CONTINUAL LEARNING VIA SELF-PROBE GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting pretrained models to new data can cause catastrophic forgetting of previously learned behavior. When only a few past samples remain, they give continual learning methods sparse and narrow evidence about what to preserve. We show that language models can expand this evidence through self-probing, in which the frozen model generates new inputs from the retained samples and records its own predictions on them. Unlike prior work that replays such data as training examples, our method, CPLUS uses self-probe and past-sample gradients to scale down parameter updates that conflict with prior behavior. Experiments with five language models on four benchmarks show three results. First, the same probes preserve more prior behavior as gradient signals than as replay data. Second, CPLUS learns the new data while consistently reducing forgetting more than existing baselines, especially when past data are scarce, and this protection extends to benchmarks not used for training. Third, we observe that CPLUS also becomes more effective as models grow: within the Qwen3 model family, it recovers an increasing share of the forgetting caused by standard fine-tuning.

1 INTRODUCTION

Foundation models are pretrained with substantial compute, data, and engineering effort, yet often require further training to correct errors, incorporate new facts, or acquire domain-specific capabilities. Such updates can overwrite prior behavior, a phenomenon known as catastrophic forgetting (McCloskey & Cohen, 1989). Continual learning (CL) methods mitigate forgetting in several ways, and many of them rely on evidence of prior behavior, such as retained data or statistics derived from it (Rolnick et al., 2019; Zhou et al., 2024) For publicly released pretrained models (Yang et al., 2025; Team et al., 2024; Abdin et al., 2024), however, users typically receive model weights without the original training corpus or train-time statistics. At best, they can collect a few examples of behavior worth keeping, such as inputs the model already answers correctly (*i.e.*, anchors). Anchors give only a sparse and narrow view of the behavior that should be preserved, which makes forgetting harder to prevent (Knoblauch et al., 2020). Figure 1 shows both effects: preservation weakens as fewer anchors are available, and preserving the task they represent does not guarantee preservation elsewhere.

A natural way to compensate for limited past data is to recover additional evidence from the base model itself. Given a few past samples as in-context demonstrations (Brown et al., 2020), the model can generate new inputs for the same task and label them with its own predictions, denoted as self-probes. Prior work has mainly used such model-generated data as reply data or distillation targets (Huang et al., 2024). When used as training targets, however, self-probes enter the update itself: possibly incorrect predictions are learned as if they were labels, and they do not indicate which parts of a new data update would disrupt prior behavior. In a complementary direction, Yang et al. (2026) show that, when the relevant past data are fully available, their gradients can identify *collaborative parameters*, whose proposed updates are predicted to preserve prior behavior. With only a few anchors, however, this evidence becomes sparse and narrow, and CPL forgets more (Figure 1). Together, these observations suggest using self-probe gradients to broaden the evidence for identifying collaborative parameters.

We develop this idea as **CPLUS**: **C**ollaborative **P**arameter **L**earning **U**sing **S**elf-Probes. CPLUS combines gradients from anchors and self-probes to estimate, for each parameter, whether the update proposed by learning new data would preserve or disrupt prior behavior. It keeps collaborative updates and scales down those predicted to cause forgetting. Because these decisions can change from step to step, CPLUS smooths them over training steps into a soft mask. Note that the self-probes

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.

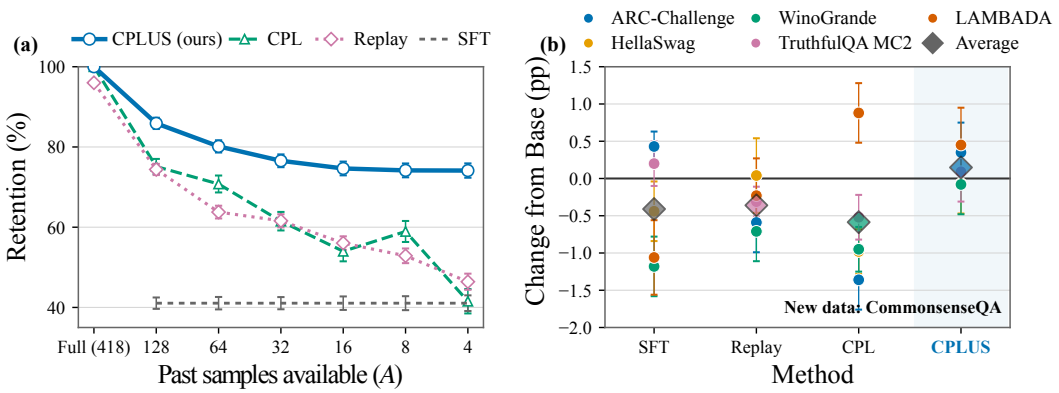

Figure 1: **Limited past-data access leaves preservation signals both sparse and narrow.** **(a) Preservation weakens when past samples are scarce.** Qwen3-0.6B is adapted to new CommonsenseQA data. Replay, Collaborative Parameter Learning (CPL), and CPLUS (Ours) use A retained examples (anchors); supervised fine-tuning (SFT) uses only the new data. Retention is the percentage of initially correct examples that remain correct after adaptation (Correct → Correct). It generally decreases as A shrinks, indicating increased forgetting. CPLUS achieves higher Retention across anchor budgets. **(b) Preserving the anchored target task does not necessarily preserve performance elsewhere.** After the same CommonsenseQA adaptation, we evaluate five benchmarks that are not used for adaptation or as anchors. Colored points report each task's accuracy change from the base model, and diamonds report the five-task average. Points and error bars are means and standard errors over five seeds; all methods attain 100% Acquisition (Incorrect → Correct) on CommonsenseQA. CPLUS remains closest to the base model across the profile.

are never used as training targets. They only decide how much of each proposed update is applied, so an incorrect probe can slow down a change but cannot pull the model toward its own error.

We evaluate CPLUS on three Qwen3 models (0.6B, 1.7B, and 4B), Gemma-2-2B-it, and Phi-3.5 Mini, using CommonsenseQA, ARC-Challenge, MMLU, and MedQA as new data with 4 to 128 anchors. CPLUS learns the new data while achieving higher Retention, the share of initially correct answers that remain correct, than all main baselines in 118 of 120 settings. The advantage is clearest when anchors are scarce: with four or eight anchors, CPLUS leads in all 24 Qwen 3 settings, by 7.73 points on average. It also preserves performance more broadly, maintaining near-base or better scores across five benchmarks not used for training (Figure 1(b)). In sequential adaptation, CPLUS achieves 78.87% average task accuracy, compared with 69.73% for the strongest baseline. In addition, CPLUS becomes more effective as models grow: under limited anchor budgets, the share of SFT forgetting it recovers rises from 46.6% for Qwen3-0.6B to 69.4% for Qwen3-4B. Our contributions are as follows:

- We show, in matched comparisons, that self-probes preserve more prior behavior when used to guide updates than when replayed as training samples.
- We introduce CPLUS, which combines gradients from past samples and self-probes with a smoothed update mask to reduce forgetting under limited past-data access.
- We provide a systematic evaluation across five models, four benchmarks, and six anchor budgets, complemented by broader-capability, sequential, and compute studies.

2 Related Work

Continual Learning. Continual learning methods are commonly grouped into regularization-based approaches, which discourage changes to parameters important for earlier tasks (Kirkpatrick et al., 2017; Aljundi et al., 2018); rehearsal-based approaches, which store and replay past samples (Rolnick et al., 2019; Rebuffi et al., 2017); optimization-based approaches, which modify the update using gradients from past data (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Farajtabar et al., 2020); and expansion-based approaches, which allocate additional capacity to balance stability & plasticity (Yan et al., 2021; Wang et al., 2022; Zhou et al., 2024). These approaches navigate the

trade-off between preserving prior behavior and learning new data (McCloskey & Cohen, 1989), commonly relying on past data or information derived from them (Cho et al., 2025; Dohare et al., 2024). CPL Yang et al. (2026) makes this dependence explicit, showing that parameter-wise gradient analysis can retain updates predicted to learn the new data without interfering with prior behavior, approaching zero forgetting when past data are fully available. Yet full past-data access is rarely possible when adapting open-weight models, which are typically released without their original training data or associated statistics (Zhou et al., 2024; Marek et al., 2026). Note that CPLUS follows this optimization-based line of work. Instead of projecting the whole update as A-GEM (Chaudhry et al., 2019) and OGD (Farajtabar et al., 2020) do, it rescales each coordinate of the proposed update.

Self-Generated Data. The model's own generations provide one source of this additional evidence, but existing methods differ in how they use the generated data. Prior work shows that replay can reduce forgetting and improve data efficiency during adaptation when the replay data approximate the past-data distribution (Bethune et al., 2025; Kotha & Liang, 2026; Marek et al., 2026). Accordingly, generative replay and Self-Synthesized Rehearsal (SSR) seek to approximate unavailable past data and train directly on generated samples (Shin et al., 2017; Huang et al., 2024; Resta & Bacciu, 2024). Self-Distillation Fine-Tuning instead uses a demonstration-conditioned model as an on-policy teacher (Shenfeld et al., 2026). CPLUS uses self-generated data differently. Rather than training on self-probes as replay data, it uses their gradients to identify which parts of a proposed new-data update may interfere with prior behavior. Thus, new data determine what the model learns, while self-probes guide how it learns without disrupting what it already knows. See Appendix G for more.

3 CPLUS: Motivation and Method

This section first shows why limited past-data access weakens preservation (Section 3.1) and then introduces CPLUS (Section 3.2).

3.1 Continual Learning under Limited Past-Data Access

We study continual learning when only A retained past samples, which we call *anchors*, remain available from the broader pool of past data. We examine how both the amount and coverage of this evidence affect the preservation of prior behavior.

The two limitations illustrated in Figure 1 motivate our method. **First,** preservation weakens as fewer anchors are available. Figure 1(a) holds the anchored task fixed and reduces the CommonsenseQA anchors from 418 to four; Retention decreases for both replay and CPL as A shrinks. **Second,** the available anchors represent only a narrow part of a base model's capabilities. Figure 1(b) evaluates five benchmarks used neither for adaptation nor as anchors and shows that preserving the represented target does not necessarily preserve behavior beyond it.

Both limitations arise because all preservation evidence comes from the anchors. When this evidence is sparse, it is hard to tell which changes will preserve prior behavior. When it is narrow, preserving the anchored task does not guarantee preserving other capabilities. The challenge, therefore, is to obtain a broader preservation signal despite limited access to past data.

One way to broaden this signal is to generate self-probes from the base model. Replaying them as training data (Huang et al., 2024), treats the model's predictions as labels even though they may be wrong. Learning from incorrect predictions can reinforce errors through confirmation bias (Arazo et al., 2020), and noisy replay labels can exacerbate forgetting (Kim et al., 2021). We instead treat self-probes as records of the frozen base model's behavior and use their gradients only to assess proposed new data updates.

3.2 Proposed Method

Notation and Preliminaries. Building on the parameter-wise view of CPL (Yang et al., 2026), we let new data propose updates and use anchor and self-probe gradients to assess which changes may disrupt prior behavior. Let f_{θ_0} denote the frozen pretrained base model, and let θ_t denote the adapted parameters before update $t \geq 1$, initialized as $\theta_1 = \theta_0$. Adaptation learns from $\mathcal{D}_{\text{new}}$, with only A retained past samples in the anchor set $\mathcal{P}_A$. Let δ_t denote the optimizer's proposed new-data update

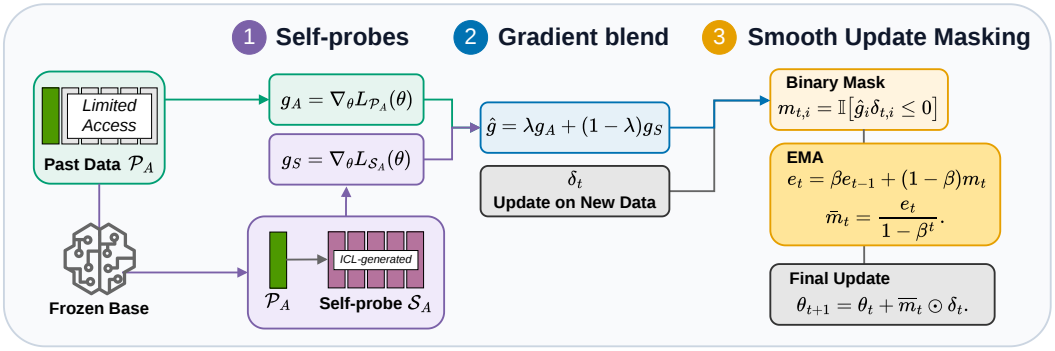

Figure 2: **Overview of CPLUS. 1. Self-probes.** The frozen base model uses the limited anchors $\mathcal{P}_A$ as in-context demonstrations to generate a self-probe bank $\mathcal{S}_A$. **2. Gradient blend.** Gradients from the anchors and self-probes are blended into $\widehat{g}$. **3. Smooth Update Masking.** The blended preservation gradient gates the optimizer's proposed new-data update δ_t parameter-wise, and the binary mask is smoothed by an exponential moving average. Finally, the smoothed mask is applied to δ_t. Thus, self-probes supply preservation evidence rather than replay targets.

at adaptation step t, and let g_A denote the most recently computed mean anchor-loss gradient. The product $g_{A,i}\delta_{t,i}$ estimates parameter i's first-order contribution to the change in anchor loss. We call parameter i *collaborative* when this product is nonpositive and *conflicting* otherwise. These labels depend on the current update and preservation signal; they are not fixed parameter types.

To broaden the evidence used by this criterion, we propose **CPLUS**: Collaborative Parameter Learning Using Self-Probes. CPLUS generates self-probes from the frozen base model and blends their gradients with anchor gradients. It then uses this augmented signal to identify collaborative parameters and smooths the resulting decisions over time. Figure 2 summarizes these three components.

3.2.1 Self-Probes

Before adaptation, CPLUS uses the frozen base model f_{θ_0} to generate self-probes that reflect its prior behavior. For each probe, a small context $C_j \subseteq \mathcal{P}_A$ provides in-context demonstrations (Brown et al., 2020). The model generates a new input $\widetilde{x}_j$ and assigns a prediction $\widetilde{y}_j$ using a fixed decoding rule. Repeating this process produces a self-probe bank $\mathcal{S}_A = \{(\widetilde{x}_j, \widetilde{y}_j)\}_{j=1}^{N}$, which remains fixed during adaptation. The self-probes record the base model's predictions rather than externally verified labels.

During adaptation, the fixed self-probe bank supplies gradients that guide proposed new-data updates. Neither the anchor loss nor the self-probe loss is added to the new-data objective that produces the proposed update δ_t. Their gradients determine how much of that proposal is applied to each parameter. We compare this use of self-probes with direct replay in Figure 5; construction and audit details appear in Sections C and C.2.

3.2.2 Gradient Blending

To combine evidence from the two sources, CPLUS computes mean anchor and self-probe gradients at the beginning of each epoch, using the current adapted model parameters θ:

$$g_A = \nabla_\theta L_{\mathcal{P}_A}(\theta), \qquad g_S = \nabla_\theta L_{\mathcal{S}_A}(\theta). \tag{1}$$

It combines them into a preservation gradient,

$$\widehat{g} = \lambda g_A + (1-\lambda) g_S. \tag{2}$$

We use $\lambda = 1/2$ and do not normalize or rescale either gradient. The two sources therefore receive equal coefficients, but their influence on any particular parameter can differ with gradient magnitude. We reuse g_A, g_S, and $\widehat{g}$ throughout each epoch, while computing the new update δ_t at every step.

This blend provides one preservation signal that accounts for evidence from both sources. By linearity, $\widehat{g}$ is the gradient of $\lambda L_{\mathcal{P}_A} + (1-\lambda) L_{\mathcal{S}_A}$, evaluated at the refresh parameters θ. This weighted preservation loss supplies the signal for masking proposed updates; it is not added to the new-data training objective. Table 1 compares alternative ways to construct the gradients.

3.2.3 SMOOTH UPDATE MASKING

At each adaptation step, CPLUS uses the most recently refreshed $\widehat{g}$ to assess the optimizer's proposed new-data update δ_t. For parameter i, it forms the binary mask

$$m_{t,i} = \mathbb{I}[\widehat{g}_i \delta_{t,i} \leq 0]\,. \tag{3}$$

Here, $m_{t,i} = 1$ identifies a parameter as collaborative with respect to the blended preservation gradient, while $m_{t,i} = 0$ identifies conflict. The criterion uses the optimizer's proposed parameter change, not the raw new-data gradient.

Because δ_t is recomputed at each step while $\widehat{g}$ is refreshed only once per epoch, a parameter's mask can flip from step to step. CPLUS smooths them with a bias-corrected exponential moving average:

$$e_t = \beta e_{t-1} + (1-\beta) m_t, \qquad \overline{m}_t = \frac{e_t}{1-\beta^t}, \tag{4}$$

Here m_t is the binary mask from Equation (3) and $\overline{m}_t$ is its bias-corrected, smoothed version. We set $e_0 = 0$ and $\beta = 0.9$. The applied update is:

$$\theta_{t+1} = \theta_t + \overline{m}_t \odot \delta_t. \tag{5}$$

Intuitively, persistent collaborative decisions move a mask coefficient toward one, while persistent conflicting decisions move it toward zero. Because the coefficients lie in $[0, 1]$, the mask can limit each proposed update but cannot amplify or reverse it. Smoothing also relaxes the instantaneous sign constraint in exchange for less sensitivity to isolated mask flips. Our ablations show that it improves Retention at full Acquisition (Tables 1 and 3). Implementation details appear in Section C.

4 EXPERIMENTS

We evaluate CPLUS under limited data access and beyond the anchored task, compare ways of using self-probes, and then study sequential learning, training cost, and learning dynamics.

4.1 EXPERIMENTAL SETTING

Models and benchmarks. We study within-family scale with Qwen3-0.6B, 1.7B, and 4B (Yang et al., 2025), and test out-of-family generalization with Gemma-2-2B-it (Team et al., 2024) and the Phi-3.5 Mini model (Abdin et al., 2024). The adaptation benchmarks are CommonsenseQA (Talmor et al., 2019), ARC-Challenge (Clark et al., 2018), MMLU (Hendrycks et al., 2021), and MedQA (Jin et al., 2021). Following Yang et al. (2026), we train on initially incorrect examples and measure forgetting on initially correct examples, from which anchors are drawn. This lets us measure what the model learns and what it forgets separately.

We vary the anchor budget over $A \in \{4, 8, 16, 32, 64, 128\}$; plots show the count, whereas tables also report its percentage of the full past dataset. Baselines are standard supervised fine-tuning (SFT), exemplar replay (Rolnick et al., 2019), self-probe replay (Resta & Bacciu, 2024), and CPL (Yang et al., 2026) under matched data access. Frozen distillation (Hinton et al., 2015), Self-Synthesized Rehearsal (SSR) (Huang et al., 2024), A-GEM (Chaudhry et al., 2019), and OGD-GTL (Farajtabar et al., 2020) appear only in the compute comparison (Figure 7). Ablations appear in Table 1.

Hyperparameter protocol. Following the two-phase protocol of Cha & Cho (2025), we use Qwen3-0.6B on CommonsenseQA for development. We retain CPL's 25-epoch training budget (Yang et al., 2026) and use `lr=1e-6` by default. The CPLUS-specific defaults are $N = 512$ self-probes, two in-context examples, generation temperature 0.8, equal-weight mean-gradient blending, no norm alignment, and mask EMA $\beta = 0.9$. We keep the self-probe and masking settings fixed across models, benchmarks, and anchor budgets. The compute comparison uses 32 self-probes; other main comparisons use the default bank.

Evaluation. Retention (Correct $\rightarrow$ Correct) is $\Pr(\text{correct after} \mid \text{correct before})$, and Acquisition (Incorrect $\rightarrow$ Correct) is $\Pr(\text{correct after} \mid \text{incorrect before})$. For sequential adaptation, Average Task Accuracy is the unweighted mean accuracy over tasks encountered so far. Primary single-task and broader-capability comparisons report means and standard errors over five runs.

4.2 Preservation with Limited Anchors

We first evaluate whether CPLUS reduces forgetting under limited data access. In Figure 3, CPLUS achieves higher Retention than the strongest baseline in 70 of 72 Qwen3 settings, including all 24 at $A \in \{4, 8\}$. At these budgets, its mean gain is 7.73 points, while recovery of SFT forgetting rises from 46.6% at 0.6B to 50.6% at 1.7B and 69.4% at 4B (Figure 10). CPLUS also leads in all 48 Gemma and Phi-3.5 Mini settings. All methods in the grid attain 100% Acquisition, yet differ in Retention even when the adaptation examples are fully learned.

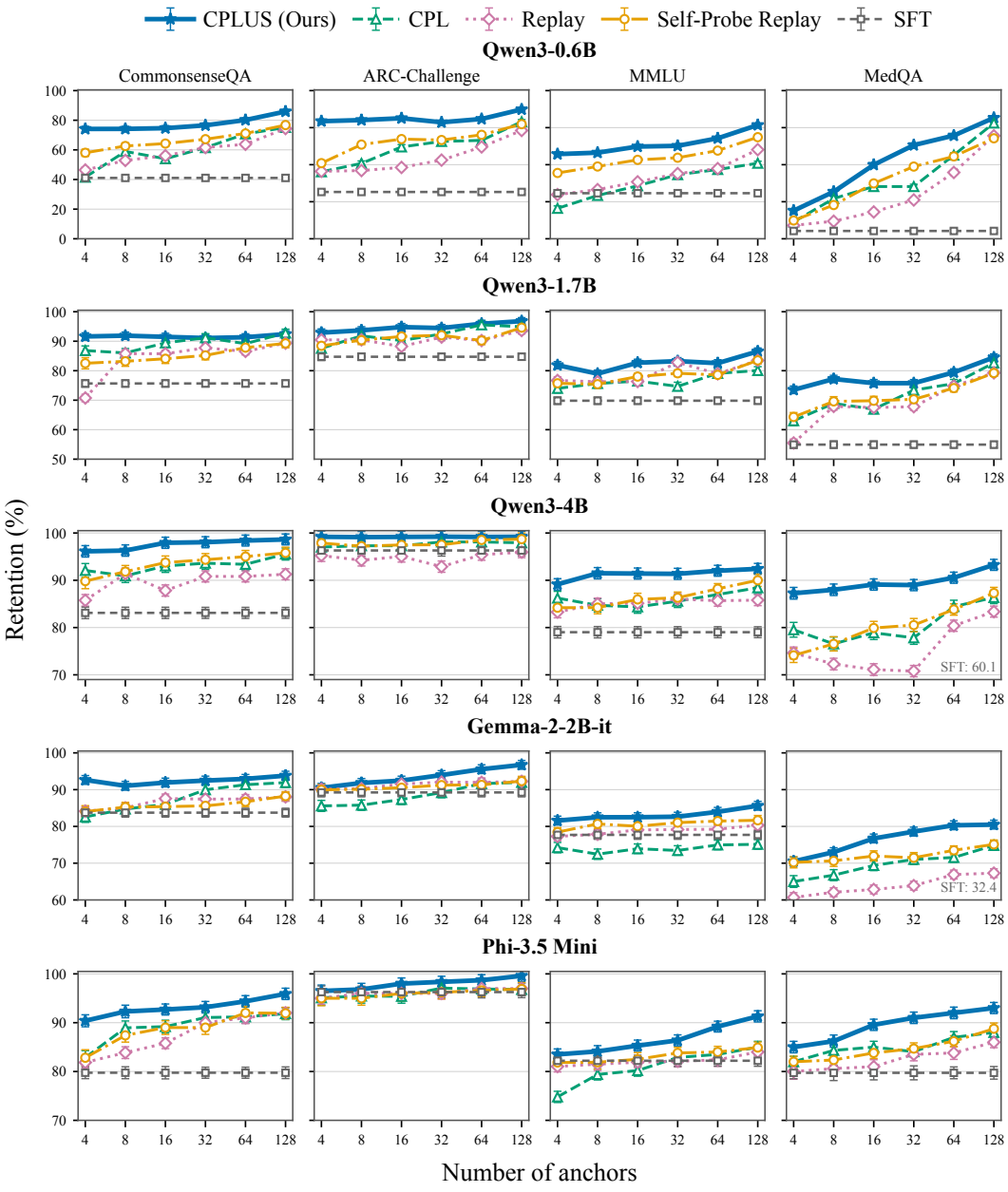

Figure 3: **CPLUS reduces forgetting even with few past examples.** Curves show **CPLUS (Ours)**, **CPL**, **Replay**, **Self-Probe Replay**, and **SFT** across anchor budgets. Retention is the percentage of initially correct examples that remain correct after adaptation (Correct → Correct). Error bars show five-seed standard errors. Panels share a vertical scale within each model row. Off-scale MedQA SFT values are annotated in the 4B and Gemma rows.

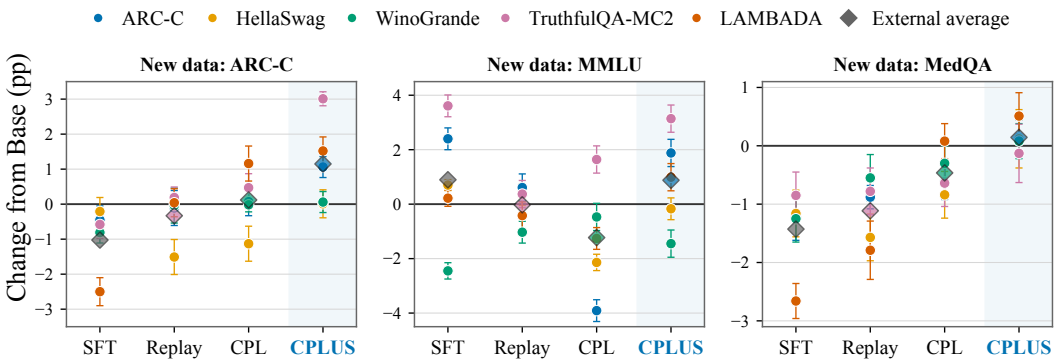

Figure 4: **CPLUS preserves broader performance on other tasks while learning new data.** Panels show changes from the base model after ARC-C, MMLU, or MedQA adaptation. Diamonds are unweighted averages over benchmarks external to the adaptation task; in the ARC-C panel, its own point is excluded from that average. Points and error bars show means and standard errors over five seeds. Each panel has its own labeled vertical range.

4.3 Preservation Beyond the Represented Target

Retention on the anchored task does not establish preservation elsewhere. After CommonsenseQA adaptation, the right panel of Figure 1 shows broader losses with CPL on four of five benchmarks, while CPLUS stays near or above the base model on each.

The pattern extends to three other adaptation tasks (Figure 4). On benchmarks external to each new task, CPLUS improves average performance relative to the base model by 1.15, 0.88, and 0.14 percentage points after ARC-C, MMLU, and MedQA adaptation, respectively.. It has the highest external average after ARC-C and MedQA adaptation and is within 0.02 points of SFT after MMLU adaptation.

We also test how anchor composition affects preservation and error correction by varying whether anchors were initially answered correctly (Section E.4 and Table 4).

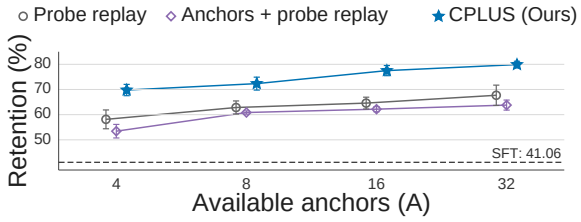

Figure 5: **Self-probes work better as gradients than replay.** Mean Retention and standard error over three matched seeds on Qwen3-0.6B/CommonsenseQA. Methods use the same probes and update budget.

Variant	Retention (%)
Anchor gradient only	55.13
Probe gradient only	71.49
CPLUS without EMA	78.71
CPLUS	**79.96**
CPLUS + norm alignment	79.94

Table 1: **Component ablation.** Qwen3-0.6B, CommonsenseQA, $A = 8$; one seed. All variants attain 100% Acquisition.

4.4 How Should Self-Probes Be Used?

We compare alternative uses of the same self-probes. Across four anchor budgets, CPLUS retains more previously correct answers than either replay variant (Figure 5). Here, Probe replay trains on frozen-model labels, while anchor + probe replay additionally includes the available real anchors. All methods attain near-perfect Acquisition.

We next isolate the contributions of anchors, self-probes, and smoothing (Table 1). Combining both gradient sources performs best in this diagnostic; smoothing adds a smaller gain, while norm alignment changes little.

We further test whether smaller updates alone explain the gains. Neither matched update-norm nor randomized-mask controls reproduce CPLUS's Retention at full Acquisition, suggesting that coordinate selection matters beyond aggregate attenuation in this diagnostic (Table 6 and Section E.7).

4.5 Sequential Multi-Task Adaptation

We next test whether these benefits persist when learning multiple tasks sequentially. We adapt to CommonsenseQA, ARC-Challenge, MMLU, and MedQA in that order, with eight anchors per preceding task. After the final stage, CPLUS achieves 78.87% Average Task Accuracy, compared with 69.73% for replay and 69.65% for CPL; it also has the highest accuracy on every task in this sequence (Figure 6). Full results appear in Tables 7 and 8.

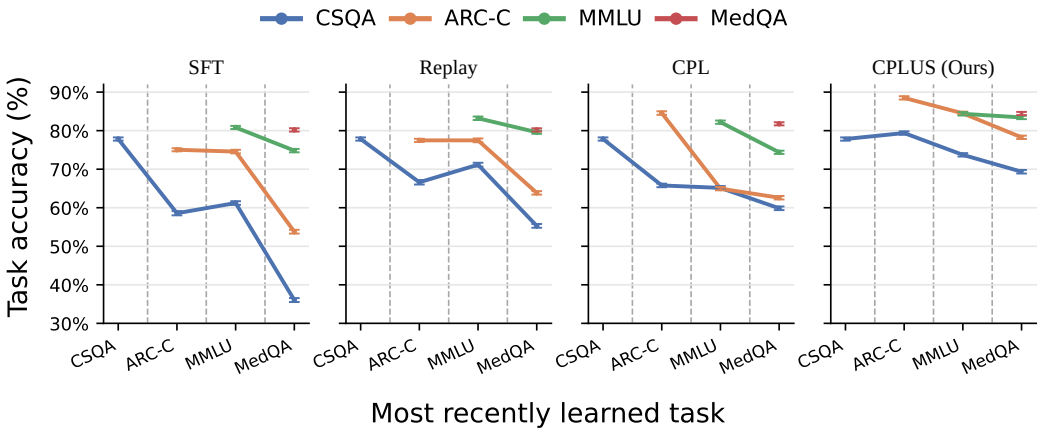

Figure 6: **CPLUS in limited-anchor sequential multi-task adaptation.** Training follows CSQA → ARC-C → MMLU → MedQA, carrying forward each checkpoint; MMLU thus inherits CSQA and ARC-C training. The CSQA checkpoint is shared. Curves show accuracy on all encountered tasks (mean and standard error, three seeds). Replay, CPL, and CPLUS use $A = 8$ anchors per past task.

4.6 Efficiency and Optimization Robustness

Training cost. We examine the compute–Retention tradeoff using a smaller self-probe bank. Figure 7 compares Retention gain against training FLOPs. Across the displayed model sizes, CPLUS achieves a larger gain than the other evaluated methods.

Learning dynamics. Finally, we examine how learning and forgetting evolve during adaptation. In an independent limited-anchor diagnostic, CPLUS first reaches 100% Acquisition after five epochs, compared with ten for CPL, and thereafter maintains the highest Retention (Figure 8). Replay initially forgets less than SFT and remains above it after 25 epochs, but trails CPLUS in Retention even though both reach full Acquisition by epoch five. Its gap therefore reflects weaker preservation, not slower acquisition. The trajectories also reveal a sharp early drop in Retention followed by partial recovery; Section E.3 examines this pattern in more detail. Section E.5 tests robustness under a larger-learning-rate schedule. These trajectories distinguish temporary forgetting from the loss that remains after adaptation.

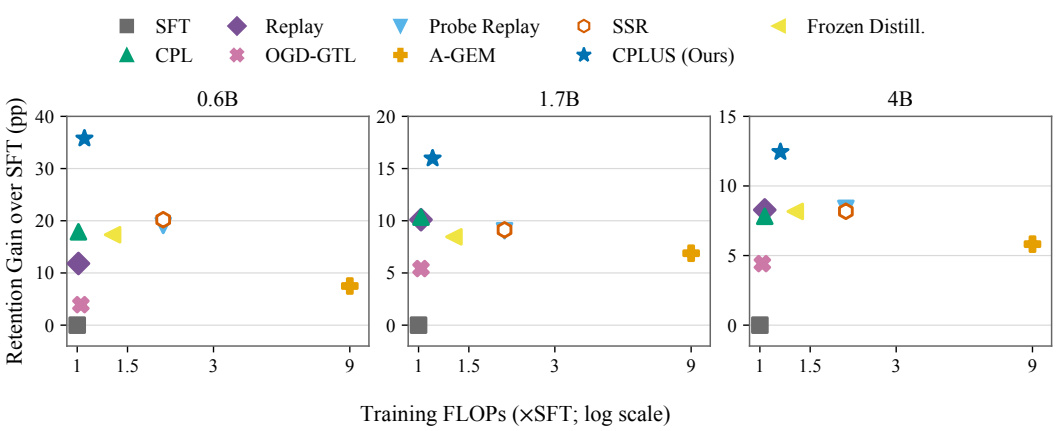

Figure 7: **CPLUS offers a favorable compute–Retention tradeoff across the evaluated model scales.** On CommonsenseQA, the axes report Retention gain over SFT and training FLOPs relative to SFT (log scale). Preservation methods use eight anchors; SFT uses none. One-time preparation is excluded. App. E.12 gives the accounting and baseline definitions. Y-axis scales differ.

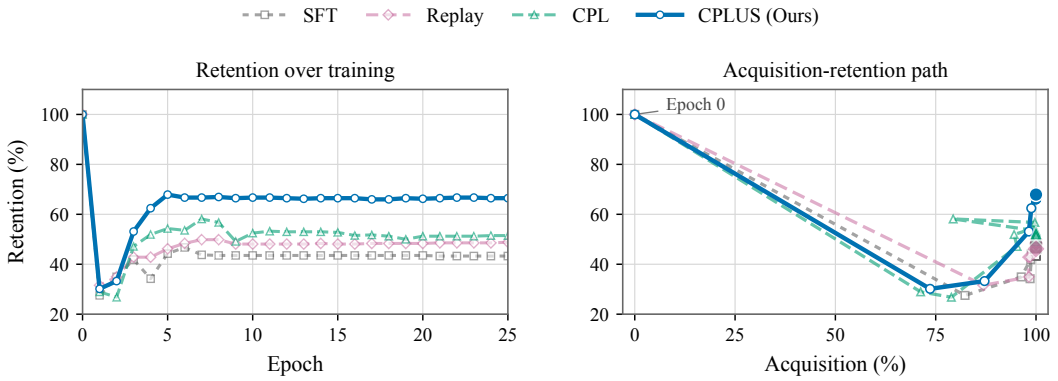

Figure 8: **Early forgetting can be followed by recovery as training continues.** Retention over epochs (left) and the Acquisition–Retention path (right) for a 25-epoch Qwen3-0.6B/CommonsenseQA diagnostic with eight anchors. The lines show epoch-level scores without uncertainty bands. This diagnostic is independent of the five-run comparison in Figure 3. Filled markers indicate the first epoch reaching 100% Acquisition.

5 Conclusion

We introduced CPLUS, a continual-learning method for adapting foundation models when only a limited anchor set is available. CPLUS elicits self-probes from a frozen base model and uses their gradients, together with anchor gradients, as preservation signals that softly constrain updates on new data. Across model sizes, benchmarks, and anchor budgets, CPLUS substantially improves Retention while maintaining Acquisition, including beyond the task represented by the anchors and in a tested four-task sequence. These results show that self-probes need not be replayed as training examples: they can instead provide actionable evidence for preserving prior behavior during continual adaptation under limited access to past data.

AI USE STATEMENT

Generative AI tools assisted with manuscript writing and editing, literature retrieval and discovery, experimental implementation, feedback on experimental design, and result interpretation. Frozen base models generated the synthetic self-probe data described in Section 3.2. The authors manually audited the central algorithms, evaluation protocol, and reported results, and verified every reference against its original paper and a persistent publisher or arXiv record. The authors made all final scientific decisions and take responsibility for this work.

REPRODUCIBILITY STATEMENT

We will publicly release the source code and exact experimental protocols, including data preparation, self-probe generation, training configurations, evaluation scripts, hyperparameters, and random seeds. We use the public benchmark datasets distributed through Hugging Face and identify the dataset configurations and splits used in each experiment. The main text specifies the evaluation protocol, while the appendix provides the complete algorithm, implementation details, ablations, and additional results needed to reproduce the reported findings.

ETHICS STATEMENT

This work introduces a method for preserving prior model behavior during continued adaptation. We collected no new human-subject or private personal data and used publicly available models and benchmark datasets. Because self-probes represent a model's existing behavior rather than verified truth, they may preserve errors or biases present in the base model; likewise, incorrect new data may introduce harmful behavior. Practical use should therefore include validation of new supervision and post-adaptation audits appropriate to the deployment domain. Results on MedQA are benchmark results and should not be interpreted as evidence of clinical readiness.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget, 2018. URL https://arxiv.org/abs/1711.09601.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning, 2020. URL https://arxiv.org/abs/1908.02983.

Louis Bethune, David Grangier, Dan Busbridge, Eleonora Gualdoni, Marco Cuturi, and Pierre Ablin. Scaling laws for forgetting during finetuning with pretraining data injection. *arXiv preprint arXiv:2502.06042*, 2025. URL https://arxiv.org/abs/2502.06042.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Sungmin Cha and Kyunghyun Cho. Hyperparameters in continual learning: A reality check. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://arxiv.org/abs/2403.09066.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *The Seventh International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1812.00420.

Dongkyu Cho, Taesup Moon, Rumi Chunara, Kyunghyun Cho, and Sungmin Cha. Forget forgetting: Continual learning in a world of abundant memory. *arXiv preprint arXiv:2502.07274*, 2025. URL https://arxiv.org/abs/2502.07274.

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. URL https://arxiv.org/abs/1803.05457.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632: 768–774, 2024. doi: 10.1038/s41586-024-07711-7. URL https://www.nature.com/articles/s41586-024-07711-7.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3762–3773, 2020. URL https://arxiv.org/abs/1910.07104.

Md Yousuf Harun, Jhair Gallardo, Tyler L. Hayes, and Christopher Kanan. How efficient are today's continual learning algorithms? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2431–2436, 2023. URL https://openaccess.thecvf.com/content/CVPR2023W/CLVision/html/Harun_How_Efficient_Are_Todays_Continual_Learning_Algorithms_CVPRW_2023_paper.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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal, 2024. URL https://arxiv.org/abs/2403.01244.

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*, 2021. URL https://arxiv.org/abs/2009.13081.

Chris Dongjoo Kim, Jinseo Jeong, Sangwoo Moon, and Gunhee Kim. Continual learning on noisy data streams via self-purified replay, 2021. URL https://arxiv.org/abs/2110.07735.

Jiyeon Kim, Hyunji Lee, Hyowon Cho, Joel Jang, Hyeonbin Hwang, Seungpil Won, Youbin Ahn, Dohaeng Lee, and Minjoon Seo. Knowledge entropy decay during language model pretraining hinders new knowledge acquisition. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f8f3250f3f31387906e49aec41602f36-Abstract-Conference.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. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://arxiv.org/abs/1612.00796.

Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard, 2020. URL https://arxiv.org/abs/2006.05188.

Suhas Kotha and Percy Liang. Replaying pre-training data improves fine-tuning. *arXiv preprint arXiv:2603.04964*, 2026. URL https://arxiv.org/abs/2603.04964.

Matthias De Lange, Gido van de Ven, and Tinne Tuytelaars. Continual evaluation for lifelong learning: Identifying the stability gap, 2023. URL https://arxiv.org/abs/2205.13452.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.08840.

Martin Marek, Dongkyu Cho, Shikai Qiu, Rumi Chunara, Pavel Izmailov, and Andrew Gordon Wilson. Forgetting in language models: Capacity, optimization, and self-generated replay. *arXiv preprint arXiv:2605.26097*, 2026. URL https://arxiv.org/abs/2605.26097.

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

Ameya Prabhu, Hasan Abed Al Kader Hammoud, Puneet K. Dokania, Philip H. S. Torr, Ser-Nam Lim, Bernard Ghanem, and Adel Bibi. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3698–3707, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Prabhu_Computationally_Budgeted_Continual_Learning_What_Does_Matter_CVPR_2023_paper.html.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning, 2017. URL https://arxiv.org/abs/1611.07725.

Michele Resta and Davide Bacciu. Self-generated replay memories for continual neural machine translation, 2024. URL https://arxiv.org/abs/2403.13130.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning, 2019. URL https://arxiv.org/abs/1811.11682.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025. URL https://arxiv.org/abs/2509.04259.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026. URL https://arxiv.org/abs/2601.19897.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay, 2017. URL https://arxiv.org/abs/1705.08690.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL-HLT*, 2019. URL https://arxiv.org/abs/1811.00937.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. FOSTER: Feature boosting and compression for class-incremental learning, 2022. URL https://arxiv.org/abs/2204.04662.

Shipeng Yan, Jiangwei Xie, and Xuming He. DER: Dynamically expandable representation for class incremental learning, 2021. URL https://arxiv.org/abs/2103.16788.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

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*, 2026. URL https://arxiv.org/abs/2601.21577.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey, 2024. URL https://arxiv.org/abs/2401.16386.

A Experiment Index

Table 2: **Experiment index.**

ID	Question	Location
E1	Does CPLUS reduce forgetting when anchors are limited?	Figure 3 and Section E.1
E2	Does preservation extend beyond the behavior represented by the anchors?	Figure 4 and Section E.10
E3	Are self-probes more effective as preservation gradients than as replay targets?	Figure 5
E4	Which CPLUS components and configurations matter?	Table 1 and Section E.2
E5	Does CPLUS remain effective in the tested task sequence?	Figure 6 and Section E.9
E6	How do Acquisition, Retention, and computational cost evolve during adaptation?	Figure 7 and Section E.3
E7	How does anchor composition affect preservation?	Section E.4
E8	Does CPLUS remain effective under an aggressive update schedule?	Section E.5
E9	Does update location matter beyond update magnitude or mask density?	Sections E.6 and E.7
E10	How does the learned mask relate to full-data and layer-wise references?	Sections E.8 and E.11

B Additional Problem-Formulation Details

Let $\mathcal{P}$ denote the unavailable full set of past examples, $g_{\mathcal{P},t}$ its loss gradient at adaptation step t, and $\widehat{g}_t$ an estimate from available evidence. For the instantaneous mask, before temporal smoothing, let

$$\widehat{m}_{t,i} = \mathbb{I}[\widehat{g}_{t,i}\delta_{t,i} \le 0], \qquad \widehat{\delta}_t = \widehat{m}_t \odot \delta_t.$$

The masked update satisfies $\widehat{g}_t^\top \widehat{\delta}_t \le 0$, but its first-order effect on the full past-data loss obeys

$$\begin{aligned} g_{\mathcal{P},t}^\top \widehat{\delta}_t &= \widehat{g}_t^\top \widehat{\delta}_t + (g_{\mathcal{P},t} - \widehat{g}_t)^\top \widehat{\delta}_t \\ &\le \|g_{\mathcal{P},t} - \widehat{g}_t\|_2 \|\widehat{\delta}_t\|_2. \end{aligned}$$

Thus, forgetting may arise from estimation error or from a large update in directions where the preservation estimate is inaccurate. Shrinking the update to zero is not a satisfactory solution because it also prevents acquisition.

Because the mask acts on the optimizer update, the relevant errors depend on that proposed update. Define

$$c_{t,i} = g_{\mathcal{P},t,i}\delta_{t,i}, \qquad [z]_+ = \max(z, 0).$$

An estimated mask admits harmful update mass when it retains a coordinate with $c_{t,i} > 0$, and removes useful update mass when it suppresses a coordinate with $c_{t,i} < 0$. We therefore use

$$E_{\text{harmful-admitted}} = \sum_i \widehat{m}_{t,i}[c_{t,i}]_+, \qquad E_{\text{safe-removed}} = \sum_i (1 - \widehat{m}_{t,i})[-c_{t,i}]_+.$$

These diagnostics distinguish preservation obtained by selectively blocking harmful updates from preservation obtained by suppressing most update energy.

C Detailed CPLUS Method

This section expands the three components summarized in Section 3.2. Throughout, let f_{θ_0} be the frozen base model, $\mathcal{D}_{\text{new}}$ the new-data set, and $\mathcal{P}_A$ the limited anchor set of size A.

C.1 Constructing the Self-Probe Bank

For each probe request j, we sample a context $C_j \subseteq \mathcal{P}_A$ containing K anchors. The frozen model uses this context to approximate the task and generate a new input. Optionally, generation can include a generic topic cue c_j drawn from 24 predefined broad areas; the reported categorical runs used this option. The resulting generation is

$$\widetilde{x}_j \sim q_{\theta_0}(\cdot \mid C_j, c_j; T),$$

where T is the generation temperature and $c_j = \varnothing$ when no cue is used. We then apply the fixed decoding rule of the base model to obtain a behavioral target,

$$\widetilde{y}_j = h_{\theta_0}(\widetilde{x}_j).$$

The target therefore records how the base model responds; it is not an externally verified label. Outputs that do not satisfy the required task interface are discarded, and duplicate probe inputs are removed. Generation continues until the bank contains N valid pairs,

$$\mathcal{S}_A = \{(\widetilde{x}_j, \widetilde{y}_j)\}_{j=1}^{N}.$$

In a one-seed diagnostic, including the topic cue did not consistently improve Retention.

The same frozen model supplies both the in-context generation distribution and the behavioral targets. The bank is constructed before adaptation, cached, and held fixed throughout a run. Neither $\mathcal{P}_A$ nor $\mathcal{S}_A$ is added to the adaptation objective. They are used only to estimate which components of an adaptation update should be preserved. In the reported configuration, $K = 2$, $T = 0.8$, and $N = 512$.

Self-probes are intended to broaden behavioral coverage, not to reconstruct the unavailable past-data distribution. Their usefulness depends on what the frozen model can infer from the context. Model scale may also affect baseline susceptibility to forgetting; Section E.1 therefore reports both absolute Retention and recovery relative to that baseline.

C.2 Self-Probe Bank Audit

We audit the sealed self-probe banks for duplication, model confidence, and overlap with the evaluation questions. Across three banks, all 1,536 normalized probe prompts are unique. The frozen model's assigned behavioral targets have mean, median, and tenth-percentile confidence of 0.915, 0.984, and 0.696, respectively. A post-hoc comparison with 1,096 evaluation prompts finds one normalized exact match (0.07%) and two probes with token Jaccard similarity of at least 0.9 (0.13%). These targets are frozen-model behavioral predictions rather than independently verified ground-truth annotations.

C.3 Why Use Self-Probes as Preservation Evidence?

The key design choice is how generated data enter adaptation. Direct self-generated replay treats the model outputs as additional training targets. For example, an augmented replay objective would form a gradient of the form

$$g_{\text{replay}} = g_{\text{new}} + \alpha\big(\lambda g_A + (1-\lambda) g_S\big),$$

where g_{new} is the new-data gradient and α controls rehearsal. The probe gradient can therefore create an optimization direction even when it is unrelated to acquiring the new behavior. Repeated replay may also reinforce systematic errors inherited from the frozen model.

CPLUS instead computes the optimizer proposal δ_{new} using only the supervised adaptation objective and uses the probes to control that proposal:

$$\delta_{\text{CPLUS}} = \overline{m}(g_A, g_S, \delta_{\text{new}}) \odot \delta_{\text{new}}, \qquad \overline{m}_i \in [0, 1].$$

Consequently, a self-probe cannot introduce, amplify, or reverse an update coordinate; it can only attenuate a component of the supervised adaptation update. This makes probe use conditional on the update that is actually being proposed, rather than rehearsing every generated example indiscriminately. It also keeps acquisition supervision separate from preservation evidence.

To test this distinction, we compare three ways to use each sealed self-probe bank across limited anchor budgets: probe replay, anchor + probe replay, and CPLUS. Probe replay trains on frozen-model argmax labels. Anchor + probe replay also receives the same real anchors and 512 self-probes

as CPLUS; anchors are repeated to balance their aggregate sampling weight against the probes. CPLUS instead uses anchor and probe gradients to construct its preservation mask. Each seed uses its own matched bank, and all methods share the update budget. Figure 5 summarizes the matched comparison across anchor budgets.

C.4 Estimating and Blending Preservation Gradients

Let τ_r be the first adaptation step of epoch r. At that epoch's preservation refresh, CPLUS evaluates two losses at θ_{τ_r}. Here we show the refresh index explicitly; the main text omits it. The past-example loss is

$$\mathcal{L}_A(\theta_{\tau_r}) = \frac{1}{A} \sum_{(x,y)\in\mathcal{P}_A} \ell(f_{\theta_{\tau_r}}(x), y),$$

and the self-probe loss is

$$\mathcal{L}_S(\theta_{\tau_r}) = \frac{1}{N} \sum_{(\widetilde{x},\widetilde{y})\in\mathcal{S}_A} \ell(f_{\theta_{\tau_r}}(\widetilde{x}), \widetilde{y}).$$

Their mean gradients are

$$g_{A,r} = \nabla_\theta \mathcal{L}_A(\theta_{\tau_r}), \qquad g_{S,r} = \nabla_\theta \mathcal{L}_S(\theta_{\tau_r}).$$

The first source is verified but sparse; the second is broader but inherits the behavior of the base model. CPLUS does not require either gradient to be an unbiased estimate of the unavailable complete-set gradient.

CPLUS combines the two sources before constructing an update mask:

$$\widehat{g}_r = \lambda g_{A,r} + (1-\lambda) g_{S,r}.$$

We use $\lambda = \frac{1}{2}$. This is a raw blend: we apply neither norm alignment nor source-specific rescaling before addition. Equal coefficients do not imply equal coordinate-wise influence because the source gradients may differ in magnitude and structure. The norm-aligned alternative is evaluated as an ablation rather than included in the default method.

C.5 Masking the Optimizer Update

Let $\mathcal{L}_{\text{new}}$ be the new-data objective. Adaptation steps start at $t = 1$, with $\theta_1 = \theta_0$. At step t, the optimizer uses its complete internal state—including adaptive preconditioning and momentum where applicable—to propose an update

$$\delta_t = \mathrm{OPT}_t(\nabla_\theta \mathcal{L}_{\text{new}}(\theta_t)).$$

Thus, without CPLUS, the parameters would change as $\theta_{t+1} = \theta_t + \delta_t$. CPLUS masks this proposed update rather than the raw new-data gradient.

Let $r(t)$ denote the epoch containing step t, whose gradient was refreshed at $\tau_{r(t)}$. The instantaneous mask is

$$m_{t,i} = \mathbb{I}\left[\widehat{g}_{r(t),i}\delta_{t,i} \le 0\right].$$

A coordinate is retained when its contribution to the loss linearization at the latest refresh is nonpositive. Consequently,

$$\left(\widehat{g}_{r(t)}\right)^\top (m_t \odot \delta_t) \le 0.$$

This is CPL's coordinate-wise first-order decision, applied to the blended preservation gradient. Following CPL's released implementation (Yang et al., 2026), we refresh the preservation gradients once per epoch and reuse their blend within that epoch. The update mask is recomputed at every adaptation step.

C.6 Temporal Smoothing

Binary decisions may fluctuate when the preservation signal is estimated from limited data. CPLUS maintains an exponential moving average of the instantaneous masks,

$$e_t = \beta e_{t-1} + (1-\beta) m_t, \qquad e_0 = 0, \tag{6}$$

$$\overline{m}_t = \mathrm{clip}\left(\frac{e_t}{1-\beta^t}, 0, 1\right), \tag{7}$$

and applies

$$\theta_{t+1} = \theta_t + \overline{m}_t \odot \delta_t.$$

We use $\beta = 0.9$. Each effective coefficient remains in $[0, 1]$, so the soft mask can attenuate a proposed update but cannot amplify it or reverse its direction. Values near one indicate repeated compatibility with the preservation signal; values near zero indicate persistent conflict. This is an operational interpretation of the mask, not a claim that individual coordinates have an identifiable semantic meaning.

Temporal smoothing is a controlled relaxation rather than a strict per-step safety guarantee. Because the EMA includes earlier decisions, a coordinate rejected by the current binary mask can retain a nonzero coefficient. The nonpositive inner-product condition above holds for $m_t \odot \delta_t$ with the cached preservation gradient, but not necessarily for the EMA-smoothed update. After the refresh step, the cached gradient need not equal the gradient at the current parameters. We evaluate this stability–constraint trade-off using update-weighted mask diagnostics.

C.7 Complete Procedure and Cost

Self-probe generation is a one-time operation. During adaptation, CPLUS refreshes the two mean preservation gradients at the beginning of each epoch, forms one blended gradient, and reuses it for the adaptation updates in that epoch. The model is optimized only on $\mathcal{D}_{\text{new}}$; the anchors and self-probes influence adaptation exclusively through the mask.

Relative to ordinary fine-tuning, each refresh adds forward and backward passes over $A+N$ reference examples. CPLUS stores one blended preservation gradient and one EMA state with the parameter shapes being masked; the two source gradients are required only transiently during fusion. Algorithm Algorithm 1 gives the complete procedure.

D CPLUS Pseudocode

Algorithm 1 CPLUS

Require: Base-model parameters θ_0, new data $\mathcal{D}_{\text{new}}$, anchors $\mathcal{P}_A$, N, K, T, λ, β

Freeze f_{θ_0} and construct self-probes $\mathcal{S}_A$
Initialize $t \leftarrow 1$, $\theta_1 \leftarrow \theta_0$, and $e_0 \leftarrow 0$
for epoch $r = 1, \ldots, R$ **do**
  $\tau_r \leftarrow t$
  Compute mean gradient $g_{A,r}$ from $\mathcal{P}_A$ at θ_{τ_r}
  Compute mean gradient $g_{S,r}$ from $\mathcal{S}_A$ at θ_{τ_r}
  $\widehat{g}_r \leftarrow \lambda g_{A,r} + (1-\lambda) g_{S,r}$
  for new-data minibatch $B \subset \mathcal{D}_{\text{new}}$ **do**
    $\delta_t \leftarrow \mathrm{OPT}_t(\nabla_\theta \mathcal{L}_B(\theta_t))$
    $m_t \leftarrow \mathbb{I}[\widehat{g}_r \odot \delta_t \le 0]$
    $e_t \leftarrow \beta e_{t-1} + (1-\beta) m_t$
    $\overline{m}_t \leftarrow \mathrm{clip}(e_t/(1-\beta^t), 0, 1)$
    $\theta_{t+1} \leftarrow \theta_t + \overline{m}_t \odot \delta_t$
    $t \leftarrow t + 1$
  end for
end for
return θ_t

E Additional Analyses

The Qwen3 and Gemma evaluation grids use JAX/BF16 on TPUs. The Phi-3.5 Mini out-of-family grid is a separate Torch/BF16 replication on two A100 GPUs with FSDP and learning rate `1e-5`; the default learning rate is `1e-6`. Each auxiliary diagnostic below keeps its execution setup fixed across the compared methods; exact environments are recorded with the released configurations.

E.1 Model-Scale Analyses

Figure 9 reports the access-matched comparison between CPL and CPLUS with four anchors on MMLU and MedQA. Absolute Retention increases with model size for both methods. We interpret the comparison as a within-family trend.

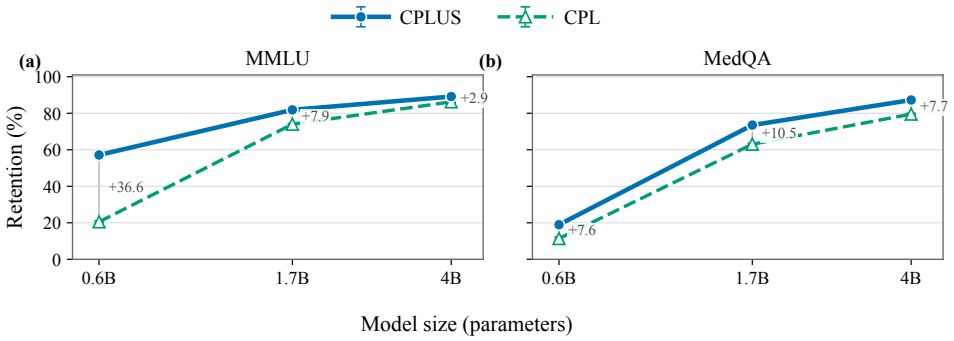

Figure 9: **Retention under limited anchor access across model scale.** Retention denotes Correct → Correct. We compare access-matched CPL and CPLUS with four anchors on MMLU and MedQA. Labels report the absolute Retention improvement of CPLUS over CPL; points and error bars show means and standard errors. Every setting attains 100% Acquisition (Incorrect → Correct).

The complementary four-benchmark perspective appears in Figure 10. Its headroom-normalized panel accounts for the smaller amount of native forgetting available at larger model sizes and shows the fraction of that forgetting recovered by CPLUS.

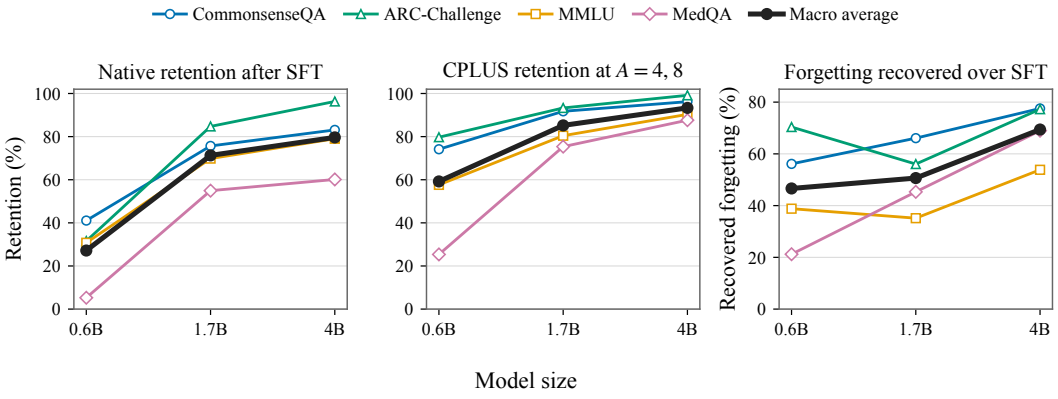

Figure 10: **Model-scale perspective.** Colored lines show individual benchmarks and the black line is their macro average. Retention denotes Correct → Correct. **Left:** Retention after unconstrained fine-tuning, measuring each model's native vulnerability to forgetting. **Center:** CPLUS Retention averaged over $A \in \{4, 8\}$. **Right:** the fraction of SFT forgetting recovered by CPLUS, computed as $100(R_{\text{CPLUS}} - R_{\text{SFT}})/(100 - R_{\text{SFT}})$, with CPLUS Retention averaged over the same two values of A. Every CPLUS run in the fixed grid attains 100% Acquisition (Incorrect → Correct). Because each model induces its own mastered and new-data sets, the right panel measures the share of each model's available forgetting headroom that CPLUS recovers.

E.2 Hyperparameter Selection

We select CPLUS hyperparameters using only the Qwen3-0.6B CommonsenseQA development scenario. Figure 11 summarizes the generation and source-weight searches. With $A = 4$, two-example generation at $T = 0.8$ achieves the highest observed Retention in the generation screen. Equal-weight blending also outperforms self-probe-only preservation and 75:25 blending at every tested anchor budget.

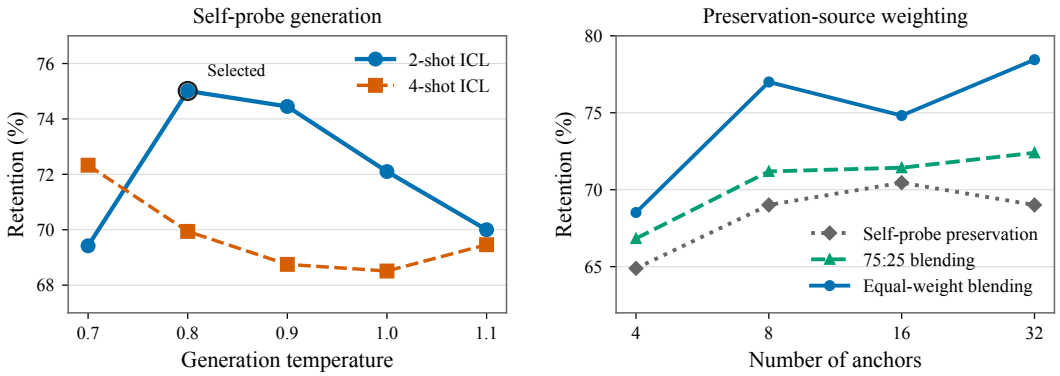

Figure 11: **Development-only hyperparameter searches.** Retention denotes Correct → Correct. **Left:** Retention for two- and four-example in-context generation across sampling temperatures, evaluated with Qwen3-0.6B on CommonsenseQA at $A = 4$ and $N = 512$. The selected configuration uses two examples and $T = 0.8$. **Right:** Retention across $A \in \{4, 8, 16, 32\}$ for self-probe-only preservation, 75:25 blending, and equal-weight blending under a separate development protocol. Equal-weight blending is strongest at all four budgets. All runs achieve 100% Acquisition (Incorrect → Correct).

The separate reference-reduction and smoothing screen in Table 3 favors mean reduction with $\beta = 0.9$, the default used throughout the paper.

Table 3: **Reference reduction and mask smoothing.** Development results with $A = 64$, averaged over three runs.

Reduction	EMA	Retention (%)	Acquisition (%)
Mean	None	81.11 ± 2.95	100
Mean	0.9	$\mathbf{82.32 \pm 0.64}$	100
CVaR	0.9	77.40 ± 0.28	100

E.3 Epoch-Level Acquisition–Retention Dynamics

To examine how methods approach their final solutions, we evaluate Acquisition and Retention after every epoch. This separate Torch/FP32 diagnostic uses one A100 per run, rather than the JAX/BF16 TPU setup of the Qwen3 five-run grid; its endpoints are not the endpoints of Figure 3. Figure 12 shows Qwen3-0.6B on CommonsenseQA with $A = 8$, `lr=1e-6`, and 25 epochs.

The trajectories reveal a sharp early drop in Retention as Acquisition rises rapidly, followed by partial recovery during continued training. This pattern is consistent with the previously reported *stability gap* (Lange et al., 2023). The methods differ in both how quickly they learn the new examples and how much prior behavior they recover: CPLUS reaches complete Acquisition by epoch 5 and maintains a higher Retention plateau, whereas CPL first reaches complete Acquisition at epoch 10.

These trajectories distinguish temporary forgetting from the forgetting that remains at the end of adaptation. An early evaluation alone can miss subsequent recovery, while final Retention alone can hide substantial temporary forgetting. Reaching complete Acquisition also does not imply equal recovery of prior behavior: methods that fully learn the adaptation examples can still settle at different Retention levels.

E.4 Anchor Composition Across Tasks

We test whether preservation anchors must be examples the base model already answers correctly. We adapt Qwen3-0.6B to MedQA with CPLUS while fixing an $A = 8$ budget of ground-truth-labeled CommonsenseQA anchors. Across rows in Table 4, only the composition of this small anchor set changes; "correct" and "wrong" denote the base model's prediction before adaptation. Retention and

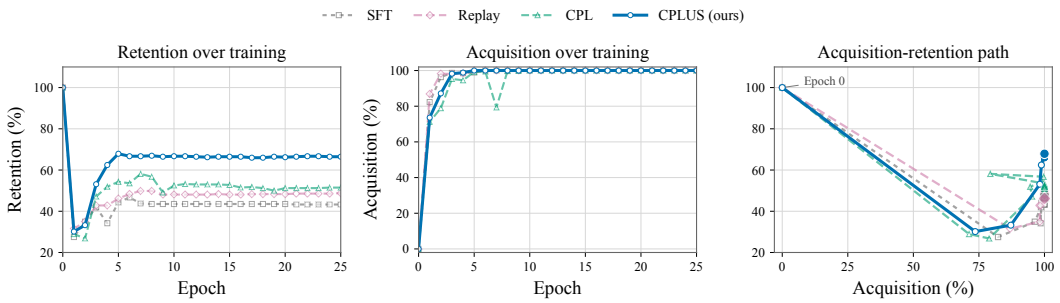

Figure 12: **Epoch-level Acquisition–Retention dynamics under limited anchor access.** We evaluate SFT, Replay, CPL, and CPLUS after every epoch over 25 epochs on Qwen3-0.6B and CommonsenseQA with $A = 8$ anchors and `lr=1e-6`. Left and center show Retention and Acquisition over training; right traces the corresponding Acquisition–Retention path. Both Retention panels use the same vertical scale, labeled on the left. Per-epoch uncertainty is not shown. Filled markers identify the first epoch reaching 100% Acquisition.

correction are measured on the complete held-out base-correct and base-wrong CommonsenseQA subsets, respectively, excluding the eight anchors.

The endpoint comparison shows the tradeoff: base-correct anchors more strongly preserve established predictions, while base-wrong anchors improve correction and overall accuracy.

Table 4: Effect of anchor composition when adapting Qwen3-0.6B to MedQA with $A = 8$ ground-truth-labeled CommonsenseQA anchors. All rows obtain 100% MedQA Acquisition and 100% accuracy on the eight supplied anchors. Results use one seed.

Anchor composition		CommonsenseQA (%)		
Base-correct	Base-wrong	Retention	Correction	Overall
8	0	83.49	13.42	40.15
6	2	81.34	21.24	44.16
4	4	80.62	23.16	45.07
2	6	80.38	20.80	43.52
0	8	80.86	23.45	45.35

E.5 Fixed Aggressive-Update Stress Test

We test optimization robustness under a fixed aggressive-update schedule. Forgetting remains substantial at the default `lr=1e-6`, despite the benefits of small finetuning learning rates (Marek et al., 2026); the larger-learning-rate schedule below produces more severe forgetting (Figure 3 and Table 5).

For this stress test, we apply one pre-specified aggressive schedule to every method: AdamW with learning rate `lr=1e-5` for exactly 10 epochs. We disable early stopping, epoch selection, and Acquisition-based run filtering, and evaluate the final epoch. The model, CommonsenseQA split, $A = 8$ access budget, optimizer, and seed are shared across methods.

Table 5 shows that CPLUS matches SFT's 98.97% Acquisition while retaining 17.22 percentage points more of the base model's correct predictions. Replay attains complete Acquisition but retains less, and CPL reaches only 42.63% Acquisition under this deliberately aggressive schedule. The

result therefore complements the Acquisition-controlled comparisons: CPLUS continues to preserve more established behavior when every method receives the same fixed optimization pressure.

Table 5: Fixed aggressive-update stress test on Qwen3-0.6B and CommonsenseQA with $A = 8$. Every method uses AdamW at `lr=1e-5` for exactly 10 epochs, with no early stopping, epoch selection, or Acquisition filtering. Results are from seed 0.

Method	Acquisition (%)	Retention (%)	Overall accuracy (%)
SFT	98.97	21.77	69.53
Replay	**100.00**	12.68	66.70
CPL	42.63	9.57	30.02
CPLUS	98.97	**39.00**	**76.09**

E.6 Matched Update Controls

We test whether CPLUS retains more behavior merely because its mask reduces the magnitude or density of the applied update. Norm-matched SFT preserves the full SFT direction but rescales its update norm to the norm produced by CPLUS at the same model state. The randomized-mask control permutes CPLUS's effective mask coefficients within each parameter tensor at every step. This preserves the exact per-tensor coefficient multiset and nonzero fraction, while breaking the association between masked coordinates and the preservation gradient.

This diagnostic uses the canonical Qwen3-0.6B CommonsenseQA setting with $A = 8$, seed 0, the same sealed $N = 512$ self-probe bank, and the same 25-epoch schedule for all rows. Table 6 shows that neither matched update norm nor matched mask coefficients reproduce CPLUS's Retention at the same Acquisition.

Table 6: Matched update controls. Every row achieves 100% Acquisition. The randomized mask preserves CPLUS's exact within-tensor coefficient multiset, while norm-matched SFT matches the norm of the update applied by CPLUS.

Method	Matched property	Retention (%)	Acquisition (%)
Norm-matched SFT	Applied update norm	45.45	100
Randomized CPLUS mask	Per-tensor mask coefficients	47.13	100
CPLUS	Probe-informed coordinate placement	**71.05**	100

E.7 Where the Preservation Mask Acts During Training

We examine how the preservation signals constrain proposed updates over training in the canonical limited-anchor setting. CPL and CPLUS use separate, matched Qwen3-0.6B CommonsenseQA runs with $A = 8$, seed 0, AdamW at `lr=1e-6`, and 25 epochs; CPLUS uses its sealed $N = 512$ self-probe bank. At seven pre-update states, a read-only audit previews the AdamW proposal on the same deterministic 32-example adaptation subset and computes the instantaneous gradient-sign mask for each method. For CPLUS, the diagnostic uses the blended probe–anchor gradient before the mask's EMA smoothing. The audit does not change training updates.

Figure 13 shows that CPLUS's instantaneous mask admits fewer coordinates and less candidate-update energy in the early states. From nine completed epochs onward, both methods admit approximately half of the coordinates and half of their respective proposals' squared update energy. These similar late-stage aggregate fractions do not account for the final gap: both methods reach 100% Acquisition, while CPLUS retains 71.05% of initially correct behavior and CPL retains 50.48%.

The placement control in Table 6 isolates the missing spatial information: permuting CPLUS's mask coefficients within each parameter tensor preserves their exact multiset at every step but lowers

Retention to 47.13% at the same 100% Acquisition. Together, the temporal audit and the placement control support the interpretation that *which coordinates are attenuated matters*, not only how many coordinates or how much update energy the mask admits.

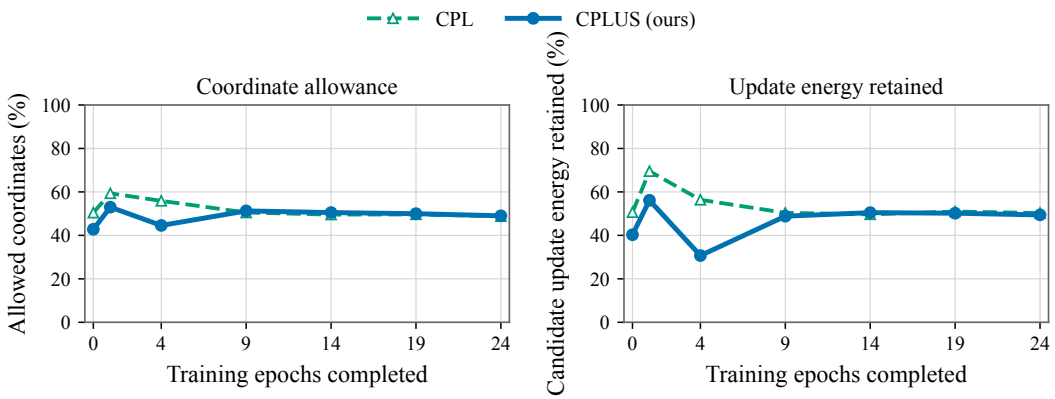

Figure 13: **Instantaneous mask geometry under limited anchor access.** CPL and CPLUS are audited at seven pre-update states in separate, matched Qwen3-0.6B CommonsenseQA runs with $A = 8$. The horizontal axis counts training epochs already completed. **Left:** fraction of coordinates admitted by the diagnostic gradient-sign mask. **Right:** fraction of each method's candidate AdamW update squared norm admitted by that mask. The CPLUS diagnostic uses its blended probe–anchor gradient before EMA smoothing. The audit is read-only; results use seed 0.

E.8 Mask Behavior

We ask whether CPLUS's advantage comes from closer agreement with a full-data update mask. We replay the same proposed updates across anchor budgets and compare each method with its own full-data reference.

Figure 14 summarizes agreement with these references, while Figure 15 shows the corresponding coordinate-level masks. CPL more closely reconstructs its own full-data mask at every budget. At $A = 128$, update-weighted agreement reaches 0.95 for CPL and 0.89 for CPLUS. Nevertheless, CPLUS achieves higher Retention under limited access. Its improvement therefore does not arise from greater agreement with its own full-data reference. Instead, blending the self-probe and past-example gradients produces a distinct, temporally smoothed update mask. Because this audit covers one model–benchmark pair, we treat it as a diagnostic rather than a universal mechanistic explanation.

E.9 Sequential Multi-Task Learning

We update the model sequentially on CommonsenseQA, ARC-Challenge, MMLU, and MedQA. All methods begin from the same post-CommonsenseQA checkpoint and subsequently continue from their own preceding checkpoint. At each stage, evaluation covers every task encountered so far. CPLUS maintains a separate past-example set and self-probe bank for each preceding task and averages the resulting task-level preservation gradients before masking the current-task update.

Table 7 reports the stage averages, while Table 8 breaks down accuracy by task after the final stage.

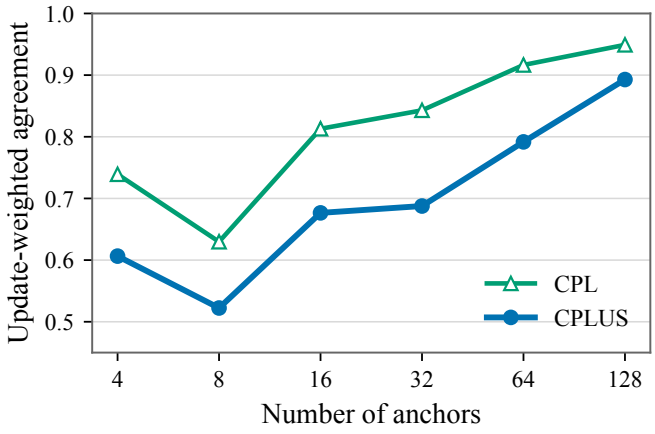

Figure 14: **CPLUS need not recover the full-data mask.** Each method is compared with its own full-data reference. CPL is closer at every budget despite lower Retention (Correct → Correct), so mask similarity alone cannot explain CPLUS's advantage. Coordinate-level masks appear in Figure 15.

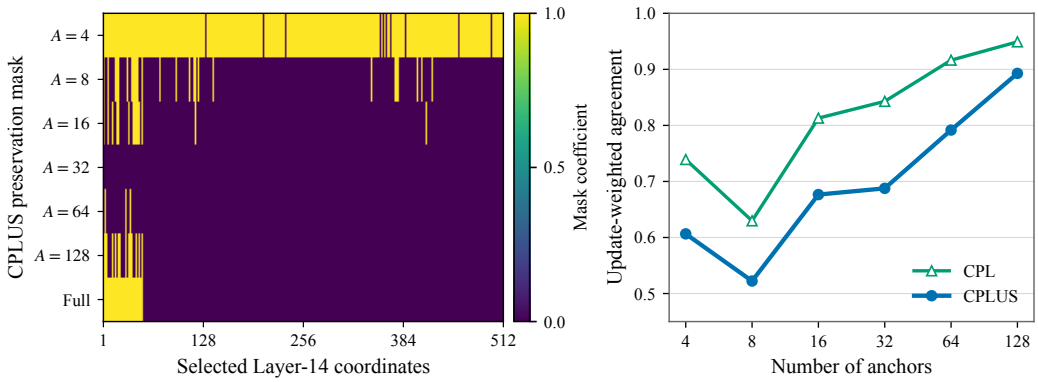

Figure 15: **Coordinate-level view of the mask audit.** We replay the same 678 proposed optimizer updates under controlled probe-side inputs. **Left:** each row shows the update-weighted mean CPLUS mask at one anchor budget; "Full" uses all 418 past examples. Columns are 512 pre-registered coordinates from a middle-layer MLP down-projection, ordered identically in every row. Purple indicates a suppressed update coefficient (0) and yellow a retained coefficient (1). **Right:** each method is compared with its own full-data reference mask—CPL with full-data CPL and CPLUS with full-data CPLUS. Agreement is update-energy weighted, and 1 denotes identical masks.

Table 7: **Average Task Accuracy during sequential multi-task learning.** Each entry is the unweighted mean of absolute accuracy over all tasks encountered through the indicated stage, averaged across three seeds. Task-level standard errors are reported in Table 8 for the final stage and visualized throughout Figure 6.

Method	After ARC-C	After MMLU	After MedQA
SFT	66.81	72.20	61.21
Replay	72.04	77.29	69.73
CPL	75.18	70.77	69.65
CPLUS	**83.94**	**80.86**	**78.87**

Table 8: **Accuracy after the final sequential-learning stage.** Following adaptation to MedQA, CPLUS achieves the highest accuracy on every task encountered in the sequence. The final Average Task Accuracy is the unweighted mean of the four absolute task accuracies. Values are means ± standard errors across three seeds; the Average column is computed from the task means.

Method	CSQA	ARC-C	MMLU	MedQA	Average
SFT	36.04 ± 0.51	53.78 ± 0.49	74.82 ± 0.44	80.19 ± 0.43	61.21
Replay	55.29 ± 0.49	63.86 ± 0.45	79.59 ± 0.41	80.19 ± 0.42	69.73
CPL	59.85 ± 0.47	62.60 ± 0.45	74.38 ± 0.42	81.77 ± 0.40	69.65
CPLUS	$\mathbf{69.34 \pm 0.46}$	$\mathbf{78.27 \pm 0.44}$	$\mathbf{83.43 \pm 0.44}$	$\mathbf{84.45 \pm 0.42}$	**78.87**

E.10 Base-Normalized Capability Scores

The main text reports per-benchmark changes after adaptation in the right panel of Figure 1. For completeness, Figure 16 shows the same evaluation normalized by each benchmark's base-model score. CPLUS remains closest to the base capability profile across the five benchmarks not used for adaptation.

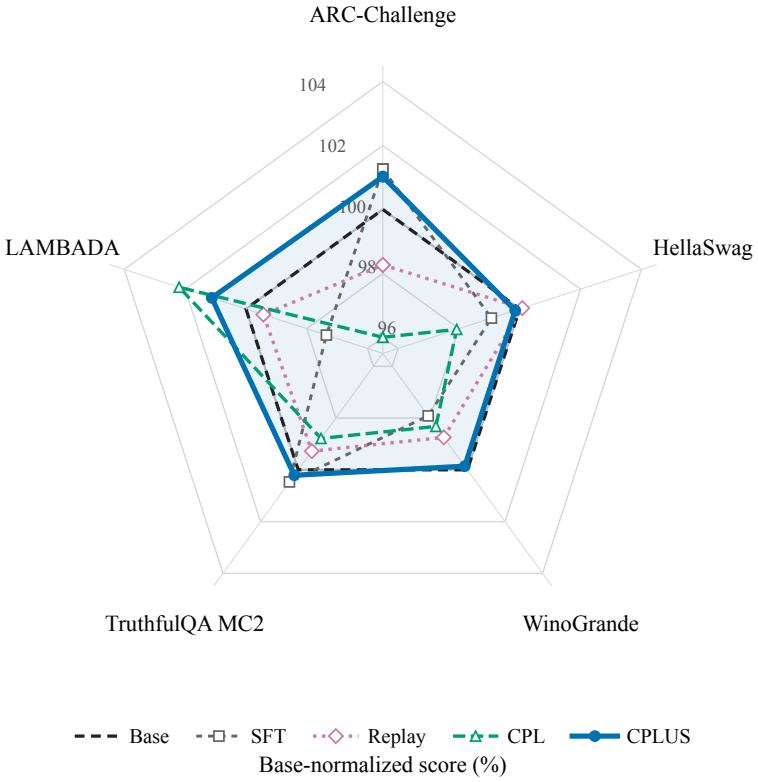

Figure 16: **Base-normalized capability scores after adaptation.** Each axis reports the post-adaptation score as a percentage of the corresponding base-model score. The dashed polygon marks the base-model level (100%); values above 100 indicate improvement. Raw score changes are shown in Figure 1.

E.11 Layer-Wise Mask Audit

We extend the mask-behavior audit in Section E.8 across transformer layers. The diagnostic uses Qwen3-0.6B on CommonsenseQA, the sealed $N = 512$ self-probe bank, and a common sequence of 678 proposed updates. Model weights are held fixed throughout the audit. For CPL, the full-data reference uses all 418 past examples. For CPLUS, the method-matched reference blends the same full past dataset with the controlled self-probe inputs used throughout the audit.

For candidate mask trajectory m, reference trajectory $m^\star$, and proposed updates δ, we report update-weighted agreement,

$$\text{Agreement}(m, m^\star) = 1 - \frac{\sum_{t,i} \delta_{t,i}^2 |m_{t,i} - m_{t,i}^\star|}{\sum_{t,i} \delta_{t,i}^2}.$$

This emphasizes coordinates that materially contribute to the proposed updates. The coordinate-level mask shown in Figure 15 is the corresponding update-weighted temporal mean. Its 512 coordinates are selected once using aggregate update energy and then held fixed across all budgets and references.

Figure 17 shows the method-matched comparison across all transformer layers. CPL remains closer to its full-data reference than CPLUS across budgets and depth. Both methods become more stable as access increases overall, but both exhibit a drop at $A = 8$; we therefore do not claim monotonic convergence at the smallest budgets.

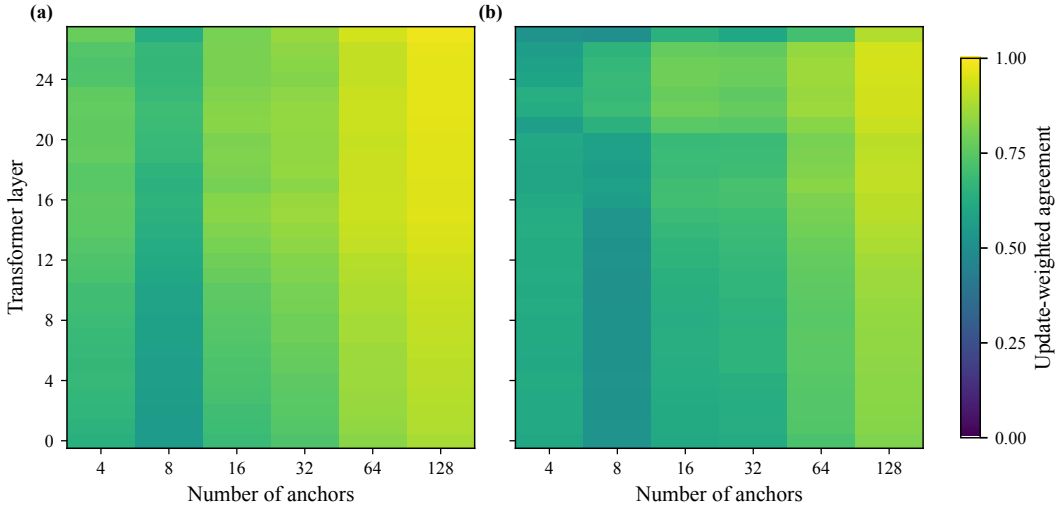

Figure 17: **Method-matched mask agreement across transformer depth.** Each cell reports update-weighted agreement for one layer and anchor budget. **(a)** CPL with A anchors compared with CPL using all 418 past examples. **(b)** CPLUS with A anchors compared with CPLUS using all 418 past examples under the same controlled probe inputs. The shared color scale ranges from zero to one.

For completeness, Figure 18 directly compares CPLUS with the all-real CPL reference. This is a cross-method diagnostic, not the target that CPLUS is intended to reconstruct. The low agreement shows that the two methods suppress different updates. Whether that difference preserves broader function is evaluated behaviorally rather than inferred from mask similarity.

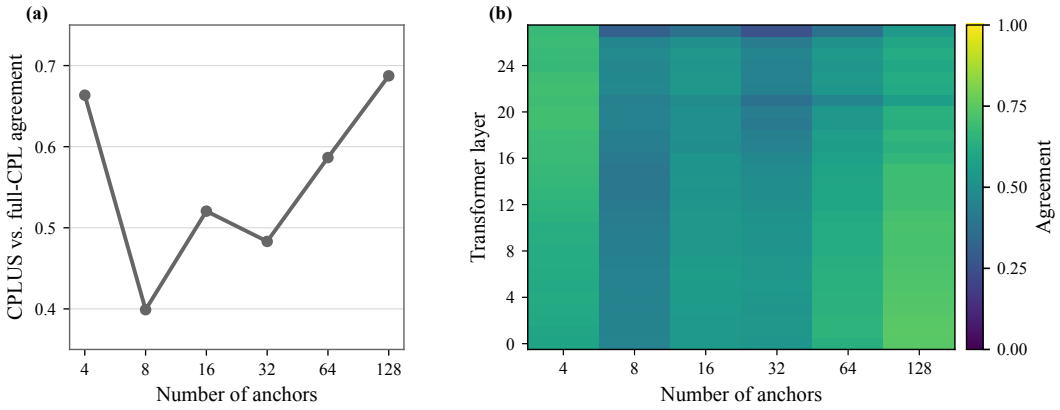

Figure 18: **Cross-reference comparison with full-data CPL.** CPLUS masks are compared directly with the mask constructed by CPL from all 418 past examples. **(a)** Agreement for the pre-registered Layer-14 MLP down-projection. **(b)** Agreement across all transformer layers. This deliberately mismatched reference measures how differently CPLUS suppresses updates; it is not an oracle target for CPLUS.

E.12 Evaluation and Compute Details

In the primary grid, Acquisition covers the initially incorrect adaptation examples; Retention covers all initially correct examples, including anchors. The anchor-composition diagnostic excludes its anchors (Section E.4); external-task averages exclude adaptation and anchor benchmarks.

Figure 7 compares the three Qwen3 scales on CommonsenseQA at $A = 8$. CPLUS uses $N = 32$ rather than the default 512 to reduce compute. Frozen distillation matches frozen-teacher output distributions during adaptation (Hinton et al., 2015). OGD-GTL projects new-data gradients orthogonally to a basis of anchor ground-truth-logit gradients (Farajtabar et al., 2020).

Training FLOPs are computed using the leading dense-transformer approximations $2PL$ per forward pass and $6PL$ per forward-plus-backward pass, where P is model size and L is padded sequence length. Repeated preservation passes are included; evaluation and one-time preparation, including probe generation and SSR refinement, are excluded. FLOPs describe arithmetic cost, not elapsed time.

F Limitations

Self-probes record the frozen model's behavior, not independently verified truth. If that behavior is biased or poorly covered by the probes, the resulting preservation signal can limit useful adaptation updates. Our scale results characterize the tested models, and the sequential results concern the tested task order. These findings do not establish preservation of every capability or robustness to arbitrary task sequences.

G Extended Related Work

Gradient- and parameter-level preservation. Continual learning methods often mitigate interference by constraining updates according to information retained from earlier tasks. Elastic Weight Consolidation penalizes changes to parameters estimated to be important for prior tasks (Kirkpatrick et al., 2017), while GEM and A-GEM use gradients from episodic memory to prevent new-task updates from increasing previous-task loss (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019). OGD instead projects updates away from directions that alter previously learned outputs (Farajtabar et al., 2020). More recently, CPL decomposes gradient interaction coordinate-wise and suppresses updates predicted to conflict with previously correct behavior (Yang et al., 2026). CPL also evaluates partial access to these samples. With limited data, its preservation gradient can miss behavior outside

the retained examples. CPLUS complements the available samples with self-probes from the frozen base model and blends the two signals before masking the adaptation update.

Model-generated supervision. Recent work shows that a language model's own predictive distribution can support continued adaptation. Self-generated replay uses samples from the model's prior distribution as replay data and can substantially reduce forgetting when sufficient capacity remains (Marek et al., 2026). Self-Distillation Fine-Tuning conditions the model on demonstrations and uses its on-policy predictions as supervision for acquiring new skills (Shenfeld et al., 2026). RL's Razor attributes the lower forgetting of on-policy reinforcement learning to its preference for solutions that remain closer to the original policy (Shenfeld et al., 2025). CPLUS shares the principle that the base model contains information worth preserving, but self-probes are neither replay samples nor targets for the new behavior. They are used only to construct the gradient that guides the preservation mask.

Capacity, plasticity, and model scale. Forgetting depends on the model's pretraining state, remaining capacity, and ability to remain plastic. Knowledge entropy has been observed to decline during language-model pretraining alongside weaker knowledge acquisition and retention (Kim et al., 2025), while forgetting can persist when a model approaches capacity saturation (Marek et al., 2026). Loss of plasticity is distinct from forgetting: a model may preserve earlier behavior yet become progressively less able to acquire new information (Dohare et al., 2024). When memory is abundant, the principal difficulty can consequently shift from retention toward acquisition (Cho et al., 2025). These findings motivate measuring both quantities and separating native robustness, absolute CPLUS performance, and the incremental gain over matched baselines.

Evaluation and efficiency. Tuning and evaluating a method within the same continual learning scenario can overestimate its ability to generalize (Cha & Cho, 2025). We adopt the corresponding two-phase principle by selecting hyperparameters on one development setting and applying the self-probe and masking settings unchanged across the reported model sizes, benchmarks, past-data budgets, and random seeds. Under explicit compute constraints, simple memory-sampling baselines can match or outperform more elaborate methods (Prabhu et al., 2023), and some continual learning systems can consume more computation than retraining from scratch (Harun et al., 2023). These findings motivate our comparison with replay and access-matched CPL, and our use of a simple global gradient blend.