= J_{\mathcal{F}}(t)^\top \nabla_{f_{\mathcal{F}}}\mathcal{L}_{\mathcal{F}}(\theta_t) = -\frac{1}{|\mathcal{F}|}J_{\mathcal{F}}(t)^\top(\mathbf{1} - p_{\mathcal{F}}(t)). \end{aligned} \tag{3}$$

The cross-group and fine-tuning blocks of the log-odds NTK are

$$K_{\mathcal{M}\mathcal{F}}(t) = J_{\mathcal{M}}(t)J_{\mathcal{F}}(t)^\top, \qquad K_{\mathcal{F}\mathcal{F}}(t) = J_{\mathcal{F}}(t)J_{\mathcal{F}}(t)^\top. \tag{4}$$

With constant learning rate η and a fitted global scale ρ, a first-order Taylor expansion around θ_t gives

$$\begin{aligned} \theta_{t+1} - \theta_t &= -\eta\nabla_\theta\mathcal{L}_{\mathcal{F}}(\theta_t) = \frac{\eta}{|\mathcal{F}|}J_{\mathcal{F}}(t)^\top(\mathbf{1} - p_{\mathcal{F}}(t)), \\ \begin{bmatrix} f_{\mathcal{M}}(t+1) - f_{\mathcal{M}}(t) \\ f_{\mathcal{F}}(t+1) - f_{\mathcal{F}}(t) \end{bmatrix} &\simeq \begin{bmatrix} J_{\mathcal{M}}(t) \\ J_{\mathcal{F}}(t) \end{bmatrix}(\theta_{t+1} - \theta_t) \\ &\simeq \frac{\eta\rho}{|\mathcal{F}|}\begin{bmatrix} K_{\mathcal{M}\mathcal{F}}(t) \\ K_{\mathcal{F}\mathcal{F}}(t) \end{bmatrix}(\mathbf{1} - p_{\mathcal{F}}(t)). \end{aligned} \tag{5}$$

The first block shows how the cross-group kernel transfers the fine-tuning residual to mastered scores. Summing both blocks of Eq. (5) from the initial model to step T gives

$$\begin{bmatrix} f_{\mathcal{M}}(T) - f_{\mathcal{M}}(0) \\ f_{\mathcal{F}}(T) - f_{\mathcal{F}}(0) \end{bmatrix} \simeq \frac{\eta\rho}{|\mathcal{F}|}\sum_{t=0}^{T-1}\begin{bmatrix} K_{\mathcal{M}\mathcal{F}}(t) \\ K_{\mathcal{F}\mathcal{F}}(t) \end{bmatrix}(\mathbf{1} - p_{\mathcal{F}}(t)). \tag{6}$$

3.2 Linearized response and its continuous-time approximation

Let $H = \mathrm{diag}(p_{\mathcal{F}}(0) \odot (\mathbf{1} - p_{\mathcal{F}}(0)))$. Linearizing the CE residual around its initial score gives

$$\mathbf{1} - p_{\mathcal{F}}(t) \simeq \mathbf{1} - p_{\mathcal{F}}(0) - H[f_{\mathcal{F}}(t) - f_{\mathcal{F}}(0)]. \tag{7}$$

Since $p_{\mathcal{F}}(t) = \sigma(f_{\mathcal{F}}(t))$, the fine-tuning block of Eq. (5) gives the probability update

$$p_{\mathcal{F}}(t+1) \simeq \sigma\left(f_{\mathcal{F}}(t) + \frac{\eta\rho}{|\mathcal{F}|}K_{\mathcal{F}\mathcal{F}}(t)(\mathbf{1} - p_{\mathcal{F}}(t))\right). \tag{8}$$

Applying the linearization above to this update gives the residual transition

$$\mathbf{1} - p_{\mathcal{F}}(t+1) \simeq \left(I - \frac{\eta\rho}{|\mathcal{F}|}HK_{\mathcal{F}\mathcal{F}}(t)\right)(\mathbf{1} - p_{\mathcal{F}}(t)). \tag{9}$$

Existing NTK-based analyses commonly replace the time-varying kernel with its initial value, $K_{X\mathcal{F}}(t) \approx K_{X\mathcal{F}}(0)$ (Doan et al., 2021; Malladi et al., 2023; Liu et al., 2026). We instead represent its evolution through the exact decomposition

$$K_{X\mathcal{F}}(t) = c(t)K_{X\mathcal{F}}(0) + R_{X\mathcal{F}}(t), \qquad X \in \{\mathcal{M}, \mathcal{F}\}, \tag{10}$$

The first term captures a shared change in kernel scale, while the second captures the remaining geometric change that cannot be explained by a global scale factor. Eq. (9) gives the first expression below with $K_{\mathcal{F}\mathcal{F}}(t)$, while the mastered-score block in Eq. (5) gives the second with $K_{\mathcal{M}\mathcal{F}}(t)$. Replacing both kernels using Eq. (10) yields

$$\begin{aligned} \frac{d}{dt}(\mathbf{1} - p_{\mathcal{F}}(t)) &\simeq -\frac{\eta\rho}{|\mathcal{F}|}HK_{\mathcal{F}\mathcal{F}}(t)(\mathbf{1} - p_{\mathcal{F}}(t)) \\ &= -\frac{\eta\rho}{|\mathcal{F}|}H\left[c(t)K_{\mathcal{F}\mathcal{F}}(0) + R_{\mathcal{F}\mathcal{F}}(t)\right](\mathbf{1} - p_{\mathcal{F}}(t)), \\ \frac{d}{dt}f_{\mathcal{M}}(t) &\simeq \frac{\eta\rho}{|\mathcal{F}|}K_{\mathcal{M}\mathcal{F}}(t)(\mathbf{1} - p_{\mathcal{F}}(t)) \\ &= \frac{\eta\rho}{|\mathcal{F}|}\left[c(t)K_{\mathcal{M}\mathcal{F}}(0) + R_{\mathcal{M}\mathcal{F}}(t)\right](\mathbf{1} - p_{\mathcal{F}}(t)). \end{aligned} \tag{11}$$

Integrating the mastered-score dynamics gives

$$f_{\mathcal{M}}(T) - f_{\mathcal{M}}(0) \simeq \frac{\eta\rho}{|\mathcal{F}|} \int_0^T \left[c(t)K_{\mathcal{MF}}(0) + R_{\mathcal{MF}}(t)\right] (\mathbf{1} - p_{\mathcal{F}}(t))\, dt. \tag{12}$$

Let $C(t) = \int_0^t c(\tau)\, d\tau$. Adding and subtracting the residual trajectory induced by the scaled initial kernel separates the response into a predictable term and a remainder:

$$f_{\mathcal{M}}(T) - f_{\mathcal{M}}(0) \simeq m(T) + \mathcal{R}(T). \tag{13}$$

The term $m(T)$ is the response predicted by the scaled initial kernel:

$$\begin{aligned} m(T) = \frac{\eta\rho}{|\mathcal{F}|} K_{\mathcal{MF}}(0) \int_0^T c(t) \exp\left(-\frac{\eta\rho}{|\mathcal{F}|} C(t) H K_{\mathcal{FF}}(0)\right) dt \\ \cdot (\mathbf{1} - p_{\mathcal{F}}(0)). \end{aligned} \tag{14}$$

The remainder $\mathcal{R}(T)$ captures the accumulated deviation from this prediction:

$$\begin{aligned} \mathcal{R}(T) = \frac{\eta\rho}{|\mathcal{F}|} \int_0^T \Bigg\{ c(t) K_{\mathcal{MF}}(0) \Bigg[\mathbf{1} - p_{\mathcal{F}}(t) \\ - \exp\left(-\frac{\eta\rho}{|\mathcal{F}|} C(t) H K_{\mathcal{FF}}(0)\right) (\mathbf{1} - p_{\mathcal{F}}(0)) \Bigg] \\ + R_{\mathcal{MF}}(t)(\mathbf{1} - p_{\mathcal{F}}(t)) \Bigg\} dt. \end{aligned} \tag{15}$$

In the experiments, we estimate $\mathcal{R}(T)$ as the observed log-odds change $f_{\mathcal{M}}(T) - f_{\mathcal{M}}(0)$ minus the prediction $m(T)$. Using the substitution $u = C(t)$ and $du = c(t)\, dt$, we evaluate the integral defining $m(T)$ as

$$\begin{aligned} &\frac{\eta\rho}{|\mathcal{F}|} K_{\mathcal{MF}}(0) \int_0^T c(t) \exp\left(-\frac{\eta\rho}{|\mathcal{F}|} C(t) H K_{\mathcal{FF}}(0)\right) dt (\mathbf{1} - p_{\mathcal{F}}(0)) \\ &= \frac{\eta\rho}{|\mathcal{F}|} K_{\mathcal{MF}}(0) \int_0^{C(T)} \exp\left(-\frac{\eta\rho}{|\mathcal{F}|} u H K_{\mathcal{FF}}(0)\right) du \\ &\quad \cdot (\mathbf{1} - p_{\mathcal{F}}(0)) \\ &= K_{\mathcal{MF}}(0) [H K_{\mathcal{FF}}(0)]^{-1} \\ &\quad \cdot \left[I - \exp\left(-\frac{\eta\rho}{|\mathcal{F}|} C(T) H K_{\mathcal{FF}}(0)\right)\right] (\mathbf{1} - p_{\mathcal{F}}(0)), \end{aligned} \tag{16}$$

With the finite-time response factor

$$D(T) = I - \exp\left(-\frac{\eta\rho}{|\mathcal{F}|} C(T) H K_{\mathcal{FF}}(0)\right), \tag{17}$$

the predictable main term can be written as

$$m(T) = K_{\mathcal{MF}}(0) [H K_{\mathcal{FF}}(0)]^{-1} D(T) (\mathbf{1} - p_{\mathcal{F}}(0)). \tag{18}$$

Based on its effect on the predicted response, we decompose the remainder $\mathcal{R}(T)$ into components parallel and orthogonal to $m(T)$:

$$\mathcal{R}_{\parallel}(T) = \frac{\langle \mathcal{R}(T), m(T) \rangle}{\|m(T)\|_2^2} m(T), \qquad \mathcal{R}_{\perp}(T) = \mathcal{R}(T) - \mathcal{R}_{\parallel}(T), \qquad \langle \mathcal{R}_{\perp}(T), m(T) \rangle = 0, \tag{19}$$

The response thus consists of three terms: $m(T)$, predicted by the initial kernel with the fitted shared scale $c(t)$, and two remainders capturing the effects of kernel evolution beyond this shared scaling. The parallel remainder $\mathcal{R}_{\parallel}(T)$ accounts for deviations in prediction magnitude, while the orthogonal remainder $\mathcal{R}_{\perp}(T)$ accounts for deviations in prediction direction:

$$f_{\mathcal{M}}(T) - f_{\mathcal{M}}(0) \simeq m(T) + \mathcal{R}_{\parallel}(T) + \mathcal{R}_{\perp}(T). \tag{20}$$

The conventional fixed-NTK squared-loss response is a limiting case of this formulation (Liu et al., 2026). Setting $c(t) = 1$ and $\mathcal{R}(T) = 0$ recovers the initial-kernel prediction without a remainder correction. Replacing CE with MSE sets the curvature factor to the identity. As $T \to \infty$, the matrix exponential vanishes and $D(T) \to I$, giving

$$\Delta f_{\mathcal{M}}(\infty) \simeq K_{\mathcal{MF}} K_{\mathcal{FF}}^{-1} (\mathbf{1} - p_{\mathcal{F}}(0)). \tag{21}$$

3.3 PMP-GUIDED REPLAY PARAMETER OPTIMIZATION

Replay trains on stored mastered examples alongside fine-tuning examples, allowing their gradients to oppose updates that reduce previously correct-answer scores (Jin & Ren, 2024; Wang et al., 2024). Let $R \subset \mathcal{M}$ be the replay buffer and $\mathcal{B} = \mathcal{F} \cup R$. In the full-batch setting, we combine the fine-tuning and replay loss terms with weights 1 and $w_e \geq 0$, respectively

Choosing w_e separately for each epoch allows the learning–retention balance to change over training. Pontryagin's maximum principle (PMP) treats this schedule as a finite-horizon control problem, using forward trajectories and backward adjoints to measure each weight's effect on the terminal loss.

Obtaining these trajectories by repeatedly training the target LLM is costly; proxy-model trajectories still require additional training and may differ from the target model's dynamics (Gu et al., 2025; Chen et al., 2026b). We instead approximate the target model's trajectory from its initial scores and an estimated gradient-kernel trajectory using Eq. (5).

We take one epoch as one control step. Let $Z_e = |\mathcal{F}| + w_e|R|$ denote the total weight, let $s_e = f_{\mathcal{B}}(e)$ be the log-odds state after epoch e, and let $K^{(e)}_{\mathcal{B}\mathcal{B}} = J_{\mathcal{B}}(e-1)J_{\mathcal{B}}(e-1)^\top$ be the kernel governing epoch e. Applying Eq. (5) with replay weights gives the epoch-level transition

$$s_e = \Phi_e(s_{e-1}, w_e) = s_{e-1} + \frac{\eta}{Z_e} K^{(e)}_{\mathcal{B}\mathcal{B}}\big[v(w_e) \odot (\mathbf{1} - \sigma(s_{e-1}))\big], \quad v_j(w) = \begin{cases} 1, & j \in \mathcal{F}, \\ w, & j \in R, \end{cases} \tag{22}$$

where σ acts componentwise. This rollout allows the kernel geometry or scale to evolve across epochs. Given an estimated sequence $\{K^{(e)}\}_{e=1}^T$, we evaluate a proposed schedule by rolling out this transition. With $\ell(s) = \log(1 + e^{-s})$, the terminal objective is the group-balanced mean CE:

$$\Psi(s) = \frac{1}{2|\mathcal{F}|} \sum_{j \in \mathcal{F}} \ell(s_j) + \frac{1}{2|R|} \sum_{j \in R} \ell(s_j). \tag{23}$$

For a prescribed kernel trajectory, the discrete adjoint differentiates $\Psi(s_T)$ through the epoch-level transitions. Starting with $\lambda_T = \nabla_s \Psi(s_T)$, define

$$q_e = (\mathbf{1} - \sigma(s_{e-1})) \odot \big(K^{(e)}_{\mathcal{B}\mathcal{B}} \lambda_e\big).$$

The adjoint recursion and weight gradient are then

$$\begin{aligned} \lambda_{e-1} &= \big(\partial_{s_{e-1}} \Phi_e\big)^\top \lambda_e, \\ \frac{\partial \Psi(s_T)}{\partial w_e} &= \frac{\eta}{Z_e^2} \left(|\mathcal{F}| \sum_{j \in R} q_{e,j} - |R| \sum_{j \in \mathcal{F}} q_{e,j} \right). \end{aligned} \tag{24}$$

Algorithm 1 uses the forward rollout and backward sweep to evaluate terminal CE and its gradient for a candidate schedule. Pontryagin's maximum principle Guided Replay (PG-Replay) uses these quantities to select its weights.

We further regularize the schedule with $\sum_{e=1}^T w_e^2$ to discourage unnecessarily large replay weights under a terminal-loss tolerance. We solve the reference search with L-BFGS-B and the resulting constrained problem with SLSQP, both using the adjoint gradient; Appendix A gives the formulation and solution details.

Algorithm 1 PMP trajectory and adjoint

Input: Candidate weights $w_{1:T}$, initial scores s_0, kernel trajectory $\{K^{(e)}\}_{e=1}^T$, sets $\mathcal{F}$, R, learning rate η.
Output: Terminal CE $\Psi(s_T)$ and weight gradient $g_{1:T}$.

Forward: for $e = 1, \ldots, T$, set $s_e \leftarrow \Phi_e(s_{e-1}, w_e)$ [Eq. (22)].
Backward: set $\lambda_T \leftarrow \nabla_s \Psi(s_T)$; for $e = T, \ldots, 1$, set $\lambda_{e-1} \leftarrow (\partial_{s_{e-1}} \Phi_e)^\top \lambda_e$.
Gradient: for $e = 1, \ldots, T$, set $g_e \leftarrow (\partial_{w_e} \Phi_e)^\top \lambda_e$ [Eq. (24)].
return $\Psi(s_T)$ and $g_{1:T}$.

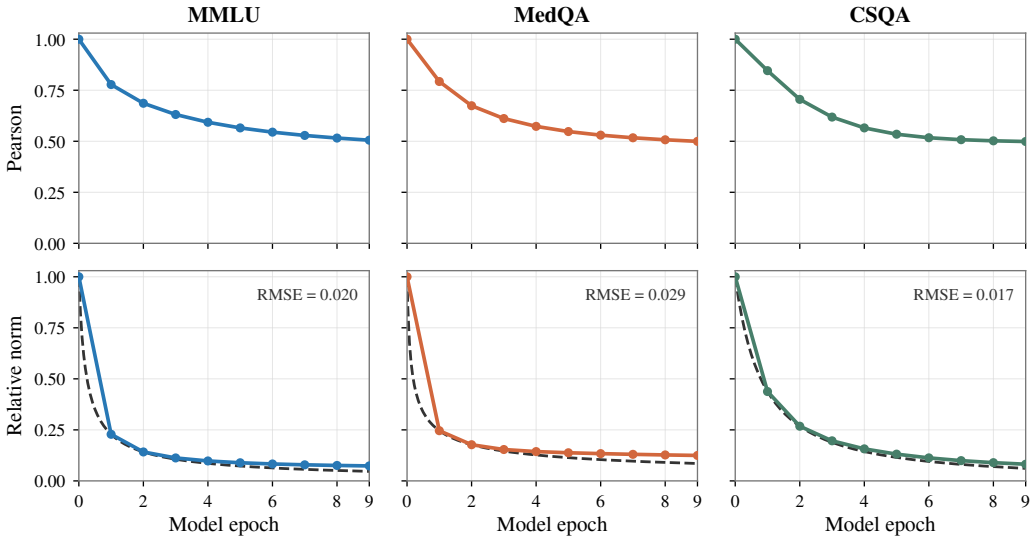


Figure 1: **Gradient-kernel evolution during fine-tuning of *Qwen 1.5B*.** The top row reports the entrywise Pearson correlation between $K_{\mathcal{MF}}(t)$ and $K_{\mathcal{MF}}(0)$; the bottom row reports $\|K_{\mathcal{MF}}(t)\|_F / \|K_{\mathcal{MF}}(0)\|_F$. Dashed curves are fitted using Eq. (30).

4 Experiments

4.1 Setup

Datasets. We use three datasets: MMLU (Hendrycks et al., 2021), CommonsenseQA (CSQA) (Talmor et al., 2019), and MedQA (Jin et al., 2020). For each dataset, we independently construct three train/test partitions using a 7:3 ratio.

Models. Our experiments involve nine LLMs: *Qwen 1.5B* and *Qwen 7B* (Qwen Team, 2024); *Llama 1B* and *Llama 3B* (Meta, 2024), and *Llama 8B* (Grattafiori et al., 2024); *Gemma 2B* and *Gemma 9B* (Gemma Team, 2024); *GLM 9B* (Team GLM, 2024); and *Mistral 7B* (Jiang et al., 2023).

Protocol. Within each partition, we divide the training and test portions into initially incorrect sets $\mathcal{F}_{\text{train}}, \mathcal{F}_{\text{test}}$ and mastered sets $\mathcal{M}_{\text{train}}, \mathcal{M}_{\text{test}}$. Fine-tuning uses $\mathcal{F}_{\text{train}}$, and reference or replay examples come from $\mathcal{M}_{\text{train}}$. We fit response parameters on $\mathcal{F}_{\text{train}} \cup \mathcal{M}_{\text{train}}$ and evaluate predictions on held-out test examples. No test example is used for fitting. Results are means $\pm$ sample standard deviations over the three partitions. The training configuration is given in Appendix B.

Baselines. For replay control, we compare PG-Replay with nine baselines: projection methods GPM and OPGD (Zheng et al., 2026); regularization method EWC (Kirkpatrick et al., 2016); parameter-restriction methods O-LoRA and bottom-layer freezing (BLF); and replay methods ER, ER + DWA, DER++, and Smart Replay (Chen et al., 2026a). Several baselines originated in continual learning, but we include them for their forgetting-mitigation mechanisms; our setting is a single model-refinement step on fixed data partitions rather than a sequence of tasks.

Metrics. For the response studies, Pearson correlation and RMSE compare predicted with observed correct-answer log-odds drift. For replay control, let $\widehat{y}_i$ be the model's answer after fine-tuning. We define accuracy (Acc) and forgetting rate (FT) as

$$\text{Acc} = \frac{\#\{i \in \mathcal{M}_{\text{test}} \cup \mathcal{F}_{\text{test}} : \widehat{y}_i = y_i\}}{|\mathcal{M}_{\text{test}} \cup \mathcal{F}_{\text{test}}|}, \qquad \text{FT} = \frac{\#\{i \in \mathcal{M}_{\text{test}} : \widehat{y}_i \neq y_i\}}{|\mathcal{M}_{\text{test}}|}. \tag{25}$$

Table 1: Endpoint prediction and remainder decomposition for *Qwen 1.5B*. Both no-decay and fitted-decay predictions maintain high directional agreement with the observed response, while fitting decay substantially reduces the parallel remainder.

Dataset	Predictor	$\|m\|_2$	$\|\mathcal{R}_{\parallel}\|_2$	$\|\mathcal{R}_{\perp}\|_2$	$\cos(m, m+\mathcal{R})$
MMLU	No decay	239 ± 2	115 ± 1	19.8 ± 0.0	0.987 ± 0.000
	Decay	127 ± 1	3.2 ± 0.5	20.3 ± 0.1	0.987 ± 0.000
MEDQA	No decay	120 ± 1	72.4 ± 0.4	8.0 ± 0.1	0.986 ± 0.001
	Decay	69.6 ± 0.1	22.2 ± 0.6	7.7 ± 0.1	0.987 ± 0.001
CSQA	No decay	389 ± 4	215 ± 3	32.5 ± 1.4	0.983 ± 0.001
	Decay	158 ± 2	14.9 ± 0.7	34.0 ± 1.6	0.981 ± 0.002

4.2 Does the gradient kernel evolve during LLM fine-tuning?

Figure 1 answers this question affirmatively. The entrywise Pearson correlation with the initial kernel remains moderate to high, but the kernel norm decreases rapidly. Thus fine-tuning preserves much of the kernel's relative structure while changing its absolute scale. To quantify this scale evolution, we fit the positive power-law decay

$$c(t) = (1 + \beta t)^{-\gamma}, \qquad \beta > 0, \quad \gamma > 0. \tag{26}$$

The resulting curve extrapolates the later norm decay with RMSE 0.017–0.029 across the three datasets. Appendix D gives the fitting methods.

4.3 How does kernel evolution affect the NTK-based approximation framework?

We first consider the no-decay setting, $c(t) = 1$ in Eq. (10), and decompose the remainder using Eq. (19). Table 1 shows that the parallel remainder is comparable in magnitude to $m(T)$, whereas the orthogonal remainder is much smaller, with prediction–target cosine similarity of 0.983–0.987. Incorporating the fitted kernel decay reduces the parallel remainder from 72.4–215 to 3.2–22.2, while the orthogonal remainder changes by at most 1.5. Thus, the initial kernel predicts the response direction accurately but misestimates its magnitude; fitting kernel decay largely corrects this scale mismatch while preserving directional agreement.

To assess whether this pattern carries over to held-out response prediction, we substitute the fitted decay function in Eq. (26) into Eq. (18), fit the parameters on training examples, and compare predicted and observed log-odds change on test examples. Across nine model–dataset pairs in Table 2, the predictor achieves Pearson correlation 0.968 ± 0.017 and RMSE 0.386 ± 0.249. Without decay, Pearson correlation differs by only -0.001 to $+0.003$, but RMSE is $1.69\times$–$7.55\times$ higher. These results reinforce that kernel decay primarily reduces error in the predicted magnitude, while relative ordering remains well captured with or without decay. Appendix E gives the evaluation details.

4.4 How does kernel evolution affect training-parameter optimization?

Incorporating kernel decay preserves predictive trends while reducing magnitude error, but requires kernel computation at two additional epochs. Directly using the initial kernel yields high Pearson correlation despite substantially higher RMSE. This motivates a question: can such a trend-aligned but magnitude-biased approximation still guide effective parameter optimization? To test this, we study two PMP variants that differ only in the kernel trajectory. ***PG-NoDe*** sets $c(t) = 1$ and treats the kernel as constant, whereas ***PG-De*** uses the fitted decay function $c(t)$.

Limited additional benefit from magnitude calibration. Table 3 shows that PG-NoDe achieves the highest mean Acc on all three datasets and the second-lowest FT on two. These results suggest that a trend-aligned but magnitude-biased approximation can guide replay-weight optimization toward effective schedules. PG-De performs similarly (even with mean Acc 0.24–0.68 percentage points lower). Thus, calibrating the absolute prediction magnitudes provides limited additional benefit for

Table 2: Pearson correlations and RMSE between locally approximated and observed log-odds change. "Full" evaluates the predictable term $m(T)$ in Eq. (18); "w/o" removes the indicated factor. Green cells indicate degradation relative to Full, red cells indicate improvement.

Model	Variant	MMLU		MedQA		CSQA	
		Pearson↑	RMSE↓	Pearson↑	RMSE↓	Pearson↑	RMSE↓
Qwen 1.5B	Full	0.97 ±0.00	0.29 ±0.00	0.97 ±0.00	0.32 ±0.01	0.95 ±0.00	0.58 ±0.02
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	1.89 ±0.01	0.00 ±0.00	1.03 ±0.01	0.00 ±0.00	2.90 ±0.04
	w/o $H_{\mathcal{F}}$	0.97 ±0.00	1.03 ±0.00	0.96 ±0.00	0.37 ±0.01	0.96 ±0.00	1.92 ±0.03
	w/o $K_{\mathcal{F}F}$	0.29 ±0.01	1.72 ±0.00	0.70 ±0.03	0.84 ±0.01	0.33 ±0.04	2.79 ±0.04
	w/o $1 - p_{\mathcal{F}}$	0.94 ±0.00	0.37 ±0.01	0.96 ±0.00	0.48 ±0.01	0.90 ±0.01	0.66 ±0.04
	w/o decay	0.98 ±0.00	1.02 ±0.01	0.97 ±0.00	0.92 ±0.02	0.95 ±0.00	2.07 ±0.03
Qwen 7B	Full	0.98 ±0.00	0.47 ±0.02	0.98 ±0.00	0.30 ±0.01	0.95 ±0.01	0.94 ±0.19
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	3.13 ±0.03	0.00 ±0.00	2.12 ±0.06	0.00 ±0.00	5.21 ±0.06
	w/o $H_{\mathcal{F}}$	0.97 ±0.00	2.02 ±0.03	0.98 ±0.00	1.18 ±0.05	0.95 ±0.00	3.96 ±0.07
	w/o $K_{\mathcal{F}F}$	0.27 ±0.03	2.91 ±0.03	0.89 ±0.02	1.90 ±0.06	0.37 ±0.08	5.11 ±0.06
	w/o $1 - p_{\mathcal{F}}$	0.95 ±0.00	0.71 ±0.03	0.98 ±0.00	0.41 ±0.01	0.94 ±0.01	0.91 ±0.17
	w/o decay	0.98 ±0.00	3.02 ±0.03	0.98 ±0.00	2.29 ±0.09	0.95 ±0.00	3.43 ±0.18
Llama 8B	Full	0.98 ±0.01	0.16 ±0.03	0.94 ±0.01	0.25 ±0.01	0.99 ±0.00	0.16 ±0.00
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	0.92 ±0.01	0.00 ±0.00	0.38 ±0.01	0.00 ±0.00	0.99 ±0.02
	w/o $H_{\mathcal{F}}$	0.98 ±0.00	0.36 ±0.04	0.95 ±0.01	0.09 ±0.01	0.99 ±0.00	0.31 ±0.01
	w/o $K_{\mathcal{F}F}$	−0.26 ±0.08	0.97 ±0.04	−0.03 ±0.00	0.56 ±0.01	0.38 ±0.03	0.90 ±0.02
	w/o $1 - p_{\mathcal{F}}$	0.77 ±0.04	0.40 ±0.02	0.46 ±0.01	0.48 ±0.01	0.91 ±0.01	0.33 ±0.01
	w/o decay	0.98 ±0.01	0.57 ±0.07	0.94 ±0.01	0.43 ±0.01	0.99 ±0.00	0.84 ±0.02

Table 3: Acc and FT after fine-tuning for *Qwen 1.5B*. Best means are bold and second-best means are underlined. PG-NoDe achieves the highest Acc on all three datasets and the second-lowest FT on two. PG-NoDe and PG-De perform similarly, suggesting that better-calibrated prediction magnitudes offer limited additional benefit for replay control in this setting.

Method	MMLU		CSQA		MedQA	
	Acc↑	FT↓	Acc↑	FT↓	Acc↑	FT↓
GPM	52.19 ±0.96	30.73 ±1.37	68.71 ±0.96	16.68 ±0.69	35.80 ±1.05	48.74 ±0.83
OPGD	48.80 ±0.20	38.41 ±0.69	66.70 ±0.64	20.75 ±0.29	33.66 ±1.20	55.82 ±1.99
EWC	53.21 ±0.46	29.31 ±1.36	67.75 ±0.72	18.59 ±0.63	37.09 ±0.97	45.15 ±1.16
O-LoRA	57.61 ±0.49	**1.86** ±0.30	74.16 ±0.89	**0.56** ±0.02	43.21 ±1.15	**4.69** ±0.51
BLF	52.15 ±0.68	31.91 ±1.32	67.46 ±0.99	19.20 ±0.81	35.93 ±1.01	48.38 ±0.60
ER	54.84 ±0.99	26.09 ±1.08	69.81 ±1.11	15.57 ±0.99	38.58 ±1.22	41.00 ±1.12
ER + DWA	54.95 ±1.00	25.75 ±0.88	70.04 ±1.09	15.19 ±0.91	38.93 ±1.28	40.00 ±1.16
DER++	56.40 ±0.69	22.16 ±0.86	70.78 ±1.27	13.83 ±1.00	42.21 ±0.94	29.26 ±1.50
Smart Replay	57.69 ±0.51	16.74 ±0.98	72.00 ±1.49	11.31 ±1.41	44.31 ±0.86	14.65 ±1.89
PG-NoDe	**59.81** ±0.26	<u>9.40</u> ±1.72	**77.09** ±1.64	<u>5.14</u> ±1.17	**46.27** ±0.92	11.28 ±0.72
PG-De	<u>59.13</u> ±0.56	10.47 ±0.79	<u>76.52</u> ±1.78	7.62 ±2.22	<u>46.03</u> ±1.34	<u>10.98</u> ±0.59

PMP replay-weight optimization in this setting. Implementation and evaluation details are given in Appendix F.

Analysis of the two approximation methods. The schedules optimized by PG-NoDe and PG-De yield predicted terminal CE values within 0.85%–1.61% of each other when evaluated under the same fitted-decay objective (Table 4). Their objective gradients with respect to replay weights have cosine similarities of 0.816–0.979, indicating aligned local optimization directions. Appendix G bounds the objective gap between the two approximations' optimal schedules. The bound depends

Table 4: Terminal-CE gap under the fitted-decay objective, initial gradient alignment, and empirical local affine bound. The small terminal-CE gaps, aligned gradients, and small empirical local bounds suggest that omitting decay has limited impact on replay-weight selection in this setting.

Dataset	Terminal-CE gap (%)	Gradient cosine	Empirical affine bound (%)
MMLU	$1.20_{\pm 0.05}$	$0.939_{\pm 0.002}$	$1.44_{\pm 0.06}$
MedQA	$0.85_{\pm 0.07}$	$0.979_{\pm 0.001}$	$1.25_{\pm 0.08}$
CSQA	$1.61_{\pm 0.74}$	$0.816_{\pm 0.015}$	$1.76_{\pm 0.85}$

on the non-affine residual relative to the objective scale. Positive rescaling and constant shifts alone leave the optimum unchanged. The empirical local estimate of this relative bound is 1.25%–1.76% on sampled schedules. Together, these results suggest that the two approximations yield similar PMP optimization outcomes.

Time-to-quality. Although PG-NoDe requires an initial gradient-kernel computation, it quickly surpasses Smart Replay's highest mean Acc over its full run. Its total elapsed time, including kernel setup, is only 44%–76% of Smart Replay's full training time (Table 7). PG-De requires two additional gradient-kernel computations, substantially increasing setup time while yielding similar endpoint performance. These results further support using the less costly, trend-aligned but magnitude-biased approximation for replay-weight optimization.

Parameter sensitivity. Varying PG-Replay's terminal-CE tolerance from 1% to 10% yields within-dataset ranges of 0.22–0.55 points for Acc and 0.18–2.36 for FT (Table 6).

5 Conclusion

Kernel decay primarily calibrates the magnitude of forgetting predictions rather than redefining their relative response pattern. Accounting for decay therefore reduces prediction RMSE. In our experimental setting, the relative ordering retained by PG-NoDe provides useful information for replay-weight optimization, while further calibration of prediction magnitudes offers limited additional benefit. Consistent with this observation, PG-NoDe yields the highest mean endpoint accuracy on all three datasets, while fitting decay after two warm-up epochs improves prediction RMSE but does not improve replay control in these experiments.

Limitations. PG-NoDe avoids the two additional rounds of kernel computation required by PG-De, but still incurs the cost of constructing the initial kernel. Future work could explore efficient subset sampling to estimate kernel properties across the full dataset from fewer examples, further reducing this computational overhead. Additionally, the regret bound in Appendix G compares ideal minimizers of the terminal-CE objectives over a common schedule region and does not account for the CE-tolerance constraint used by PG-Replay. Extending the analysis to this constrained formulation remains future work.

AI USE STATEMENT

We used generative AI tools to assist with drafting and revising manuscript text, clarifying mathematical derivations, interpreting reported results, LaTeX formatting, and figure preparation. These tools were not used to generate experimental data or results. All authors reviewed and revised the manuscript and its AI-assisted content and take responsibility for the final work.

ETHICS STATEMENT

This study uses publicly available benchmarks and pretrained models to investigate forgetting during full fine-tuning of large language models. It does not involve human participants, collection of private or personally identifiable data, or deployment in real-world decision-making systems. Dataset and model reuse should comply with their respective licenses and terms of use.

REPRODUCIBILITY STATEMENT

The appendices describe the optimization formulation, training configuration, kernel estimation and scale fitting, response-prediction protocol, baseline implementations, and theoretical analysis. Appendix A details the constrained replay-weight optimization; Appendices B–F provide experimental protocols; and Appendix G presents the gradient comparison, empirical affine-bound estimation, and schedule cross-evaluation.

REFERENCES

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html.

Jianting Chen, Dianzhi Yu, and Irwin King. Smart replay: Adaptive scheduling of memory rehearsal for computational resource-aware incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Chen_Smart_Replay_Adaptive_Scheduling_of_Memory_Rehearsal_for_Computational_Resource-Aware_CVPR_2026_paper.pdf.

Sirui Chen, Yunzhe Qi, Mengting Ai, Yifan Sun, Ruizhong Qiu, Jiaru Zou, and Jingrui He. Influence-preserving proxies for gradient-based data selection in LLM fine-tuning. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2602.17835.

Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A theoretical analysis of catastrophic forgetting through the NTK overlap matrix. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pp. 1072–1080. PMLR, 2021. URL https://proceedings.mlr.press/v130/doan21a.html.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data selection via optimal control for language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9ad4891facabf17aa11580686bacfe4e-Abstract-Conference.html.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.03300.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1806.07572.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*, 2020. URL https://arxiv.org/abs/2009.13081.

Xisen Jin and Xiang Ren. What will my model forget? forecasting forgotten examples in language model refinement. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 22145–22159. PMLR, 2024. URL https://proceedings.mlr.press/v235/jin24d.html.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:1612.00796*, 2016. URL https://arxiv.org/abs/1612.00796.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4297–4308. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-emnlp.249/.

Jingren Liu, Zhong Ji, YunLong Yu, Jiale Cao, Yanwei Pang, Jungong Han, and Xuelong Li. Parameter-efficient fine-tuning for continual learning: A neural tangent kernel perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(7):8297–8314, 2026. doi: 10.1109/TPAMI.2026.3670952.

Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Liu_End-To-End_Multi-Task_Learning_With_Attention_CVPR_2019_paper.html.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 23610–23641. PMLR, 2023. URL https://proceedings.mlr.press/v202/malladi23a.html.

Meta. Llama 3.2 model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3AOj0RCNC2.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL-HLT*, pp. 4149–4158, 2019. URL https://aclanthology.org/N19-1421/.

Team GLM. ChatGLM: A family of large language models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793*, 2024. URL https://arxiv.org/abs/2406.12793.

Chenxu Wang, Yilin Lyu, Zicheng Sun, and Liping Jing. Continual gradient low-rank projection fine-tuning for LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 14815–14829. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.721/.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023. doi: 10.18653/v1/2023.findings-emnlp.715.

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. In *Proceedings of NAACL-HLT*, pp. 663–677. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.naacl-long.37/.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 54104–54132. PMLR, 2024. URL https://proceedings.mlr.press/v235/xia24c.html.

Mutian Yang, Jiandong Gao, and Ji Wu. Decoupling knowledge and reasoning in LLMs: An exploration using cognitive dual-system theory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34268–34276, 2026a. doi: 10.1609/aaai.v40i40.40723. URL https://ojs.aaai.org/index.php/AAAI/article/view/40723.

Mutian Yang, Zisen Zhan, Yutong Chen, Haolin Li, Kaiwen Wang, Kaili Zheng, Yuguang Wang, Qi Wang, Jiandong Gao, and Ji Wu. Collaborative parameter learning: Mitigating forgetting via parameter-level gradient analysis. *arXiv preprint arXiv:2601.21577*, 2026b. doi: 10.48550/arXiv.2601.21577. URL https://arxiv.org/abs/2601.21577.

Xinxin You, Qixin Sun, Chenwei Yan, Xiao Zhang, Chen Ning, Xiangling Fu, Si Liu, Guoping Hu, Shijin Wang, Ji Wu, and Xien Liu. FACT: Mitigating inconsistent hallucinations in LLMs via fact-driven alternating code-text training. In *Advances in Neural Information Processing Systems*, volume 38, pp. 144079–144103, 2025. doi: 10.52202/085713-4327. URL https://papers.nips.cc/paper_files/paper/2025/hash/bc75254bc4b8b42f401d0ab5d6e9aa4b-Abstract-Conference.html.

Guodong Zheng, Peng Wang, Shengchao Hu, Quan Zheng, and Li Shen. Understanding the dynamics of forgetting and generalization in continual learning via the neural tangent kernel. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1f8638a87bddd34e090698f0bfaa9c02-Abstract-Conference.html.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a774503daed55eb53c634847ae071ec7-Abstract-Conference.html.

Yuxuan Zhou, Xien Liu, Xiao Zhang, Chen Ning, Shijin Wang, Guoping Hu, and Ji Wu. Investigating and mitigating catastrophic forgetting in medical knowledge injection through internal knowledge augmentation learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0308. URL https://papers.neurips.cc/paper_files/paper/2025/hash/0d2c50691b832d8e8901715998e7c22c-Abstract-Conference.html.

A REGULARIZED PG-REPLAY

Let $w = (w_1, \ldots, w_T)$ denote the nonnegative epoch-wise replay weights and $\Psi(s_T(w))$ the predicted terminal CE. We first obtain a low-loss reference Ψ_{ref} by minimizing $\Psi(s_T(w))$ over $w \geq 0$. To favor smaller replay weights while allowing a relative terminal-loss tolerance δ, we then solve

$$\min_{w \geq 0} \sum_{e=1}^{T} w_e^2 \quad \text{subject to} \quad \Psi(s_T(w)) \leq (1+\delta)\Psi_{\mathrm{ref}}. \tag{27}$$

This constrained form expresses the same local trade-off as a quadratic regularizer: with multiplier $\mu \geq 0$, its Lagrangian is

$$\mathcal{L}(w, \mu) = \sum_{e=1}^{T} w_e^2 + \mu[\Psi(s_T(w)) - (1+\delta)\Psi_{\mathrm{ref}}]\,. \tag{28}$$

For an interior epoch weight, stationarity gives $2w_e + \mu\, \partial\Psi(s_T)/\partial w_e = 0$; for $w_e = 0$, the corresponding one-sided derivative is nonnegative. The adjoint gradient in Eq. (24) provides $\partial\Psi(s_T)/\partial w_e$, while the multiplier is determined by the terminal-loss constraint and complementary slackness. We solve the reference search with L-BFGS-B and the constrained problem with sequential least-squares quadratic programming (SLSQP), both using the adjoint gradient.

B COMMON TRAINING CONFIGURATION

All training runs use SGD for 10 epochs in FP32, with a fixed learning rate of 5×10^{-7} and batch size 8. Inference uses temperature 0 (greedy decoding). The replay-baseline runs used NVIDIA A800-SXM4-80GB GPUs with Python 3.10, PyTorch 2.10.0+cu128, and Transformers 4.57.6. Score-space replay planning uses NumPy and Numba for trajectory evaluation and SciPy for constrained optimization; figures use Matplotlib.

C KERNEL MEASUREMENTS

Motivated by the Johnson–Lindenstrauss lemma, we represent per-example log-odds gradients using 8192-dimensional sketches for approximate kernel estimation. For ***Qwen 1.5B***, we use a shared sketch mapping across examples and checkpoints, and estimate the cross-group and fine-tuning kernel blocks from sketch inner products. Figure 1 reports the resulting cross-group kernel's entrywise Pearson correlation with its initial value and its relative Frobenius norm over training.

D KERNEL-SCALE FITTING

We estimate the shared scale directly from the kernel matrices. At each observed checkpoint, the joint Frobenius least-squares amplitude over the cross-group and fine-tuning blocks is

$$\begin{aligned} \widetilde{c}(t) &= \arg\min_{c} \sum_{X \in \{\mathcal{M}, \mathcal{F}\}} \|K_{X\mathcal{F}}(t) - cK_{X\mathcal{F}}(0)\|_F^2 \\ &= \frac{\sum_{X \in \{\mathcal{M}, \mathcal{F}\}} \langle K_{X\mathcal{F}}(0), K_{X\mathcal{F}}(t)\rangle_F}{\sum_{X \in \{\mathcal{M}, \mathcal{F}\}} \|K_{X\mathcal{F}}(0)\|_F^2}. \end{aligned} \tag{29}$$

We fit the positive power-law parameters using the amplitudes at checkpoints 1–2:

$$(\widehat{\beta}, \widehat{\gamma}) = \arg\min_{\beta, \gamma > 0} \sum_{t \in \{1,2\}} \left[\log \widetilde{c}(t) + \gamma \log(1 + \beta t)\right]^2. \tag{30}$$

The logarithmic residual gives relative deviations comparable weight at the two checkpoints. For the visual diagnostic in Figure 1, we apply the same two-checkpoint fit to the displayed cross-kernel norm and compute RMSE only over the extrapolation checkpoints.

E Response Prediction

Our response-prediction protocol separates fitting from evaluation along both the sample and time axes. We obtain the early amplitudes from Eq. (29), fit their decay with Eq. (30), and calibrate ρ from the early training-score change. We evaluate the predictable term $m(T)$ in Eq. (18) from checkpoint 2 using its scores and gradient sketches, with $c(t) = ((1 + \beta t)/(1 + 2\beta))^{-\gamma}$ and the integral taken from 2 to T. No test example or post-checkpoint-2 state enters parameter fitting.

Table 5 extends the factor ablation to the six models not reported in the main-text table. Each run fits ρ, β, γ using states 0–2 and evaluates predicted mastered-example drift over epochs 3–10. We set the inverse-only damping to $\mu = 10^{-4}\lambda_{\max}(H_{\mathcal{F}}^{1/2} K_{\mathcal{F}\mathcal{F}} H_{\mathcal{F}}^{1/2})$.

Ablation definitions. In Tables 2 and 5, "w/o $K_{\mathcal{M}\mathcal{F}}$" and "w/o $K_{\mathcal{F}\mathcal{F}}$" replace the corresponding kernel block with an all-ones matrix of the same shape. "w/o $H_{\mathcal{F}}$" replaces the curvature matrix with the identity. "w/o $1 - p_{\mathcal{F}}$" replaces the initial residual vector with an all-ones vector, and "w/o D" replaces the finite-time response factor with the identity. "w/o decay" sets $c(t) = 1$ over the prediction interval.

F Replay-Control Comparisons

Training protocol. Within each training partition, we uniformly sample 10% of the initially correct training examples, $\mathcal{M}_{\text{train}}$, to form the replay buffer and hold this buffer fixed across epochs. All methods use the same initially incorrect training set; methods that require stored or reference examples draw them from this buffer. PG-Replay uses a 3% terminal-CE tolerance by default; Table 6 varies this value.

Baseline implementations. For the projection baselines, GPM forms an orthonormal basis U from replay-example activations and projects each affected weight gradient G as $G \leftarrow G(I - UU^\top)$ (Saha et al., 2021). OPGD instead forms U from mastered-example output gradients and projects the current gradient as $g \leftarrow (I - UU^\top)g$ (Zheng et al., 2026). Thus the methods differ in which preserved subspace defines the projection.

EWC estimates a diagonal empirical Fisher $F_j = |R|^{-1} \sum_{i \in R} (\partial \ell_i / \partial \theta_j)^2$ on the replay buffer and penalizes movement from the initial parameters θ_0 (Kirkpatrick et al., 2016):

$$\mathcal{L}_{\text{EWC}}(\theta) = \mathcal{L}_{\mathcal{F}}(\theta) + \frac{\lambda_{\text{EWC}}}{2} \sum_j F_j (\theta_j - \theta_{0,j})^2. \tag{31}$$

O-LoRA freezes the base matrix and writes $W = W_0 + B_{\mathcal{M}} A_{\mathcal{M}} + B_{\mathcal{F}} A_{\mathcal{F}}$, discouraging overlap between old and new low-rank adapters through $\|A_{\mathcal{M}} A_{\mathcal{F}}^\top\|_1$ (Wang et al., 2023). BLF instead freezes the lower-layer parameters, so $\nabla_{\theta_{\text{lower}}} \mathcal{L} = 0$ and only upper layers are updated (Zheng et al., 2025).

ER + DWA makes the two group weights depend on their recent CE ratios (Liu et al., 2019). Let $\mathcal{L}_h(e)$ be the mean CE for group $h \in \{\mathcal{F}, R\}$ at epoch e; the DWA softmax rule is

$$a_h(e) = \frac{\mathcal{L}_h(e-1)}{\mathcal{L}_h(e-2)}, \qquad w_h(e) = \frac{2 \exp(a_h(e)/\tau_{\text{DWA}})}{\sum_{h' \in \{\mathcal{F}, R\}} \exp(a_{h'}(e)/\tau_{\text{DWA}})}. \tag{32}$$

DER++ combines CE on new and replay examples with matching the replay examples' stored initial-model logits z_0 (Buzzega et al., 2020):

$$\mathcal{L}_{\text{DER++}}(\theta) = \frac{1}{|\mathcal{B}|} \left[\sum_{i \in \mathcal{F}} \ell_i(\theta) + \sum_{i \in R} \big(\beta \ell_i(\theta) + \alpha \, \text{MSE}(z_\theta(x_i), z_0(x_i)) \big) \right]. \tag{33}$$

Here MSE averages the squared logit difference over the vocabulary, and α, β weight the two replay terms. Smart Replay dynamically schedules the replay ratio across minibatches from recent learning and retention signals under a computational budget (Chen et al., 2026a).

G Gradient Alignment and Schedule Cross-Evaluation

Let $J_0(w)$ and $J_d(w)$ denote the terminal-CE objectives predicted without and with fitted kernel decay, respectively, where w is the replay schedule. Table 4 computes their gradient cosine at the

Table 5: Pearson correlations and RMSE for component ablations on six additional models. Green cells indicate degradation relative to Full, and red cells indicate improvement.

		MMLU		MedQA		CSQA	
		Pearson↑	RMSE↓	Pearson↑	RMSE↓	Pearson↑	RMSE↓
Llama 3B	Full	0.99 ±0.00	0.12 ±0.00	1.00 ±0.00	0.11 ±0.00	0.99 ±0.00	0.09 ±0.00
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	0.81 ±0.02	0.00 ±0.00	0.63 ±0.01	0.00 ±0.00	1.14 ±0.02
	w/o $H_{\mathcal{F}}$	0.99 ±0.00	0.26 ±0.01	1.00 ±0.00	0.11 ±0.00	0.99 ±0.00	0.46 ±0.01
	w/o $K_{\mathcal{F}F}$	0.30 ±0.02	0.72 ±0.01	0.77 ±0.02	0.50 ±0.01	0.58 ±0.02	0.97 ±0.02
	w/o D	0.69 ±0.14	43.7 ±11.4	0.89 ±0.01	13.1 ±0.3	0.77 ±0.01	38.4 ±1.5
	w/o $1 - p_{\mathcal{F}}$	0.89 ±0.01	0.32 ±0.00	0.99 ±0.00	0.19 ±0.00	0.94 ±0.00	0.27 ±0.00
	w/o decay	0.99 ±0.00	0.44 ±0.01	1.00 ±0.00	0.35 ±0.01	0.99 ±0.00	0.64 ±0.01
Llama 1B	Full	0.99 ±0.00	0.10 ±0.02	1.00 ±0.00	0.03 ±0.00	0.93 ±0.06	0.54 ±0.42
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	0.79 ±0.06	0.00 ±0.00	0.51 ±0.07	0.00 ±0.00	2.03 ±1.22
	w/o $H_{\mathcal{F}}$	0.98 ±0.01	0.37 ±0.02	1.00 ±0.00	0.13 ±0.02	0.95 ±0.04	1.42 ±1.03
	w/o $K_{\mathcal{F}F}$	0.90 ±0.03	0.78 ±0.06	0.98 ±0.00	0.49 ±0.06	0.79 ±0.16	2.02 ±1.22
	w/o D	0.51 ±0.10	330 ±145	0.45 ±0.06	12.9 ±2.0	0.49 ±0.09	$(3.8 \pm 3.2) \times 10^4$
	w/o $1 - p_{\mathcal{F}}$	0.98 ±0.00	0.15 ±0.01	1.00 ±0.00	0.09 ±0.00	0.93 ±0.06	0.56 ±0.38
	w/o decay	0.99 ±0.00	0.23 ±0.01	1.00 ±0.00	0.08 ±0.02	0.93 ±0.06	0.79 ±0.53
Mistral 7B	Full	0.99 ±0.00	0.12 ±0.04	0.89 ±0.01	0.24 ±0.02	0.92 ±0.01	0.38 ±0.02
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	0.54 ±0.01	0.00 ±0.00	0.81 ±0.04	0.00 ±0.00	0.71 ±0.06
	w/o $H_{\mathcal{F}}$	0.99 ±0.00	0.26 ±0.02	0.95 ±0.01	0.43 ±0.03	0.88 ±0.01	0.39 ±0.03
	w/o $K_{\mathcal{F}F}$	0.97 ±0.01	0.54 ±0.01	0.31 ±0.11	0.79 ±0.04	0.69 ±0.04	0.68 ±0.06
	w/o D	−0.01 ±0.09	52.8 ±8.0	0.62 ±0.01	20.4 ±3.6	0.35 ±0.08	99.3 ±27.5
	w/o $1 - p_{\mathcal{F}}$	0.99 ±0.00	0.12 ±0.01	0.78 ±0.01	0.32 ±0.03	0.90 ±0.01	0.67 ±0.03
	w/o decay	0.99 ±0.00	0.17 ±0.01	0.90 ±0.01	0.32 ±0.03	0.92 ±0.01	0.62 ±0.04
Gemma 2B	Full	0.92 ±0.01	0.53 ±0.08	0.91 ±0.01	0.37 ±0.02	0.94 ±0.01	0.87 ±0.05
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	1.69 ±0.03	0.00 ±0.00	1.53 ±0.04	0.00 ±0.00	2.46 ±0.07
	w/o $H_{\mathcal{F}}$	0.94 ±0.00	1.06 ±0.01	0.92 ±0.01	0.89 ±0.03	0.73 ±0.03	2.06 ±0.07
	w/o $K_{\mathcal{F}F}$	0.00 ±0.08	1.68 ±0.02	0.35 ±0.30	1.48 ±0.04	0.87 ±0.00	2.19 ±0.06
	w/o D	0.79 ±0.05	118 ±35	0.76 ±0.04	43.1 ±6.9	−0.76 ±0.03	84.5 ±15.2
	w/o $1 - p_{\mathcal{F}}$	0.71 ±0.05	0.66 ±0.08	0.87 ±0.04	0.39 ±0.06	0.91 ±0.00	2.27 ±0.07
	w/o decay	0.93 ±0.01	0.63 ±0.08	0.91 ±0.01	0.99 ±0.03	0.94 ±0.01	0.87 ±0.05
Gemma 9B	Full	0.93 ±0.01	0.46 ±0.03	0.89 ±0.01	0.56 ±0.08	1.00 ±0.00	0.01 ±0.00
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	1.83 ±0.03	0.00 ±0.00	1.63 ±0.04	0.00 ±0.00	0.45 ±0.00
	w/o $H_{\mathcal{F}}$	0.93 ±0.01	1.14 ±0.03	0.90 ±0.01	1.05 ±0.03	1.00 ±0.00	0.11 ±0.00
	w/o $K_{\mathcal{F}F}$	−0.05 ±0.14	1.81 ±0.04	0.32 ±0.17	1.63 ±0.04	0.99 ±0.00	0.44 ±0.00
	w/o D	0.78 ±0.01	171 ±33	0.44 ±0.02	334 ±4	0.17 ±0.11	$(3.4 \pm 0.3) \times 10^4$
	w/o $1 - p_{\mathcal{F}}$	0.70 ±0.08	0.63 ±0.08	0.86 ±0.01	0.55 ±0.07	1.00 ±0.00	0.01 ±0.00
	w/o decay	0.94 ±0.01	0.89 ±0.05	0.89 ±0.01	0.47 ±0.03	1.00 ±0.00	0.02 ±0.00
GLM 9B	Full	0.91 ±0.00	0.34 ±0.01	0.84 ±0.01	0.53 ±0.06	0.79 ±0.02	0.67 ±0.13
	w/o $K_{\mathcal{M}F}$	0.00 ±0.00	0.50 ±0.02	0.00 ±0.00	1.13 ±0.08	0.00 ±0.00	5.01 ±0.24
	w/o $H_{\mathcal{F}}$	0.91 ±0.00	0.35 ±0.02	0.84 ±0.01	0.53 ±0.04	0.81 ±0.02	4.21 ±0.23
	w/o $K_{\mathcal{F}F}$	0.89 ±0.01	0.49 ±0.02	0.84 ±0.01	1.12 ±0.08	0.70 ±0.01	5.00 ±0.24
	w/o D	0.75 ±0.04	26.2 ±2.5	0.49 ±0.08	9.89 ±0.87	0.50 ±0.03	$(1.3 \pm 0.6) \times 10^5$
	w/o $1 - p_{\mathcal{F}}$	0.91 ±0.01	0.31 ±0.01	0.84 ±0.01	0.70 ±0.08	0.79 ±0.02	0.67 ±0.13
	w/o decay	0.91 ±0.00	0.46 ±0.02	0.84 ±0.01	2.39 ±0.23	0.79 ±0.02	3.13 ±0.17

zero schedule as

$$g_0 = \nabla_w J_0(w)\big|_{w=0}, \qquad g_d = \nabla_w J_d(w)\big|_{w=0}, \qquad \cos(g_0, g_d) = \frac{g_0^\top g_d}{\|g_0\|_2 \|g_d\|_2}. \tag{34}$$

Table 6: PG-Replay sensitivity to terminal-CE tolerance; the default 3% setting is highlighted. Varying the tolerance from 1% to 10% changes Acc by at most 0.55 percentage points within each dataset.

CE tolerance	MMLU		CSQA		MedQA	
	Acc↑	FT↓	Acc↑	FT↓	Acc↑	FT↓
1%	59.81 ±0.35	9.55 ±1.73	77.18 ±1.75	5.18 ±1.06	46.07 ±0.95	12.39 ±1.24
3%	59.81 ±0.26	9.40 ±1.72	77.09 ±1.64	5.14 ±1.17	46.27 ±0.92	11.28 ±0.72
5%	59.63 ±0.21	9.72 ±1.61	76.91 ±1.70	5.03 ±1.38	46.29 ±0.90	10.84 ±0.65
10%	59.56 ±0.32	9.93 ±1.40	76.63 ±1.45	5.00 ±1.45	46.16 ±1.21	10.03 ±1.00

Table 7: Time-to-quality with *Qwen 1.5B*. **Setup** is the time spent on kernel preparation before replay optimization. **First exceeds** is the cumulative time, including setup, to surpass Smart Replay's highest mean Acc over its full run. **Completion** is the total time, including setup, to complete the full training.

Dataset	Method	Setup (h)	First exceeds (h)	Completion (h)
MMLU	Smart Replay	–	–	2.304 ±0.004
	PG-NoDe	0.76 ±0.01	1.01 ±0.01	3.29 ±0.01
	PG-De	2.29 ±0.03	2.53 ±0.03	4.81 ±0.03
MedQA	Smart Replay	–	–	2.865 ±0.011
	PG-NoDe	0.92 ±0.02	1.57 ±0.02	4.17 ±0.02
	PG-De	2.76 ±0.06	3.41 ±0.06	6.01 ±0.06
CSQA	Smart Replay	–	–	0.737 ±0.007
	PG-NoDe	0.48 ±0.02	0.56 ±0.02	1.30 ±0.02
	PG-De	1.43 ±0.06	1.52 ±0.06	2.26 ±0.06

The effect of their scale difference can also be bounded under a local affine approximation. Let $\mathcal{W}$ be the tested schedule region and suppose

$$J_0(w) = c_J J_d(w) + b + r(w), \qquad c_J > 0, \qquad \sup_{w \in \mathcal{W}} |r(w)| \le \delta_{\text{aff}}. \tag{35}$$

For the ideal minimizers of the terminal-CE objectives over the same region, without the CE-tolerance constraint in Appendix A,

$$w_0^\star = \arg\min_{w \in \mathcal{W}} J_0(w), \qquad w_d^\star = \arg\min_{w \in \mathcal{W}} J_d(w), \tag{36}$$

the optimality of $w_0^\star$ gives

$$c_J[J_d(w_0^\star) - J_d(w_d^\star)] \le r(w_d^\star) - r(w_0^\star) \le 2\delta_{\text{aff}}. \tag{37}$$

Since $w_d^\star$ minimizes J_d, the fitted-decay regret satisfies

$$0 \le J_d(w_0^\star) - J_d(w_d^\star) \le \frac{2\delta_{\text{aff}}}{c_J}. \tag{38}$$

Thus a positive affine scale does not itself change the preferred schedule. The possible objective loss is controlled by the non-affine residual relative to that scale, δ_{aff}/c_J; after normalization, the relative bound is controlled by $\delta_{\text{aff}}/[c_J J_d(w_d)]$.

We estimate this bound locally for each data partition using 100 schedules in a coordinate-wise box containing the two constrained schedules. Sixty Latin-hypercube samples fit c_J and b, while 38 held-out samples together with the two constrained schedules estimate δ_{aff}. Table 4 reports the bound relative to the fitted-decay constrained objective,

$$B_{\text{aff}} = 100\,\frac{2\delta_{\text{aff}}}{c_J J_d(w_d)}. \tag{39}$$

Because δ_{aff} is the maximum residual on the sampled audit schedules, B_{aff} is an empirical local bound rather than a certificate over the continuous schedule space.

We apply the same constrained optimizer in Section A to J_0 and J_d, obtaining schedules w_0 and w_d. Both schedules are then evaluated under the fitted-decay objective, and the reported terminal-CE gap is

$$\Delta_{\mathrm{CE}} = 100\,\frac{J_d(w_0) - J_d(w_d)}{J_d(w_d)}. \tag{40}$$

Positive values favor w_d, whereas negative values favor w_0 under the same fitted-decay rollout.