UNDERSTANDING AND MITIGATING FORGETTING IN LLM FULL FINE-TUNING AT THE PARAMETER LEVEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Full fine-tuning enables large language models to acquire new knowledge but can cause them to forget previously mastered knowledge. Existing gradient-projection methods constrain overall update directions without explicitly identifying which parameter updates contribute to forgetting. We decompose gradient similarity into parameter-wise contributions to distinguish Conflicting Parameters, whose updates increase the loss on mastered examples to first order, from Collaborative Parameters, whose updates do not. Based on this analysis, we propose Collaborative Parameter Learning (CPL), which uses a gradient from reference examples to block conflicting updates while retaining collaborative ones. Experiments span five models and six datasets across three modalities. Across held-out evaluations, CPL compares favorably with existing forgetting-mitigation baselines, achieving a strong balance between accuracy and knowledge retention. Further analyses suggest that CPL's strong performance is likely related to the recovery of previously mastered answers and greater preservation of upper-layer representation structure.

1 INTRODUCTION

Large language models (LLMs) have shown strong knowledge capacity across a broad range of domains (Qwen et al., 2024; Grattafiori et al., 2024; Yang et al., 2026). Full fine-tuning enables LLMs to acquire information and capabilities beyond their initial performance (You et al., 2025). However, learning new knowledge can also cause the forgetting of previously mastered knowledge. This phenomenon, known as catastrophic forgetting, undermines the preservation of previously acquired capabilities during model adaptation (Kirkpatrick et al., 2017; Kotha et al., 2024).

Prior work mainly explains forgetting through changes in model outputs (Zhang & Wu, 2024; Kotha et al., 2024), drift in internal representations (Merchant et al., 2020; Kornblith et al., 2019), parameter importance (Kirkpatrick et al., 2017), and loss-landscape geometry (Li et al., 2024). Beyond these views, gradient-based analysis is another major perspective because it directly connects the fine-tuning objective to parameter updates and their effect on previously mastered knowledge (Doan et al., 2021). Existing mitigation methods restrict trainable parameters or update subspaces (Wang et al., 2023; Liu et al., 2026), project gradients using reference information (Yu et al., 2020; Liu et al., 2021), regularize parameter or output changes (Kirkpatrick et al., 2017; Li & Hoiem, 2016), or replay stored examples (Robins, 1995).

However, existing gradient-based analyses and constraints operate mainly at the level of gradient vectors or protected subspaces. Aggregating interactions at this level hides the fact that different parameters within the same update can have opposite effects on mastered knowledge. Consequently, projection can remove a component that causes forgetting while also perturbing parameter updates that would benefit previously mastered knowledge. This motivates a finer-grained question: which individual parameters adversely affect or help preserve mastered knowledge?

Motivated by this gap, we move from vector level to parameter level as follows:

• **Adam-aware gradient analysis.** We analyze forgetting through the interaction between each mastered example's gradient and Adam's proposed update direction. Samples with strongly negative interactions exhibit an average forgetting rate of 49.0%, compared with 3.0% for their weakly negative counterparts. This consistent gap identifies strongly negative Adam-aware interaction as an indicator of susceptibility to forgetting.

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.

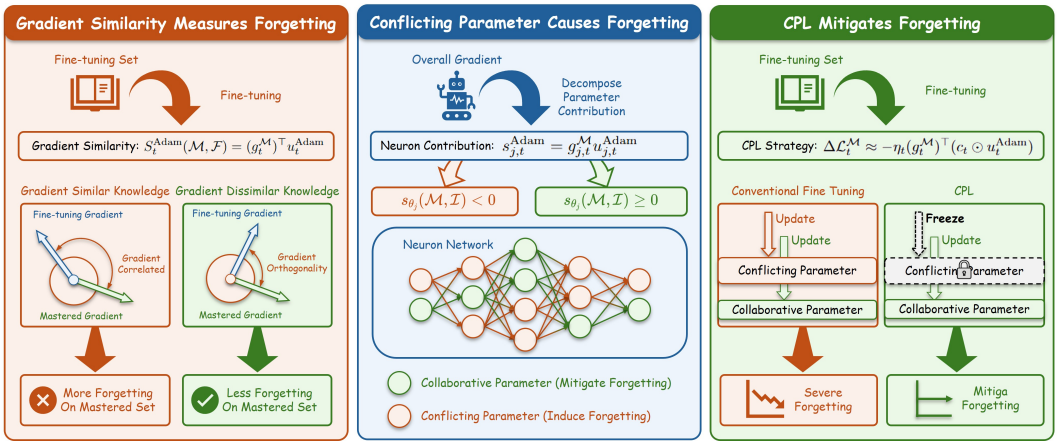

Figure 1: Conceptual illustration of our framework.

• **Conflicting and Collaborative Parameters.** We introduce a parameter-wise decomposition that distinguishes Conflicting Parameters, whose proposed updates promote forgetting, from Collaborative Parameters, whose updates help preserve mastered knowledge. Across five models, six datasets, and three modalities, Collaborative and Conflicting Parameters account for 44.1% and 55.9% of coordinates on average, respectively, showing that opposite parameter-level effects coexist within an aggregate conflicting direction.

• **Collaborative Parameter Learning.** Based on this decomposition, we propose CPL, which retains collaborative Adam updates and blocks conflicting ones using a training-set reference. CPL achieves the highest mean accuracy on all four text datasets, the lowest forgetting rate on CSQA, and the second-lowest forgetting rate on MMLU, MedQA, and ARC-C. Mechanism analyses associate these gains with stronger answer recovery and greater preservation of upper-layer representations.

2 Related Work

Understanding forgetting during full fine-tuning. Full fine-tuning keeps every model parameter trainable, so the update can alter predictions, internal features, and the parameters that support previously mastered behavior. Output-based studies characterize forgetting through changes in predictions or token probabilities (Zhang & Wu, 2024; Kotha et al., 2024). Representation-based analyses instead examine how fine-tuning reorganizes hidden features across layers (Merchant et al., 2020; Kornblith et al., 2019), while parameter- and landscape-based accounts study sensitivity to particular weights and the geometry around the adapted solution (Kirkpatrick et al., 2017; Li et al., 2024).

Mitigating forgetting during model adaptation. Forgetting-mitigation methods intervene at different points in optimization. Restricted-update approaches define which parameters or subspaces may change (Wang et al., 2023; Liu et al., 2026). Projection approaches protect stored gradients or learned subspaces by modifying the proposed update (Chaudhry et al., 2019; Saha et al., 2021; Farajtabar et al., 2020; Zheng et al., 2026). Regularization modifies the objective by penalizing parameter or output changes (Kirkpatrick et al., 2017; Li & Hoiem, 2016), while replay changes the training data so that stored examples shape the resulting gradient (Robins, 1995).

3 Methods

3.1 Definition of Forgetting

Throughout the remainder of this paper, fine-tuning refers to full fine-tuning. Following the setup of prior studies (Jin & Ren, 2024; Zhou et al., 2025), we first ask the model to answer every question before fine-tuning. Questions answered correctly form the **Mastered Set** $\mathcal{M}$, while those answered

incorrectly form the **Fine-Tuning Set** $\mathcal{F}$:

$$\mathcal{F} = \{(x_i^{\mathcal{F}}, y_i^{\mathcal{F}})\}_{i=1}^{|\mathcal{F}|}, \qquad \mathcal{M} = \{(x_i^{\mathcal{M}}, y_i^{\mathcal{M}})\}_{i=1}^{|\mathcal{M}|}. \tag{1}$$

Our goal is to fine-tune on $\mathcal{F}$ and study the change on $\mathcal{M}$. Let θ_0 and θ_T denote the parameters before and after fine-tuning, respectively, and let $\widehat{y}_\theta(x)$ denote the model's predicted answer. The average cross-entropy losses on the two sets are

$$\mathcal{L}^{\mathcal{F}}(\theta) = \frac{1}{|\mathcal{F}|} \sum_{(x,y)\in\mathcal{F}} \ell(f_\theta(x), y), \qquad \mathcal{L}^{\mathcal{M}}(\theta) = \frac{1}{|\mathcal{M}|} \sum_{(x,y)\in\mathcal{M}} \ell(f_\theta(x), y). \tag{2}$$

At step t, the raw fine-tuning gradient is

$$g_t^{\mathcal{F}} = \nabla_\theta \mathcal{L}^{\mathcal{F}}(\theta_t). \tag{3}$$

Adam (Kingma & Ba, 2015) transforms this gradient using exponential moving averages of its first and second moments:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t^{\mathcal{F}}, & v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^{\mathcal{F}} \odot g_t^{\mathcal{F}}, \\ \widehat{m}_t &= m_t/(1-\beta_1^t), & \widehat{v}_t &= v_t/(1-\beta_2^t). \end{aligned} \tag{4}$$

Here, m_t accumulates the gradient direction, v_t tracks its coordinate-wise squared magnitude, and the hatted quantities correct their initialization bias. Adam's proposed update direction is

$$u_t^{\mathrm{Adam}} = \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon}, \tag{5}$$

where the division and square root are applied coordinate-wise. With learning rate η_t, the resulting parameter displacement is

$$\Delta\theta_t = \theta_{t+1} - \theta_t = -\eta_t u_t^{\mathrm{Adam}}. \tag{6}$$

We next ask how this displacement affects the Mastered Set. Its gradient is

$$g_t^{\mathcal{M}} = \nabla_\theta \mathcal{L}^{\mathcal{M}}(\theta_t), \tag{7}$$

which points in the direction in which the mastered-set loss increases most rapidly. For a sufficiently small update, a first-order Taylor expansion around θ_t gives

$$\begin{aligned} \mathcal{L}^{\mathcal{M}}(\theta_t + \Delta\theta_t) &= \mathcal{L}^{\mathcal{M}}(\theta_t) + (g_t^{\mathcal{M}})^\top \Delta\theta_t + \mathcal{O}(\|\Delta\theta_t\|_2^2) \\ &= \mathcal{L}^{\mathcal{M}}(\theta_t) - \eta_t (g_t^{\mathcal{M}})^\top u_t^{\mathrm{Adam}} + \mathcal{O}(\eta_t^2 \|u_t^{\mathrm{Adam}}\|_2^2). \end{aligned} \tag{8}$$

Subtracting the loss before the update and omitting the second-order remainder yields

$$\Delta\mathcal{L}_t^{\mathcal{M}} = \mathcal{L}^{\mathcal{M}}(\theta_{t+1}) - \mathcal{L}^{\mathcal{M}}(\theta_t) \approx -\eta_t (g_t^{\mathcal{M}})^\top u_t^{\mathrm{Adam}}. \tag{9}$$

The sign of this loss change is therefore determined by the inner product between the mastered-set gradient and Adam's proposed direction. We call this quantity the Adam-aware gradient similarity

$$S_t^{\mathrm{Adam}}(\mathcal{M}, \mathcal{F}) = (g_t^{\mathcal{M}})^\top u_t^{\mathrm{Adam}}. \tag{10}$$

If $S_t^{\mathrm{Adam}} > 0$, then $\Delta\mathcal{L}_t^{\mathcal{M}} < 0$ to first order, so the update reduces the mastered-set loss. If $S_t^{\mathrm{Adam}} = 0$, the update has no first-order effect on that loss. If $S_t^{\mathrm{Adam}} < 0$, then $\Delta\mathcal{L}_t^{\mathcal{M}} > 0$: the Adam step increases the mastered-set loss and defined as "forgetting" in this study.

We subsequently decompose the Adam-aware similarity into parameter-wise contributions:

$$s_{j,t}^{\mathrm{Adam}} = g_{j,t}^{\mathcal{M}} u_{j,t}^{\mathrm{Adam}}, \qquad S_t^{\mathrm{Adam}} = \sum_{j=1}^{|\theta|} s_{j,t}^{\mathrm{Adam}}. \tag{11}$$

If $s_{j,t}^{\mathrm{Adam}} < 0$, the proposed update to parameter j increases the mastered-set loss to first order and thereby promotes forgetting; we call it a **Conflicting Parameter**. If $s_{j,t}^{\mathrm{Adam}} \geq 0$, the update does not increase the mastered-set loss: it decreases the loss when $s_{j,t}^{\mathrm{Adam}} > 0$ and leaves it unchanged to first order when $s_{j,t}^{\mathrm{Adam}} = 0$. Such an update mitigates or does not promote forgetting, and we call the corresponding parameter a **Collaborative Parameter**.

3.2 Collaborative Parameter Learning

During training, the complete Mastered Set is generally unavailable. Let $\mathcal{B}$ denote the accessible buffer. Its loss and reference gradient are

$$\mathcal{L}^{\mathcal{B}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \ell(f_\theta(x), y), \qquad g_t^{\mathcal{B}} = \nabla_\theta \mathcal{L}^{\mathcal{B}}(\theta_t). \tag{12}$$

We apply the parameter-wise decomposition to this reference gradient:

$$s_{j,t}^{\mathcal{B}} = g_{j,t}^{\mathcal{B}} u_{j,t}^{\mathrm{Adam}}. \tag{13}$$

CPL retains only the coordinates that are collaborative with the buffer. Define the mask

$$c_{j,t} = I\big(s_{j,t}^{\mathcal{B}} \geq 0\big), \qquad \theta_{t+1} = \theta_t - \eta_t \left(c_t \odot u_t^{\mathrm{Adam}}\right). \tag{14}$$

The corresponding first-order change in the buffer loss is

$$\Delta\mathcal{L}_t^{\mathcal{B}} \approx -\eta_t (g_t^{\mathcal{B}})^\top (c_t \odot u_t^{\mathrm{Adam}}) = -\eta_t \sum_{j=1}^{|\theta|} \max(s_{j,t}^{\mathcal{B}}, 0) \leq 0. \tag{15}$$

Thus, CPL removes coordinate contributions that would increase the buffer loss while retaining those that decrease it or leave it unchanged. For the complete Mastered Set, its first-order loss change can be decomposed as

$$\begin{aligned} \Delta\mathcal{L}_t^{\mathcal{M}} &\approx -\eta_t (g_t^{\mathcal{M}})^\top (c_t \odot u_t^{\mathrm{Adam}}) \\ &= -\eta_t \left[\sum_{j=1}^{|\theta|} \max(s_{j,t}^{\mathcal{B}}, 0) + (g_t^{\mathcal{M}} - g_t^{\mathcal{B}})^\top (c_t \odot u_t^{\mathrm{Adam}}) \right]. \end{aligned} \tag{16}$$

Therefore, the mask constructed from $\mathcal{B}$ guarantees $\Delta\mathcal{L}_t^{\mathcal{M}} \leq 0$ to first order if and only if

$$(g_t^{\mathcal{M}} - g_t^{\mathcal{B}})^\top (c_t \odot u_t^{\mathrm{Adam}}) \geq -\sum_{j=1}^{|\theta|} \max(s_{j,t}^{\mathcal{B}}, 0). \tag{17}$$

By Cauchy–Schwarz, a sufficient condition is

$$\|g_t^{\mathcal{M}} - g_t^{\mathcal{B}}\|_2 \|c_t \odot u_t^{\mathrm{Adam}}\|_2 \leq \sum_{j=1}^{|\theta|} \max(s_{j,t}^{\mathcal{B}}, 0). \tag{18}$$

This condition holds in particular when $g_t^{\mathcal{B}} = g_t^{\mathcal{M}}$. Because OGD is the closest method to CPL, Appendix A compares their first-order retention and learning effects under the same reference gradient and candidate update direction.

4 Experiments

4.1 Setup

Models and datasets. Our experiments cover six datasets across three modalities: MMLU (Hendrycks et al., 2020), MedQA (Jin et al., 2020), ARC-C (Clark et al., 2018), CSQA (Talmor et al., 2019), MMBench (image) (Liu et al., 2024), and MMSU (audio) (Wang et al., 2025). Five models are used across studies: *Qwen2.5-1.5B-Instruct*, *Qwen2.5-7B-Instruct*, *Meta-Llama-3.1-8B-Instruct*, *Qwen2.5-VL-3B-Instruct*, and *Qwen2.5-Omni-7B* (Qwen et al., 2024; Grattafiori et al., 2024; Bai et al., 2025; Xu et al., 2025). In the remainder of this paper, we omit detailed version identifiers and the "Instruct" suffix for simplicity.

Set construction. Our study follows the model-refinement setting (Jin & Ren, 2024; Zhou et al., 2025). For each dataset, we independently draw three random 7:3 training–test splits from the complete example pool. Based on the initial model predictions, each split is divided into four subsets: $\mathcal{F}_{\mathrm{train}}$ and $\mathcal{M}_{\mathrm{train}}$ from the training partition, and $\mathcal{F}_{\mathrm{test}}$ and $\mathcal{M}_{\mathrm{test}}$ from the test partition, where $\mathcal{F}$

Table 1: FT in the BOTTOM and TOP groups across models, datasets, and modalities. Groups are ranked by the magnitude of negative Adam-aware gradient similarity, with TOP denoting the strongly negative group and BOTTOM the weakly negative group.

(a) Text tasks

Model	MMLU		MedQA		ARC-C		CSQA	
	BOTTOM	TOP	BOTTOM	TOP	BOTTOM	TOP	BOTTOM	TOP
Qwen 1.5B	$0.8_{\pm 0.0}$	$57.3_{\pm 1.6}$	$10.6_{\pm 0.3}$	$58.5_{\pm 0.2}$	$0.2_{\pm 0.1}$	$5.4_{\pm 1.2}$	$0.7_{\pm 0.1}$	$22.4_{\pm 1.2}$
Qwen 7B	$0.1_{\pm 0.1}$	$28.9_{\pm 1.0}$	$0.1_{\pm 0.0}$	$35.1_{\pm 0.8}$	$0.0_{\pm 0.0}$	$7.1_{\pm 1.2}$	$0.7_{\pm 0.5}$	$24.7_{\pm 2.8}$
LLaMA 8B	$14.9_{\pm 2.6}$	$80.6_{\pm 3.5}$	$2.6_{\pm 0.8}$	$99.8_{\pm 0.4}$	$0.5_{\pm 0.4}$	$86.8_{\pm 4.0}$	$4.5_{\pm 1.4}$	$91.0_{\pm 1.3}$

(b) Image and audio tasks

Model	Dataset	BOTTOM	TOP
Qwen VL 3B	MMBench (Image)	$0.4_{\pm 0.4}$	$26.5_{\pm 15.5}$
Qwen Omni 7B	MMSU (Audio)	$5.9_{\pm 6.5}$	$61.5_{\pm 11.8}$

and $\mathcal{M}$ denote initially incorrect and correct examples, respectively. $\mathcal{F}_{\text{train}}$ provides the fine-tuning examples. Since mastered examples are generally unavailable a priori in real-world scenarios, we use only $\mathcal{M}_{\text{train}}$ as the buffer during training and evaluate forgetting on $\mathcal{M}_{\text{test}}$. Results are computed separately over the three random splits and reported as mean $\pm$ sample standard deviation. Optimization, batch size, Adam hyperparameters, numerical precision, software packages, hardware, and inference settings are detailed in Appendix B.

Baselines. We compare CPL with nine forgetting-mitigation baselines. Methods that restrict trainable parameters or update subspaces include Bottom-Layer Freezing (BLF) (Zheng et al., 2025), O-LoRA (Wang et al., 2023), and NTK-CL (Liu et al., 2026). Replay mixes initially incorrect training examples and buffer examples in equal proportions. The regularization baseline is EWC (Kirkpatrick et al., 2017). Gradient-projection baselines include A-GEM (Chaudhry et al., 2019), GPM (Saha et al., 2021), OGD (Farajtabar et al., 2020), and OPGD (Zheng et al., 2026). Although several of these methods were originally developed for continual learning, we include them because their update rules also aim to mitigate forgetting. Our study focuses on model refinement rather than continual learning, in which a model learns a sequence of tasks. Baseline implementations and the shared-buffer protocol are detailed in Appendix B.4.

Evaluation metrics. We report accuracy (Acc) on $\mathcal{F}_{\text{test}} \cup \mathcal{M}_{\text{test}}$ and forgetting rate (FT) on $\mathcal{M}_{\text{test}}$:

$$\text{Acc}_t = \frac{\sum_{i \in \mathcal{F}_{\text{test}} \cup \mathcal{M}_{\text{test}}} I(\hat{y}_i(\theta_t) = y_i)}{|\mathcal{F}_{\text{test}}| + |\mathcal{M}_{\text{test}}|}, \qquad \text{FT}_t = \frac{\sum_{i \in \mathcal{M}_{\text{test}}} I(\hat{y}_i(\theta_t) \neq y_i)}{|\mathcal{M}_{\text{test}}|}. \tag{19}$$

Higher Acc and lower FT indicate better overall performance and retention, respectively.

4.2 Strongly Negative Gradient Similarity Leads to More Forgetting

Using the Adam-aware gradient similarity defined in Eq. 10, we rank mastered training samples with negative initial similarities by magnitude. The top third forms the **TOP** group with strongly negative interactions, while the bottom third forms the **BOTTOM** group with weakly negative interactions; we then compare their FT after fine-tuning.

Table 1 shows that FT is consistently concentrated in the TOP group across five models, six datasets, and three modalities. Across all 14 model–dataset combinations, an average of 49.0% of TOP samples are forgotten, compared with only 3.0% of BOTTOM samples. These results indicate that mastered examples whose gradients have strongly negative inner products with Adam's proposed direction are substantially more susceptible to forgetting.

4.3 Conflicting and Collaborative Parameters Coexist During Fine-Tuning

We next investigate whether an overall conflicting update is uniformly harmful or instead contains parameters that affect mastered knowledge differently. At initialization, we compute the inner product between the mastered-set gradient and Adam's proposed direction, then apply the parameter-wise

Table 2: Parameter-wise decomposition across models, datasets, and modalities. PROP gives the percentages of Collaborative (COLL) and Conflicting (CONF) Parameters; GRAD gives their contributions to the Adam-aware gradient similarity; TOTAL sums both contributions.

(a) Text tasks

Model	Metric	MMLU		MedQA		ARC-C		CSQA	
		COLL	CONF	COLL	CONF	COLL	CONF	COLL	CONF
Qwen 1.5B	PROP	$46.9_{\pm 0.6}$	$53.1_{\pm 0.6}$	$37.6_{\pm 0.4}$	$62.4_{\pm 0.4}$	$51.3_{\pm 1.7}$	$48.7_{\pm 1.7}$	$43.2_{\pm 1.4}$	$56.8_{\pm 1.4}$
	GRAD	$361_{\pm 23}$	$-1218_{\pm 27}$	$200_{\pm 20}$	$-2673_{\pm 73}$	$624_{\pm 45}$	$-837_{\pm 13}$	$286_{\pm 21}$	$-1185_{\pm 32}$
	TOTAL	$-857_{\pm 35}$		$-2473_{\pm 55}$		$-213_{\pm 58}$		$-899_{\pm 50}$	
Qwen 7B	PROP	$43.9_{\pm 0.7}$	$56.1_{\pm 0.7}$	$44.1_{\pm 0.6}$	$55.9_{\pm 0.6}$	$50.9_{\pm 0.8}$	$49.1_{\pm 0.8}$	$48.6_{\pm 0.8}$	$51.4_{\pm 0.8}$
	GRAD	$185_{\pm 13}$	$-889_{\pm 94}$	$238_{\pm 18}$	$-890_{\pm 12}$	$412_{\pm 52}$	$-588_{\pm 50}$	$243_{\pm 22}$	$-506_{\pm 7}$
	TOTAL	$-703_{\pm 106}$		$-652_{\pm 28}$		$-177_{\pm 24}$		$-263_{\pm 18}$	
LLaMA 8B	PROP	$38.5_{\pm 0.1}$	$61.5_{\pm 0.1}$	$43.9_{\pm 0.9}$	$56.1_{\pm 0.9}$	$42.9_{\pm 1.7}$	$57.1_{\pm 1.7}$	$38.0_{\pm 0.4}$	$62.0_{\pm 0.4}$
	GRAD	$768_{\pm 21}$	$-4133_{\pm 51}$	$1696_{\pm 134}$	$-5365_{\pm 92}$	$1073_{\pm 210}$	$-4303_{\pm 311}$	$508_{\pm 31}$	$-3917_{\pm 77}$
	TOTAL	$-3365_{\pm 31}$		$-3669_{\pm 196}$		$-3230_{\pm 430}$		$-3409_{\pm 72}$	

(b) Image and audio tasks

Model	Dataset	Metric	COLL	CONF
Qwen VL 3B	MMBench (Image)	PROP	$44.6_{\pm 0.6}$	$55.4_{\pm 0.6}$
		GRAD	$447_{\pm 230}$	$-556_{\pm 237}$
		TOTAL	$-109_{\pm 7}$	
Qwen Omni 7B	MMSU (Audio)	PROP	$43.0_{\pm 0.2}$	$57.0_{\pm 0.2}$
		GRAD	$90_{\pm 17}$	$-198_{\pm 5}$
		TOTAL	$-108_{\pm 20}$	

decomposition in Eq. 11. For each model–dataset combination, we report the proportions of Collaborative and Conflicting Parameters and their respective contributions to the Adam-aware similarity.

Table 2 shows that Conflicting and Collaborative Parameters coexist in every model–dataset combination. Across all 14 combinations, Collaborative Parameters account for 44.1% of coordinates on average, while Conflicting Parameters account for the remaining 55.9% on average. Every combination has a positive collaborative contribution but a negative total inner product because the conflicting contribution is larger. Thus, an Adam update that is conflicting at the vector level still contains a substantial set of parameters that help preserve mastered knowledge.

4.4 CPL BETTER BALANCES LEARNING AND RETENTION

4.4.1 MAIN RESULTS

To evaluate whether CPL can learn initially incorrect examples while preserving mastered knowledge, we compare it with nine forgetting-mitigation methods during fine-tuning. The main comparison covers four text datasets. To further compare CPL with OGD beyond text-only settings, we extend the evaluation to multimodal tasks, including image and audio datasets.

Table 3, panel (a), shows that CPL achieves the highest mean Acc on all four text datasets: 61.0% on MMLU, 45.1% on MedQA, 79.1% on ARC-C, and 79.7% on CSQA. It also obtains the lowest FT on CSQA and the second-lowest FT on the remaining three datasets. These results suggest that CPL achieves a favorable trade-off between learning and retention.

In panel (b), CPL outperforms OGD on MMBench, increasing mean Acc from 82.4% to 97.5% and reducing FT from 15.5% to 0.4%. On MMSU, both methods achieve 70.2% mean Acc, while CPL reduces FT from 7.8% to 3.8%.

4.4.2 HYPER-PARAMETER SENSITIVITY

To assess CPL's hyperparameter sensitivity, we vary the reference-set fraction and mask-refresh interval on MMLU. As shown in Table 4, mean Acc ranges from 60.5% to 61.1% across reference-

Table 3: Comparison across text, image, and audio tasks. (a) Comparison of different methods on *Qwen 1.5B*. Results are percentages, reported as mean ± sample standard deviation over three splits. Bold and underlined values indicate the best and second-best mean in each column. (b) Comparison between CPL and OGD across modalities. Bold values indicate the best mean in each dataset–metric column, while underlined values indicate the second-best mean. CPL Better Balances Learning and Retention.

(a) Text tasks

Method	MMLU		MedQA		ARC-C		CSQA	
	Acc ↑	FT ↓	Acc ↑	FT ↓	Acc ↑	FT ↓	Acc ↑	FT ↓
BLF	55.0±1.1	30.7±0.8	42.5±0.8	47.6±0.6	76.6±0.9	10.4±0.7	74.5±1.3	16.2±0.8
O-LoRA	48.5±1.1	42.5±2.0	34.3±0.5	64.3±1.1	74.0±1.3	17.1±2.9	70.5±0.8	22.8±1.2
Replay	57.3±1.2	19.4±2.1	43.7±2.3	35.6±3.8	76.1±0.3	9.2±0.1	76.8±0.9	<u>8.5</u>±0.6
A-GEM	57.9±1.6	27.1±1.8	44.3±1.0	44.8±1.2	76.9±0.8	10.7±0.4	77.0±0.4	13.0±0.8
GPM	56.2±1.8	29.9±2.3	42.6±2.0	46.8±4.6	<u>77.3</u>±0.5	10.1±0.9	75.5±1.1	14.5±1.5
EWC	56.7±1.2	28.5±1.2	43.4±1.0	44.3±2.9	76.7±0.5	10.4±0.8	77.1±1.4	12.4±1.4
OGD	58.0±1.1	26.2±1.0	41.5±0.7	48.1±1.9	75.8±0.8	11.8±1.4	<u>77.2</u>±0.8	12.5±0.6
OPGD	55.3±0.8	31.2±1.8	39.9±0.4	52.1±0.7	77.2±0.1	10.0±1.0	76.1±0.7	13.7±0.4
NTK-CL	<u>58.2</u>±0.3	**7.8**±0.5	<u>44.9</u>±1.0	**6.4**±0.3	69.5±1.1	**2.7**±0.7	69.4±0.8	10.1±0.9
CPL	**61.0**±1.0	<u>18.0</u>±0.2	**45.1**±1.6	<u>33.9</u>±4.1	**79.1**±0.7	<u>6.3</u>±1.5	**79.7**±0.6	**7.7**±0.2

(b) Image and audio tasks

Model	Dataset	Method	Acc ↑	FT ↓
Qwen VL 3B	MMBench (Image)	OGD	<u>82.4 ± 9.4</u>	<u>15.5 ± 10.1</u>
		CPL	**97.5 ± 0.2**	**0.4 ± 0.2**
Qwen Omni 7B	MMSU (Audio)	OGD	**70.2 ± 0.5**	<u>7.8 ± 1.9</u>
		CPL	**70.2 ± 0.4**	**3.8 ± 0.7**

Table 4: Sensitivity analyses for (a) Reference-set size and (b) Mask-refresh interval. Green shading marks the configurations used in the main experiments.

(a) Reference-set size

Fraction of $\mathcal{M}_{\text{train}}$	Acc ↑	FT ↓
50.0%	60.5±1.1	20.3±1.4
66.7%	61.1±1.1	17.9±1.9
100.0%	61.0±1.0	18.0±0.2

(b) Mask-refresh interval

Refresh interval (epochs)	Acc ↑	FT ↓
0.5	61.4±0.6	18.1±0.3
1	61.0±1.0	18.0±0.2
2	60.0±1.8	20.7±2.9

set fractions and from 60.0% to 61.4% across mask-refresh intervals. This indicates that CPL is stable across those hyperparameter choices.

4.4.3 Wall-Time Accuracy

We compare CPL's wall-time accuracy trajectory and peak allocated GPU memory with the classical OGD baseline and the more recent OPGD method. Appendix Figure 6 shows that CPL exceeds the highest accuracy reached by either OGD or OPGD after 0.85 hours on MMLU, 1.28 hours on MedQA, 0.11 hours on ARC-C, and 0.38 hours on CSQA. CPL also uses 38.5 GiB of peak allocated GPU memory, compared with 59.7 GiB for OGD and 57.0 GiB for OPGD. This lower memory footprint is partly attributable to INT8 storage.

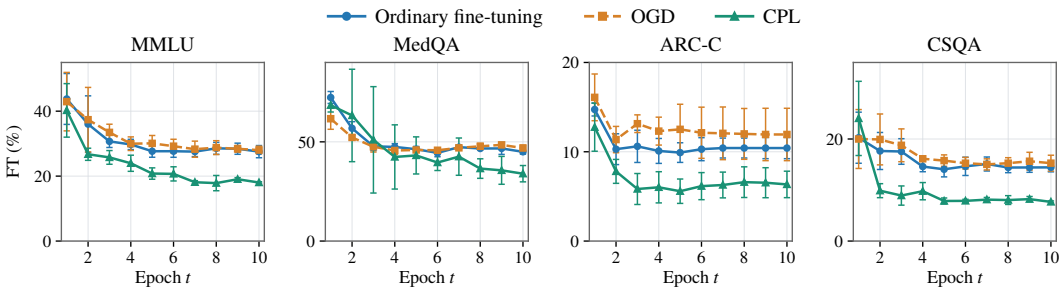

Figure 2: FT trajectories of *Qwen 1.5B* across training epochs. All three methods recover a portion of the forgotten examples during training, with CPL recovering more forgotten examples and ending with lower final FT.

Table 5: Cosine similarity ($\times 100$) between each update direction and the gradient on $\mathcal{M}_{\text{test}}$. Higher indicates better alignment between fine-tuning set and mastered set.

Method	MMLU	MedQA	CSQA	ARC-C
Fine-tuning	$-88.1_{\pm 1.0}$	$-96.9_{\pm 0.3}$	$-88.1_{\pm 0.9}$	$-54.9_{\pm 1.0}$
OGD	$-0.9_{\pm 1.7}$	$-2.6_{\pm 1.8}$	$-0.2_{\pm 1.6}$	$-2.1_{\pm 1.1}$
CPL	$\mathbf{7.7}_{\pm 0.9}$	$\mathbf{1.9}_{\pm 0.6}$	$\mathbf{8.3}_{\pm 0.8}$	$\mathbf{20.9}_{\pm 0.9}$

4.5 CPL Mechanism Analysis

4.5.1 Dynamics of Forgetting

To investigate how CPL achieves higher Acc, we track FT_t throughout fine-tuning for ordinary fine-tuning, OGD, and CPL (Figure 2). For all three methods, forgetting rises sharply at the beginning of training and then declines as some forgotten answers are recovered. CPL exhibits the strongest recovery and ends with the lowest FT.

We measure the Jaccard similarity between the sets of correctly answered mastered examples at consecutive epochs. Higher similarity indicates greater overlap in correct predictions. Appendix Figure 5 shows that this overlap generally increases as training progresses.

4.5.2 Gradient Alignment on Held-Out Mastered Examples

To assess alignment with unseen mastered knowledge, we use $\mathcal{M}_{\text{train}}$ to construct the OGD and CPL update directions and compare their cosine similarity, together with fine-tuning, against the gradient on $\mathcal{M}_{\text{test}}$. Table 5 shows that CPL achieves the highest cosine similarity on all four datasets, indicating the best gradient alignment.

4.5.3 Representation Characteristic of Forgetting

To investigate representational differences between forgotten and not-forgotten examples in $\mathcal{M}_{\text{test}}$, we use linear centered kernel alignment (CKA) (Kornblith et al., 2019) to quantify hidden-representation drift from before to after fine-tuning under ordinary fine-tuning, OGD, and CPL. We divide the network into three layer groups: lower layers (1–9), middle layers (10–19), and upper layers (20–28). Figure 3 shows similar lower-layer drift between the two groups but substantially greater upper-layer drift for forgotten examples.

We further compare representation drift across layers after ordinary fine-tuning, OGD, and CPL. Figure 4 shows that CPL consistently preserves upper-layer representations better, while sacrificing some lower- and middle-layer stability on most datasets. Since upper-layer drift is more strongly associated with forgetting (Figure 3), this trade-off may help explain CPL's lower forgetting and stronger performance.

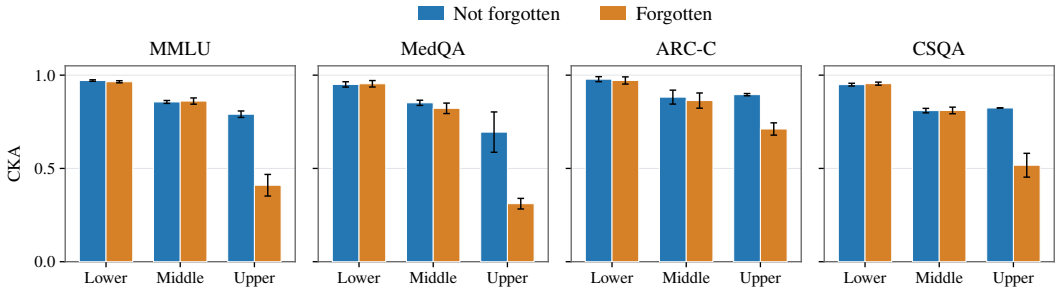

Figure 3: CKA is used to quantify changes in hidden representations before and after ordinary fine-tuning, with lower CKA indicating greater representational drift. The network is divided into three layer groups: lower layers (1–9), middle layers (10–19), and upper layers (20–28). Forgotten samples exhibit greater representational drift in the upper layers.

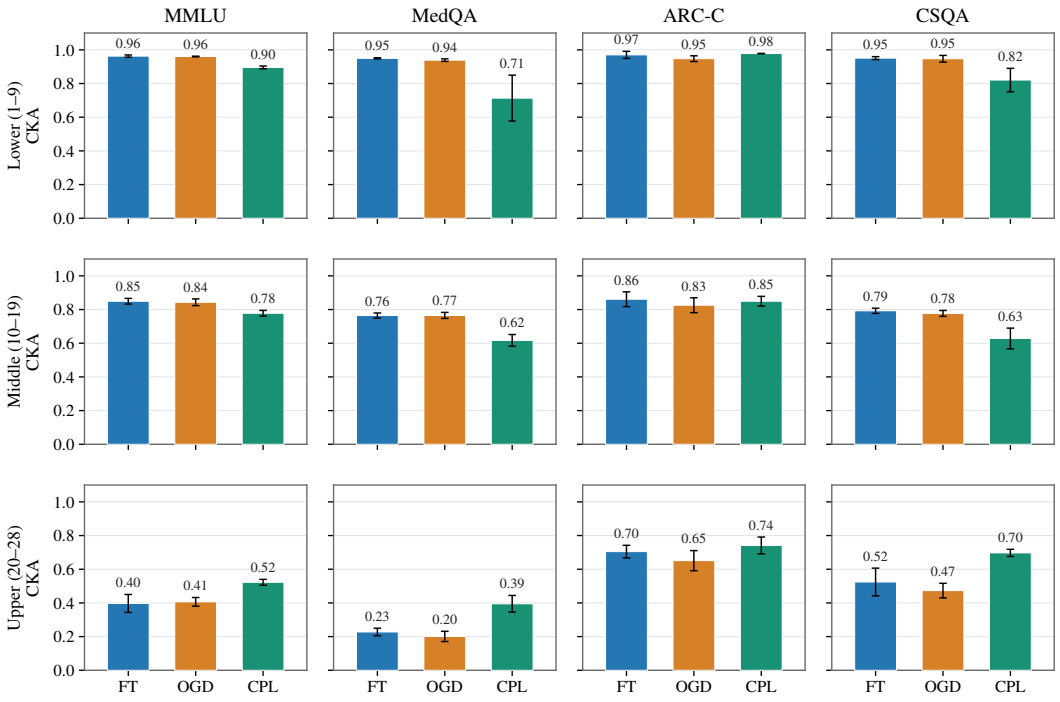

Figure 4: CKA is used to quantify changes in hidden representations before and after fine-tuning, with lower CKA indicating greater representational drift. Compared with ordinary fine-tuning and OGD, CPL better preserves upper-layer representations.

CONCLUSION

We analyze forgetting through Adam-aware parameter-level gradient interactions and introduce Collaborative Parameter Learning (CPL), which masks conflicting updates while preserving collaborative ones. Experiments across text, image, and audio tasks show that CPL balances learning and retention. Prediction and representation analyses associate its reduced forgetting with stronger answer recovery and better preservation of upper-layer representations.Our study focuses on single-task fine-tuning rather than a sequence of tasks. Future work can extend CPL to multi-task continual learning.

AI USE STATEMENT

We used generative AI tools to assist with drafting portions of the manuscript and improving the clarity and readability of the writing. These tools also assisted with the presentation of mathematical derivations, interpretation of reported results, LaTeX formatting, and figure preparation. They were not used to generate experimental data or results. All authors reviewed and revised the complete manuscript and all AI-assisted content.

ETHICS STATEMENT

This work uses publicly available benchmarks and pretrained models to study knowledge acquisition and forgetting during fine-tuning. It does not involve human participants, collection of private or personally identifiable data, or deployment in real-world decision-making systems. We identify no direct ethical concerns arising from the experiments conducted in this study. Reuse of the datasets and models should follow their respective licenses.

REPRODUCIBILITY STATEMENT

Section 4.1 describes the data splits and evaluation metrics. Appendix A provides the first-order comparison of CPL and OGD, Appendix B specifies the hardware, software, training configurations, and baseline implementations, and Appendix C describes the representation-analysis protocol.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Thang Doan, Mehdi Abbana Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A Theoretical Analysis of Catastrophic Forgetting through the NTK Overlap Matrix. In *International Conference on Artificial Intelligence and Statistics*, pp. 1072–1080, 2021. URL https://proceedings.mlr.press/v130/doan21a.html.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal Gradient Descent for Continual Learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773. PMLR, 2020.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

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. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *International Conference on Machine Learning*, pp. 3519–3529, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding Catastrophic Forgetting in Language Models via Implicit Inference. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/692ae28fda9bfbde7c01b13bf5a03395-Abstract-Conference.html.

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, 2024. doi: 10.18653/v1/2024.findings-emnlp.249. URL https://aclanthology.org/2024.findings-emnlp.249/.

Zhizhong Li and Derek Hoiem. Learning without Forgetting. In *European Conference on Computer Vision*, pp. 614–629, 2016. doi: 10.1007/978-3-319-46493-0_37.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

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. URL https://doi.org/10.1109/TPAMI.2026.3670952.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is Your Multi-modal Model an All-around Player? In *European Conference on Computer Vision*, 2024. doi: 10.1007/978-3-031-72658-3_13.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What Happens To BERT Embeddings During Fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 33–44. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.blackboxnlp-1.4. URL https://aclanthology.org/2020.blackboxnlp-1.4/.

Qwen, An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Anthony Robins. Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connection Science*, 7 (2):123–146, 1995. doi: 10.1080/09540099550039318.

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=ZK4Cvv5Eci.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Dingdong Wang, Jincenzi Wu, Junan Li, Dongchao Yang, Xueyuan Chen, Tianhua Zhang, and Helen Meng. MMSU: A Massive Multi-task Spoken Language Understanding and Reasoning Benchmark. *arXiv preprint arXiv:2506.04779*, 2025. URL https://arxiv.org/abs/2506.04779.

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. URL https://aclanthology.org/2023.findings-emnlp.715/.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni Technical Report. *arXiv preprint arXiv:2503.20215*, 2025. URL https://arxiv.org/abs/2503.20215.

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, 2026. doi: 10.1609/aaai.v40i40.40723. URL https://ojs.aaai.org/index.php/AAAI/article/view/40723.

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.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Xiao Zhang and Ji Wu. Dissecting Learning and Forgetting in Language Model Finetuning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/c4de749a4bd8802f0b4033d09a7867db-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, pp. 10066–10096, 2025. doi: 10.52202/085713-0308. URL https://papers.neurips.cc/paper_files/paper/2025/hash/0d2c50691b832d8e8901715998e7c22c-Abstract-Conference.html.

A First-Order Comparison of CPL and OGD

Let $r \in \mathbb{R}^d$ be the nonzero replay-buffer reference gradient, $p = \nabla_\theta \mathcal{L}^{\mathcal{F}}(\theta)$ the fine-tuning gradient, and $u \in \mathbb{R}^d$ the candidate update direction. Using the same r and u, OGD (Farajtabar et al., 2020) removes the component of u parallel to r:

$$d_{\mathrm{O}} = u - \frac{r^\top u}{\|r\|_2^2} r, \tag{20}$$

whereas CPL removes coordinates with negative interactions:

$$d_{\mathrm{C}} = u \odot \mathbf{1}[r \odot u \geq 0] = u - u_-, \qquad u_- = u \odot \mathbf{1}[r \odot u < 0]. \tag{21}$$

Because the parameter displacement is $-\eta d$, define the first-order retention and learning scores as

$$\mathcal{R}(d) = r^\top d, \qquad \mathcal{A}(d) = p^\top d. \tag{22}$$

Larger scores correspond to larger first-order decreases in the mastered-set and fine-tuning losses, respectively.

A.1 Retention comparison

OGD makes the update orthogonal to the reference gradient, while CPL retains all nonnegative coordinate contributions. Therefore,

$$\mathcal{R}(d_{\mathrm{O}}) = 0, \tag{23}$$

$$\mathcal{R}(d_{\mathrm{C}}) = \sum_{j=1}^{d} \max(r_j u_j, 0) \geq 0. \tag{24}$$

Thus, CPL preserves at least as much first-order reference alignment as OGD and achieves strictly greater alignment whenever $r_j u_j > 0$ for at least one coordinate.

A.2 Learning comparison

From Eqs. 20 and 21,

$$d_{\mathrm{C}} - d_{\mathrm{O}} = \frac{r^\top u}{\|r\|_2^2} r - u_-. \tag{25}$$

The difference in first-order learning signal is therefore

$$\mathcal{A}(d_{\mathrm{C}}) - \mathcal{A}(d_{\mathrm{O}}) = \frac{(r^\top u)(p^\top r)}{\|r\|_2^2} - p^\top u_-. \tag{26}$$

Consequently, CPL preserves at least as much first-order fine-tuning signal as OGD if and only if

$$\frac{(r^\top u)(p^\top r)}{\|r\|_2^2} \geq p^\top u_-. \tag{27}$$

The left side is the fine-tuning signal removed by OGD's orthogonal projection, while the right side is the signal removed by CPL's coordinate mask.

B Implementation Details

B.1 Hardware

Experiments were conducted on a server equipped with eight NVIDIA A800-SXM4-80GB GPUs. The number of GPUs allocated to each run was determined by model size.

B.2 Software packages

The software environment consists of Python 3.10, PyTorch 2.10.0 with CUDA 12.8, Transformers 4.57.6, Datasets 2.21.0, and Accelerate 1.13.0.

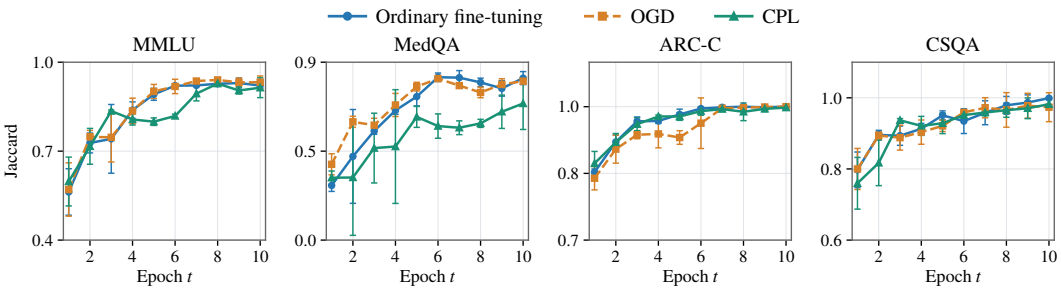

Figure 5: Consecutive-epoch Jaccard similarity of correctly answered mastered examples for *Qwen 1.5B*. Higher overlap indicates more stable recovery.

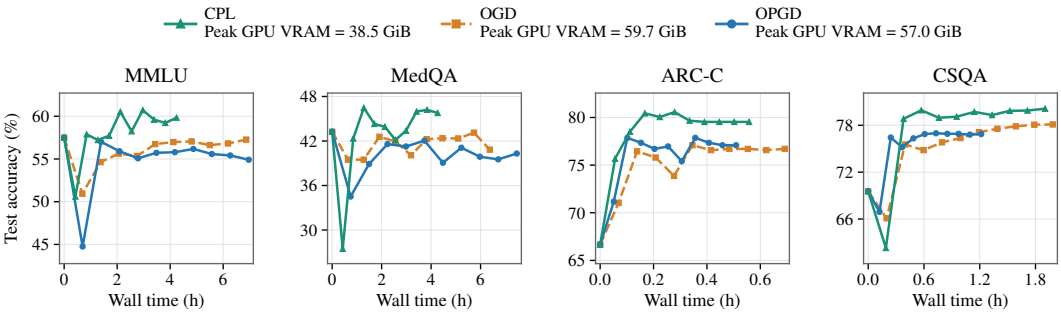

Figure 6: Test accuracy as a function of wall time.

B.3 Text and multimodal training

For the text experiments, the four-dataset trajectory suite includes three methods and three partitions per dataset. Training uses FP32 Adam, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, no weight decay, learning rate 10^{-6}, batch size 8, and 10 epochs. Adam moments are updated before the projection or mask. Because CPL uses only the sign of the reference gradient to construct its coordinate-wise mask, the stored reference can be represented in INT8.

For the multimodal experiments, training uses BF16 Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and no weight decay. *Qwen2.5-VL-3B* trains at 10^{-5} with global batch size 4, and *Qwen2.5-Omni-7B* trains at 5×10^{-6} with batch size 1. Both datasets report the results on epoch 5. Across all modalities, inference is deterministic: predictions are obtained by taking the argmax over the candidate answer-token logits, corresponding to temperature 0 with no sampling.

Baseline configurations. OGD refreshes its reference gradient at each epoch. GPM uses a 95% energy threshold, rank cap 32, and a 128-vector reservoir. OPGD uses two orthonormal output-gradient directions, modification coefficient 0.5, and perturbation radius 0.05, with Hessian–vector products restricted to the final Transformer layer. EWC uses a regularization coefficient of 5,000. Both O-LoRA adapters have rank 8, with the orthogonality penalty applied to the query and value projections. BLF freezes the token embeddings and bottom three Transformer layers.

B.4 Baseline implementations

Shared reference buffer. Within each data split, $\mathcal{M}_{\text{train}}$ is fixed before training and serves as the common buffer $\mathcal{B}$. All projection methods use exactly the same buffer, with identical examples and buffer size, as their reference set. CPL uses the same $\mathcal{B}$ to construct its coordinate mask. The methods differ only in how they use this shared information, and no example from $\mathcal{M}_{\text{test}}$ enters training, reference construction, or hyperparameter selection.

Projection methods. A-GEM computes a replay-reference gradient from cyclic mini-batches of $\mathcal{B}$ and projects the current gradient only when their inner product is negative (Chaudhry et al., 2019). OGD computes the cross-entropy gradient on $\mathcal{B}$ and removes its component from Adam's proposed direction independently within each parameter tensor (Farajtabar et al., 2020). GPM constructs per-layer activation bases from $\mathcal{B}$, then removes gradient components in those bases (Saha et al., 2021). OPGD constructs orthonormal output-gradient directions from $\mathcal{B}$ and uses them to modify the update through gradient alignment (Zheng et al., 2026).

Regularization methods. EWC estimates a diagonal empirical Fisher on $\mathcal{B}$ and penalizes movement from the initial parameters (Kirkpatrick et al., 2017).

Restricted-update methods. O-LoRA first trains and freezes a mastered adapter on $\mathcal{B}$, then trains a new adapter on the Fine-Tuning Set with an orthogonality penalty (Wang et al., 2023). BLF restricts updates by freezing the embeddings and lower Transformer layers (Zheng et al., 2025). The NTK-CL adaptation freezes the Qwen backbone, constructs historical answer prototypes from $\mathcal{B}$, and trains only its token, channel, and fusion modules on the Fine-Tuning Set (Liu et al., 2026).

Replay. Replay jointly trains on new examples and previously mastered examples stored in $\mathcal{B}$. The training objective combines losses on both groups, so the update reflects both learning new knowledge and maintaining correct predictions on stored examples. Repeated exposure to mastered examples supplies a retention signal that helps counteract forgetting during fine-tuning.

C Representation-Analysis Protocol

Let $X_{\ell,t}$ contain the last-non-padding prompt-token hidden states for a fixed sample set at layer ℓ and epoch t. Form $K_{\ell,t} = X_{\ell,t}X_{\ell,t}^\top$ and $H = I - \mathbf{1}\mathbf{1}^\top/n$. Linear CKA is

$$\mathrm{CKA}(K_{\ell,0}, K_{\ell,t}) = \frac{\langle HK_{\ell,0}H, HK_{\ell,t}H\rangle_F}{\|HK_{\ell,0}H\|_F \|HK_{\ell,t}H\|_F}.$$

Figures 3 and 4 report the complete four-dataset CKA results. It uses Qwen2.5-1.5B-Instruct, which has 28 transformer layers and hidden size 1,536. Transformer layers 1–28 are used, excluding the embedding output. Blocks are lower 1–9, middle 10–19, and upper 20–28.