ONLINE REPRESENTATION PRESERVATION FOR CONTINUAL INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual instruction tuning requires language models to acquire new capabilities while maintaining performance on previously learned tasks. Replay trains the model on stored or selected responses, whose prefixes may differ from those generated by the current student. We propose ReTAP (Representation-based Task Anchoring and Preservation), which matches the student's hidden states to those of frozen, task-specific reference models on freshly sampled student responses. Before each training stage, each reference is selected from completed student checkpoints based on validation performance on the corresponding task. This representation loss is combined with supervised learning on the current task, without requiring independently trained expert models. On the eight-task TRACE benchmark, ReTAP achieves the highest final average performance and backward transfer among the evaluated continual-learning methods across all three model backbones. Its average scores fall only 0.91, 1.47, and 0.32 points below independently trained single-task benchmarks on Qwen2.5-7B-Instruct, Qwen3-1.7B, and Qwen3-4B, respectively, compared with deficits of 2.85, 4.23, and 2.71 points for the best competing method on each backbone. These results demonstrate effective historical capability preservation within a single continually updated model.

1 INTRODUCTION

Large language models require continued adaptation to evolving tasks while retaining earlier capabilities. Continual instruction tuning addresses this challenge by training a single model on sequential instruction-following tasks (Wang et al., 2023). Among approaches to mitigating forgetting, replay interleaves historical examples with current-task data (Chaudhry et al., 2019). Conventional replay uses stored answers, while On-Policy Replay selects model-generated responses (Chen et al., 2026b); both supervise predictions conditioned on the replayed answer prefixes.

As new-task training changes the student's behavior, its generated prefixes may diverge from those in replayed answers (Figure 1, bottom left). Replay does not directly supervise continuations from these alternative prefixes, potentially weakening preservation. Moreover, the latest checkpoint may already have lost earlier capabilities. This motivates two design choices: which response prefixes to supervise and which reference model to use for each previous task.

We propose ReTAP (Representation-based Task Anchoring and Preservation), illustrated in Figure 1.Before each stage, ReTAP selects the completed student checkpoint with the highest held-out validation score for each previous task and keeps the selected reference frozen throughout the stage. Each time a historical prompt is revisited, the student generates a fresh response. The student and the selected reference model process the same generated response. ReTAP minimizes the mean squared error between their hidden states at the response positions.Training combines this representation loss with cross-entropy on gold responses for the current task and updates only the student.

On the eight-task sequence of TRACE, a comprehensive benchmark for continual learning in large language models (Wang et al., 2023), ReTAP achieves the highest average performance and backward transfer among the compared continual-learning methods across three backbones. Its average scores fall only 0.91, 1.47, and 0.32 points below independently trained single-task benchmarks on Qwen2.5-7B-Instruct, Qwen3-1.7B, and Qwen3-4B, respectively. Compared with the strongest evaluated continual-learning baseline on each backbone, ReTAP improves average performance by 1.94, 2.76, and 2.39 points, respectively.

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.

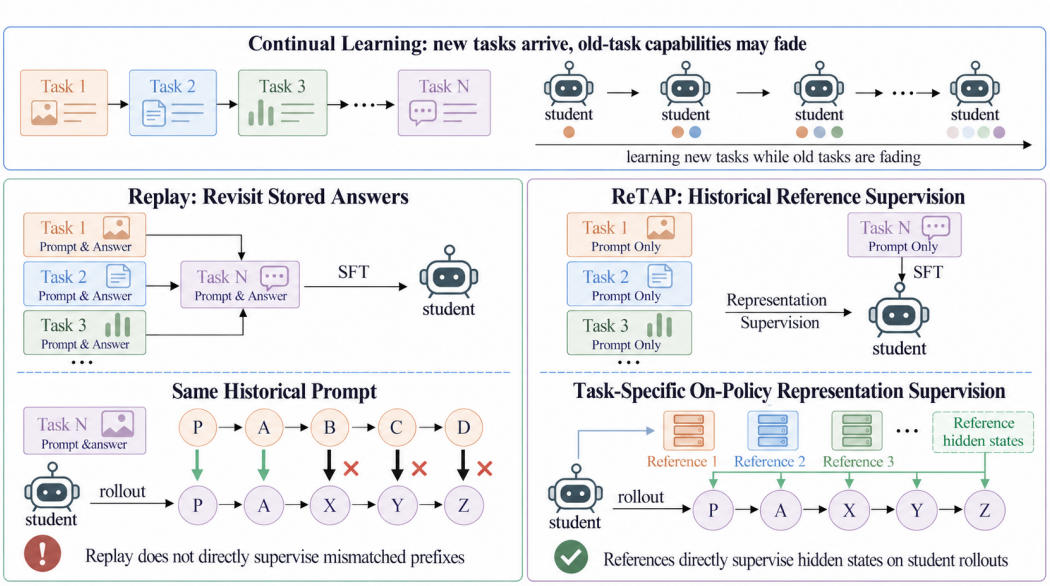

Figure 1: **Continual instruction tuning with ReTAP.** Tasks arrive sequentially, and the student learns each current task from gold answers. To preserve earlier tasks, ReTAP generates responses to historical prompts and matches the student's hidden states to those of frozen, task-specific references on the same sequences. The references come from completed student checkpoints; conventional replay instead uses stored answers.

2 Related Work

Continual instruction tuning requires learning new tasks while retaining historical capabilities (Wang et al., 2023). Rehearsal mitigates forgetting through historical examples (Chaudhry et al., 2019), while LAMOL generates pseudo-examples for earlier tasks (Sun et al., 2019). However, answer-based replay supervises fixed response prefixes, which may differ from those visited by the evolving student.

Recent studies highlight the role of on-policy data in reducing forgetting (Chen et al., 2026a). On-Policy Replay (OPR) generates responses to historical prompts at stage boundaries, filters them using task rewards or self-confidence, and replays the selected answers with cross-entropy (Chen et al., 2026b). Although these responses reflect student behavior, they remain fixed within the next stage and serve as training labels, coupling preservation to the selected answer content.

Teacher feedback on student-generated responses provides an alternative to training on fixed replay answers. GKD provides teacher feedback on student-generated sequences (Agarwal et al., 2024), while SDFT uses demonstration-conditioned self-distillation to mitigate forgetting (Shenfeld et al., 2026). Function-vector regularization preserves internal task representations during continual instruction tuning (Jiang et al., 2025).OPRD is closely related to our approach: it aligns teacher and student hidden states on student-generated rollouts (Yang et al., 2026). Building on these ideas, historical task preservation requires coordinating task-specific references, evolving student trajectories, and new-task supervision.

ReTAP applies on-policy representation matching to continual instruction tuning. It selects a reference model for each previous task from completed student checkpoints and combines representation matching on fresh student responses with supervised learning on the current task.

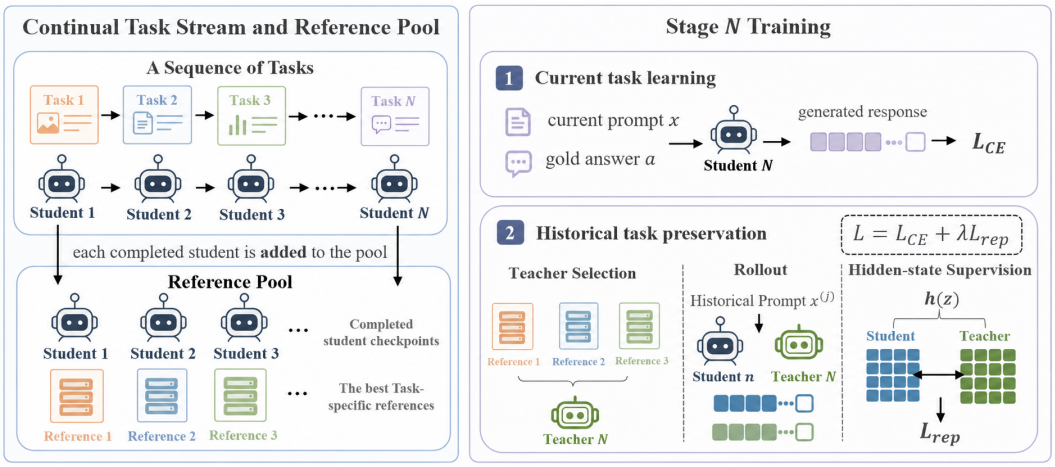

Figure 2: **ReTAP training.** Current-task answers supervise the student, while historical prompts elicit fresh student responses. Frozen task references from completed checkpoints provide hidden-state targets on those responses. Only the student is updated.

3 ReTAP: Task-Referenced Representation Preservation

3.1 Continual learning with task references

Let $\mathcal{D}_1, \ldots, \mathcal{D}_T$ be an ordered sequence of instruction–answer datasets, and let θ_t denote the student after completing task t. ReTAP updates a single student model throughout the task sequence by combining supervised learning on the current task with representation matching against frozen, task-specific reference models (Figure 2). The main configuration uses LoRA (Hu et al., 2021): the adapter learned at stage $t-1$ initializes stage t, while the base model remains frozen.

At stage $t > 1$, the anchor set $\mathcal{A}_t$ contains M task–prompt pairs (j, x) with $j < t$. By default, $M = 50$ prompts are drawn without replacement from previous training pools, allocated evenly across tasks with remainders assigned in task order. Anchors are reused within a stage and redrawn for the next; M counts prompts before rollout expansion.

ReTAP selects its reference models from checkpoints produced earlier in the same continual-learning run. For each previous task j, the candidates are completed student checkpoints $\{\theta_j, \ldots, \theta_{t-1}\}$. Before stage t, ReTAP selects the candidate with the highest score on task j evaluated on a dedicated validation split, choosing the earliest eligible checkpoint on ties, and freezes it as $\phi_{j,t}$ for the entire stage. The selected reference need not be the most recent checkpoint, because an earlier checkpoint may achieve a higher validation score on task j. This choice is made separately for each historical task, without future-task checkpoints or independently trained experts. The same policy applies to the reported ReTAP ablations.

3.2 Preservation on student-generated trajectories

For a historical prompt x from task j, sample K responses from the current student,

$$y^{(k)} \sim \pi_{\mathrm{sg}(\theta)}(\cdot \mid x), \qquad k = 1, \ldots, K, \tag{1}$$

where sg denotes stop-gradient. By default, $K = 8$ responses are sampled at temperature 1. Generation stops when EOS is produced or the response-length limit is reached. Responses are capped at one token for C-STANCE and FOMC and at 512 tokens for the other tasks, subject to the context-length limit. Student and reference process the same student-generated sequences; the reference generates no answers, and historical gold answers provide no preservation targets. Let $s_u = (x, y_{\le u})$ denote a supervised prefix and $h_\theta^\ell(s_u) \in \mathbb{R}^d$ its layer-ℓ hidden state. The preservation loss is the

mean squared discrepancy between student and reference states:

$$r(\theta, \phi; x, y) = \frac{\sum_{u \in \mathcal{P}(y)} \sum_{\ell \in \mathcal{L}} \|h_\theta^\ell(s_u) - \mathrm{sg}[h_\phi^\ell(s_u)]\|_2^2}{|\mathcal{P}(y)|\,|\mathcal{L}|\,d}, \tag{2}$$

$$\bar{r}_j(x) = \frac{1}{K} \sum_{k=1}^{K} r(\theta, \phi_{j,t}; x, y^{(k)}). \tag{3}$$

Here $\mathcal{P}(y)$ indexes response prefixes and $\mathcal{L}$ the supervised layers. By default, all non-embedding states, including the final normalized state, are supervised; padding is excluded. We average the squared errors over response positions, supervised layers, and hidden dimensions, so that longer responses are not assigned greater weight solely because of their length. We then average over the K responses sampled for each historical prompt. Responses are regenerated on each visit, allowing the prefixes used for preservation to track the evolving student. The generated tokens specify where representation matching is evaluated; supervision comes from the frozen task reference's hidden states at those prefixes.

3.3 A PROMPT-LEVEL TRAINING OBJECTIVE

For a current example (x, a), define its mean response-token loss $c_\theta(x, a) = -|\mathcal{P}(a)|^{-1} \sum_{u \in \mathcal{P}(a)} \log \pi_\theta(a_u \mid x, a_{<u})$. Let B_c and B_h be the current and historical items in a local microbatch, and $m = |B_c| + |B_h|$. ReTAP uses

$$\mathcal{L}(B) = \frac{1}{m} \left[\sum_{(x,a) \in B_c} c_\theta(x, a) + \lambda \sum_{(j,x) \in B_h} \bar{r}_j(x) \right]. \tag{4}$$

In this *prompt-mean* objective, m counts each historical prompt once before its K rollouts. Increasing K therefore adds sampled responses without increasing the aggregation weight assigned to the historical prompt. The coefficient λ controls the relative weight of the historical representation loss. We use $\lambda = 10$ for all backbones and examine the effects of both K and λ in Section 4.

The first stage uses ordinary supervised fine-tuning because no historical task is available. Subsequent stages optimize Equation (4), differentiating through student representations while holding sampled sequences and reference states fixed. Thus, the preservation gradient changes the student on its own generated prefixes, while the current-task loss continues to train on gold responses.

3.4 TRAINING PROCEDURE AND DEPLOYMENT

Within each microbatch, we compute cross-entropy loss on current-task examples and sample K student responses for each historical prompt. Frozen references process those responses without gradients; their detached hidden states provide targets for the student update. Student and reference forward passes use BF16. Reference hidden states are temporarily stored in FP16, while the MSE subtraction and reduction are performed in FP32. Appendix A specifies the decoding limits, optimizer, batch layout, and stage schedule.

With LoRA, completed student adapters are kept as task-reference candidates over a shared frozen base. At inference time, the shared base model uses only the final student adapter; no reference model is selected or evaluated.

Behavioral interpretation. In the LoRA configuration, student and references share a frozen output head W. For their next-token distributions p_θ, p_ϕ at a supervised prefix s, the final-state discrepancy gives $\mathrm{TV}(p_\theta, p_\phi) \le \|W\|_2 \|h_\theta(s) - h_\phi(s)\|_2 / (2\sqrt{2})$. The default all-layer loss includes this final normalized state, while intermediate-layer targets constrain the representations from which it is computed. This bound concerns next-token distributions only at supervised prefixes; it is not a guarantee of sequence-level task performance, and the shared-head argument does not directly apply when the output head changes under full fine-tuning. Appendix B gives the formal result and proof; Section 4.4 evaluates the supervised-layer choice.

4 EXPERIMENTS

4.1 SETUP

We evaluate final task performance, historical retention, and the design choices governing representation preservation. Experiments use the eight-task TRACE sequence (Wang et al., 2023): C-STANCE, FOMC, MeetingBank, Py150, ScienceQA, NumGLUE-cm, NumGLUE-ds, and 20Minuten. The sequence combines classification, summarization, code completion, numerical reasoning, and multilingual generation. We evaluate Qwen2.5-7B-Instruct and Qwen3-1.7B/4B (Qwen et al., 2025; Yang et al., 2025).

Training and baselines. In the main comparison reported in Table 1, all continual-learning methods use the same preprocessed source datasets, task order for each seed, training schedules, evaluation procedures, and a LoRA rank of 256. For replay-based methods (Replay, OPR-RU, OPR-SC, and ReTAP), the default replay ratio is $\rho = 0.01$, corresponding to 50 historical items incorporated into training at each stage. The sampling and response-selection procedures are described in Appendix A.2. Sequential SFT trains exclusively on current-task data, while our adaptation of LwF (Li & Hoiem, 2017) distills previous-model predictions on current-task inputs. OPR-RU and OPR-SC (Chen et al., 2026b) select replay answers based on task reward and self-confidence, respectively. ReTAP sets $\lambda = 10$, $K = 8$, and applies all-layer representation MSE across all backbones.

Ablation settings. The K sweep aggregates three paired seed-and-order settings; other ablations use seed 1234 and the original order (Appendix A).

Metrics. Each completed stage is evaluated using greedy decoding on all tasks encountered so far. Let $R_{t,j}$ denote the score on task j after stage t. We report the final average performance (Avg) and backward transfer (BWT) (Lopez-Paz & Ranzato, 2022):

$$\mathrm{Avg} = \frac{1}{T}\sum_{j=1}^{T} R_{T,j}, \quad \mathrm{BWT} = \frac{1}{T-1}\sum_{j=1}^{T-1}(R_{T,j} - R_{j,j}). \tag{5}$$

Task scores use a 0–100 scale, and both Avg and BWT are higher-is-better. Avg reflects overall final capability, while BWT measures performance change on historical tasks relative to their initial learning (where negative values indicate forgetting). We report the mean and sample standard deviation across three runs with seeds 1234, 1235, and 1236, each using a different prescribed task order. All methods, including ReTAP, use the same seed–order pairs. Metrics are computed separately for each run before aggregation. See Appendix A for detailed task sequences and scoring criteria.

Comparison with single-task learning. For each backbone, we include independently trained single-task models as evaluation benchmarks; they are never used as ReTAP training teachers. Let $\mathrm{Avg}_{\mathrm{ST}}$ be the mean of their eight task scores. We report the signed difference

$$\Delta_{\mathrm{ST}} = \mathrm{Avg} - \mathrm{Avg}_{\mathrm{ST}}. \tag{6}$$

A negative value indicates a deficit relative to single-task learning, zero indicates a match, and a positive value indicates a gain. This compares final performance with a common evaluation benchmark; BWT instead measures change from each continual learner's own initial task scores. The single-task scores provide an empirical target for continual learning, not a guaranteed upper bound.

4.2 FINAL PERFORMANCE AND CROSS-MODEL EVALUATION

ReTAP improved mean performance and retention over compared baselines on Qwen2.5-7B-Instruct (Table 1). Across three runs, it reached 66.70 ± 0.32 Avg and -0.40 ± 0.34 BWT. Corresponding gains over Replay were 2.68 and 2.61 points. It also exceeded OPR-RU by 1.94 Avg and 2.34 BWT. These results support the overall ReTAP configuration. An additional comparison shows ReTAP retains an advantage over OPR-RU when both use prompt-mean aggregation (Appendix C.2).

Using the same preservation hyperparameters, ReTAP achieves an Avg of 64.75 ± 0.57 and a BWT of -0.43 ± 0.50 on Qwen3-4B. Its mean gains over Replay were 3.00 Avg and 2.83 BWT, with both metrics also exceeding the other baselines. Appendix D reports the original-order ReTAP results.

Table 1: Final TRACE scores after eight stages. For continual-learning methods, task scores, Avg, and BWT report means with sample SD beneath over three paired seed-and-order settings. Each *Single-task* row gives independently trained per-task evaluation scores; these models do not provide ReTAP training targets, and BWT is inapplicable. $\Delta_{\mathrm{ST}} = \mathrm{Avg} - \mathrm{Avg}_{\mathrm{ST}}$ is computed from the displayed averages (points). All scores are higher-is-better. Bold marks the best continual-learning mean per column.

Method	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.	Avg.↑	BWT↑	Δ_{ST} ↑
Qwen2.5-7B-Instruct											
Single-task	59.90	69.76	65.09	68.66	89.25	76.54	71.08	40.56	67.61	–	0.00
Sequential SFT	55.33 ±2.61	68.08 ±1.34	39.66 ±15.22	62.61 ±3.35	88.40 ±0.80	69.14 ±3.27	69.13 ±0.94	41.05 ±0.10	61.67 ±2.71	−5.83 ±3.20	−5.94
Replay	56.18 ±2.89	68.08 ±3.14	56.69 ±0.90	65.88 ±0.62	88.33 ±0.68	68.72 ±1.89	67.49 ±1.08	40.79 ±0.30	64.02 ±1.03	−3.01 ±1.04	−3.59
OPR-RU	56.78 ±1.54	68.62 ±1.71	57.12 ±0.65	65.85 ±0.77	**88.78** ±0.29	68.72 ±4.34	**71.08** ±1.34	**41.15** ±0.56	64.76 ±0.83	−2.74 ±0.66	−2.85
OPR-SC	54.63 ±1.98	66.40 ±2.16	52.23 ±1.06	64.12 ±3.98	88.40 ±0.26	67.08 ±2.85	70.36 ±0.89	40.01 ±1.99	62.90 ±0.62	−4.85 ±0.56	−4.71
LwF	**60.38** ±1.48	66.73 ±0.81	51.75 ±7.30	51.48 ±10.01	87.75 ±0.39	61.73 ±3.70	44.41 ±9.86	39.18 ±0.18	57.93 ±3.89	−1.06 ±1.92	−9.68
ReTAP	59.18 ±0.25	**71.51** ±0.62	**64.88** ±0.15	**67.57** ±0.41	87.97 ±0.89	**71.19** ±0.71	70.26 ±0.47	41.07 ±0.09	**66.70** ±0.32	**−0.40** ±0.34	**−0.91**
Qwen3-1.7B											
Single-task	53.85	68.15	61.69	63.93	77.50	64.20	57.54	39.89	60.84	–	0.00
Sequential SFT	50.32 ±1.72	62.77 ±2.83	37.55 ±3.23	52.57 ±8.53	75.28 ±2.43	54.32 ±7.51	56.41 ±5.42	39.62 ±0.17	53.61 ±0.89	−8.51 ±1.47	−7.23
Replay	**55.20** ±2.00	62.84 ±0.84	48.85 ±1.66	59.24 ±0.43	77.92 ±1.25	54.32 ±2.47	54.87 ±4.27	39.62 ±0.26	56.61 ±0.47	−4.80 ±0.09	−4.23
OPR-RU	53.25 ±0.66	62.57 ±3.41	52.23 ±2.44	58.91 ±1.27	77.90 ±0.85	51.85 ±6.53	54.36 ±2.92	**39.64** ±0.56	56.34 ±1.02	−4.12 ±1.18	−4.50
OPR-SC	51.13 ±3.80	58.13 ±3.96	44.21 ±3.15	56.92 ±1.06	75.77 ±2.12	55.56 ±6.53	54.87 ±4.13	38.92 ±1.54	54.44 ±1.55	−7.21 ±1.59	−6.40
LwF	54.42 ±1.04	54.44 ±3.65	47.67 ±1.24	49.60 ±3.23	68.08 ±7.54	31.28 ±4.99	45.54 ±6.36	38.14 ±0.14	48.65 ±2.99	−3.79 ±1.14	−12.19
ReTAP	54.82 ±0.33	**65.32** ±1.32	**56.03** ±2.61	**61.53** ±0.63	**79.00** ±1.34	**62.55** ±3.77	**56.72** ±1.28	39.01 ±0.17	**59.37** ±0.69	**−0.64** ±0.75	**−1.47**
Qwen3-4B											
Single-task	59.75	69.96	64.34	65.95	87.25	64.20	68.00	41.13	65.07	–	0.00
Sequential SFT	55.62 ±0.47	64.05 ±3.41	38.93 ±3.82	63.81 ±0.42	86.15 ±1.80	64.20 ±3.27	65.85 ±1.87	39.97 ±1.03	59.82 ±1.18	−5.53 ±1.40	−5.25
Replay	56.35 ±1.03	65.12 ±2.53	52.39 ±2.33	64.14 ±0.52	**88.07** ±0.43	62.96 ±1.23	64.62 ±2.44	40.37 ±0.19	61.75 ±0.30	−3.26 ±0.64	−3.32
OPR-RU	56.05 ±0.22	65.99 ±1.37	57.71 ±2.28	64.27 ±0.12	87.27 ±1.15	62.14 ±3.97	65.23 ±0.31	40.22 ±0.48	62.36 ±0.62	−2.56 ±0.50	−2.71
OPR-SC	54.62 ±1.37	65.99 ±0.76	50.72 ±4.71	64.20 ±0.75	88.00 ±1.14	62.55 ±3.11	66.26 ±1.16	39.29 ±2.01	61.45 ±0.72	−3.98 ±0.76	−3.62
LwF	57.93 ±0.45	61.09 ±3.44	50.15 ±1.40	51.53 ±9.28	86.17 ±1.40	53.50 ±7.44	54.87 ±5.82	38.26 ±0.38	56.69 ±2.23	−3.41 ±1.46	−8.38
ReTAP	**58.38** ±1.40	**68.55** ±0.53	**62.75** ±0.78	**65.05** ±0.16	88.03 ±0.43	**67.90** ±1.23	**66.56** ±1.98	**40.80** ±0.07	**64.75** ±0.57	**−0.43** ±0.50	**−0.32**

Proximity to single-task performance. ReTAP achieves Δ_{ST} of -0.91, -1.47, and -0.32 on Qwen2.5-7B, Qwen3-1.7B, and Qwen3-4B, versus -3.59, -4.23, and -3.32 for Replay. It yields the smallest average deficit among continual-learning methods across all three backbones. On Qwen3-4B, its final average score is within 0.32 points of the single-task benchmark.

Task-level retention. MeetingBank illustrates the retention of a long-form generation task across subsequent training. On Qwen2.5-7B-Instruct, ReTAP finished at 64.88 ± 0.15, compared with means of 56.69 for Replay and 39.66 for Sequential SFT. Its final MeetingBank scores ranged from 64.71 to 64.97 across the three runs, preserving performance under the different task orders.

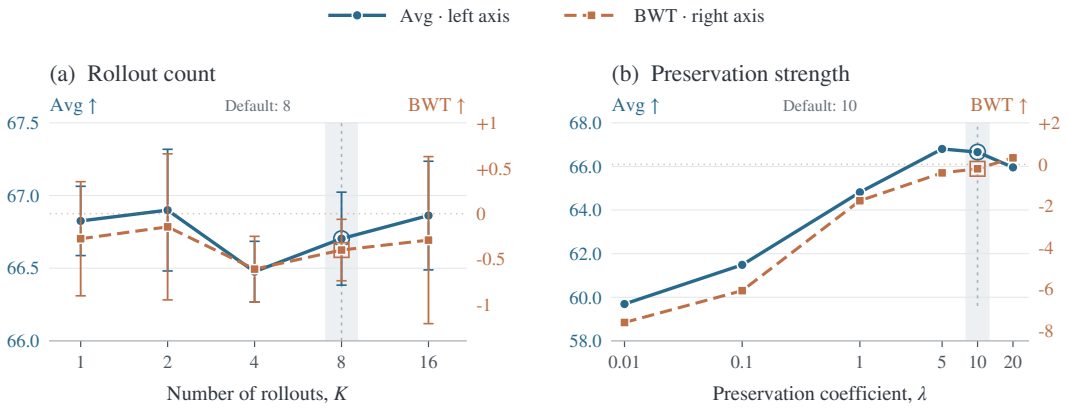

Figure 3: **Effect of ReTAP rollout count (K) and preservation strength (λ).** Evaluated on Qwen2.5-7B-Instruct with all-layer supervision. (a) Effect of K with $\lambda = 10$: mean ± sample SD across three seed–task-order pairs. (b) Impact of λ ($K = 8$): single run. Solid blue lines show average performance (left axis); dashed orange lines show backward transfer (BWT, right axis). Gray bands indicate default settings.

4.3 Rollout Count and Preservation Strength

Figure 3 examines rollout count K and preservation strength λ for ReTAP on Qwen2.5-7B-Instruct with all-layer supervision. Each sweep varies one factor while holding the other at its default: $\lambda = 10$ for K and $K = 8$ for λ. The K sweep aggregates seeds 1234, 1235, and 1236 with matched task orders across settings; the λ sweep uses seed 1234. Full task scores appear in Appendix C.

Rollout count. Across $K \in \{1, 2, 4, 8, 16\}$, mean Avg over three seed-and-order settings ranges from 66.48 to 66.90, indicating limited sensitivity to rollout count. The default $K = 8$ achieves 66.70, compared with 66.90 for $K = 2$. These results suggest that smaller K can reduce rollout computation while maintaining comparable performance under the evaluated setting.

Preservation strength. Increasing λ from 0.01 to 1 improved Avg from 59.69 to 64.81 and BWT from -7.61 to -1.75. Across the evaluated coefficients $\lambda \in \{0.01, 0.1, 1, 5, 10, 20\}$, values below 1 yielded lower Avg and more negative BWT than the larger coefficients. Among $\lambda \in \{5, 10, 20\}$, $\lambda = 5$ achieved the highest Avg (66.80), whereas $\lambda = 20$ yielded the highest BWT ($+0.31$). The default $\lambda = 10$ achieved 66.66 Avg and -0.21 BWT. It combined a final Avg close to the best observed value with BWT near zero in this sweep.

4.4 Effect of Supervised Layers

With $K = 8$, $\lambda = 10$, and the same Historical-reference selection rule, we compared supervised-layer choices on Qwen2.5-7B-Instruct (Table 6). All-layer supervision achieved 66.66 Avg and -0.21 BWT. Supervising only the final layer reached 65.84 Avg and -0.46 BWT, while supervising the last four layers reached 66.48 Avg and -1.08 BWT. By comparison, first- and middle-layer settings, including their four-layer subsets, yielded 59.27–60.49 Avg and -8.01 to -6.54 BWT. In this experiment, supervising only the final layer yields an Avg 0.82 points below that of all-layer supervision. This suggests that final-layer matching may be a useful alternative when fewer supervised layers are desired.

4.5 Replay-Ratio Sensitivity

We next examine the sensitivity of ReTAP to the replay ratio while keeping the representation objective and online response-generation procedure fixed.

For the replay-ratio study, we varied $\rho \in \{0.005, 0.01, 0.02\}$, with a default of 0.01. LoRA rank 256, $\lambda = 10$, $K = 8$, and the training schedule were held fixed. Figure 4 compares final performance and retention across these replay ratios.

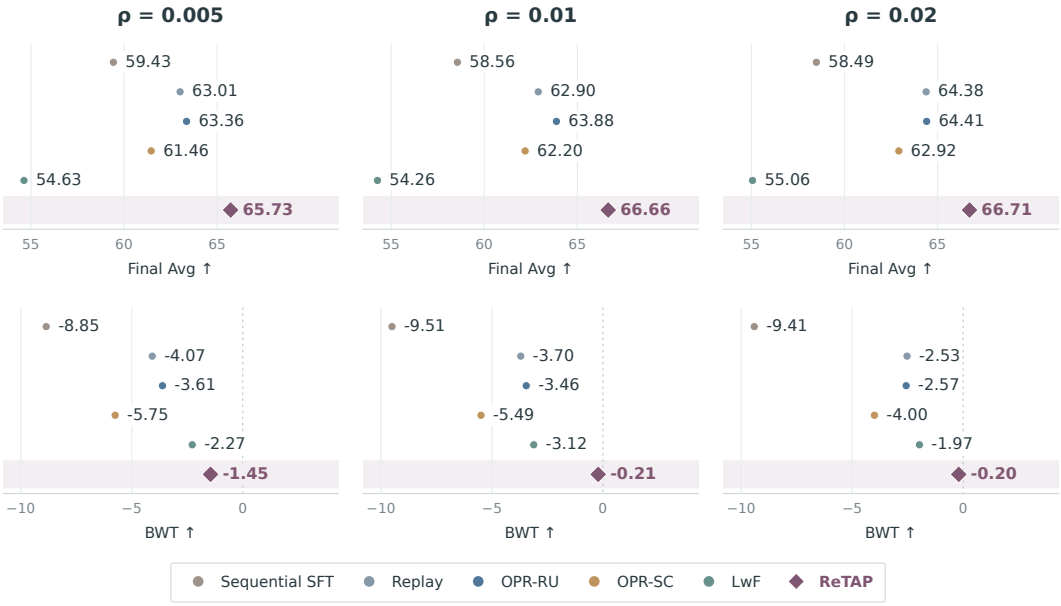

Figure 4: **Replay-ratio sensitivity on Qwen2.5-7B-Instruct.** Columns show $\rho = 0.005$, 0.01, and 0.02; the top and bottom rows show final Avg and BWT, respectively. Purple diamonds mark ReTAP, and circles mark the baselines. Ratios correspond to 25, 50, and 100 historical prompts per stage. Runs use seed 1234 and the original task order. Sequential SFT and LwF use no historical replay; the displayed columns are separately trained comparator runs.

ReTAP achieved higher Avg and BWT than all five comparators at each tested ratio. Its Avg gains over Replay were 2.72, 3.76, and 2.33 points at $\rho = 0.005$, 0.01, and 0.02, respectively. Increasing ρ from 0.005 to the default 0.01 improved Avg from 65.73 to 66.66 and BWT from -1.45 to -0.21; doubling it again yielded only a small change (66.71 Avg, -0.20 BWT). Thus, the default retained most of the observed benefit with half the largest prompt budget. Under prompt-level averaging, ρ affects both historical coverage and the historical contribution to Equation (4).

4.6 LoRA CAPACITY AND FULL FINE-TUNING

Figure 5 evaluates ReTAP across LoRA ranks 64/128/256 and full fine-tuning. ReTAP achieved the highest Avg and BWT among the four compared methods at every capacity. Under LoRA, its Avg gains over Replay were 4.53, 2.49, and 3.76 points at ranks 64, 128, and 256, respectively. The largest improvement occurred at rank 64, while substantial gains remained at both higher ranks. With full fine-tuning, ReTAP retained its advantage in both metrics, improving Avg over Replay by 0.78 points, although the margin was smaller than under LoRA. These results support the effectiveness of historical representation preservation across both adaptation regimes.

4.7 REPRESENTATION VERSUS OUTPUT-LEVEL DISTILLATION

Table 2 compares representation MSE with an OPD-inspired reverse KL objective (Lu & Lab, 2025). Among five tested reverse-KL coefficients, $\lambda = 10$ achieves the highest Average score (Avg) of 64.01 and Backward Transfer (BWT) of -2.61, with larger coefficients offering no further gains. ReTAP with representation MSE achieves an Avg of 66.66 and a BWT of 0.21, compared with 64.01 and 2.61 for the best evaluated reverse-KL setting. Under this experimental setting, representation MSE therefore achieves higher final average performance and BWT than the tested reverse-KL configurations.

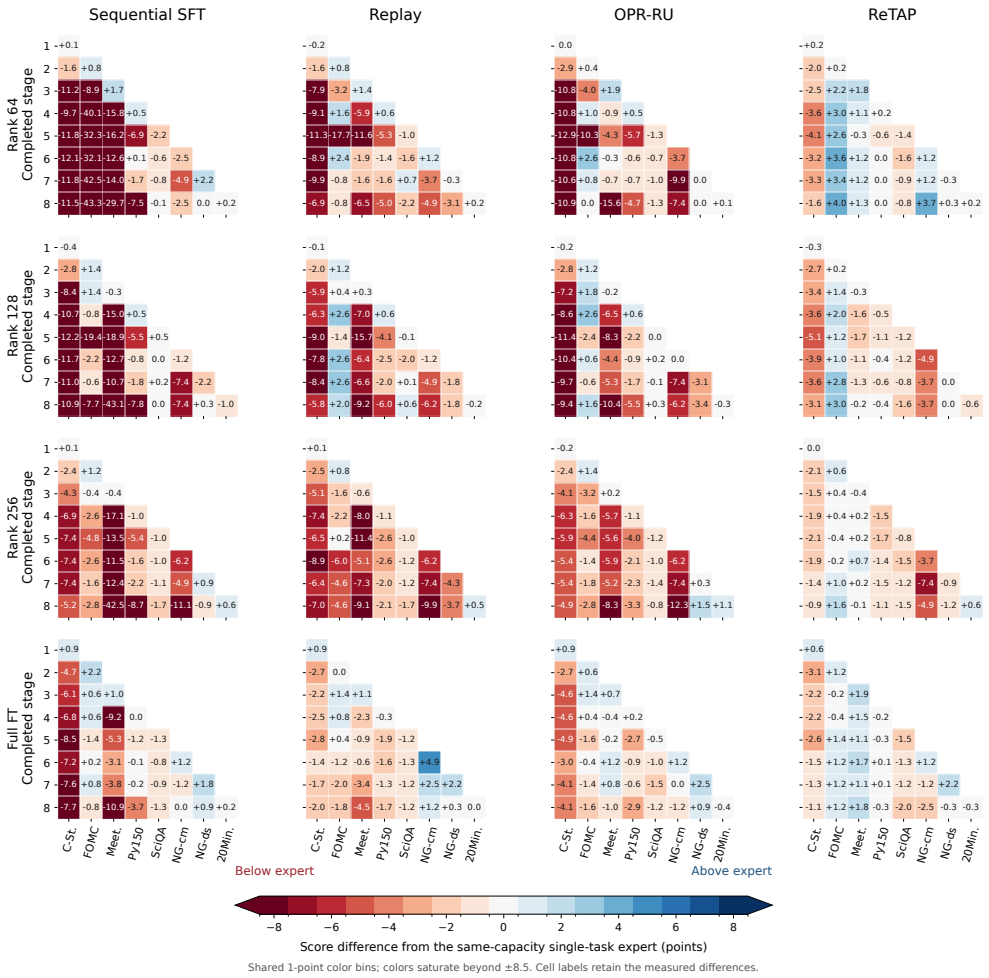

Figure 5: **Capacity-wise gaps to single-task benchmarks on Qwen2.5-7B-Instruct.** Rows: capacities; columns: methods. Each panel shows stage-by-task scores minus same-capacity single-task evaluation scores. Red/blue denote deficits/gains; colors saturate at ± 8.5. Seed 1234, original order.

Table 2: **Representation MSE versus reverse KL on Qwen2.5-7B-Instruct.** Comparison between five reverse-KL settings and ReTAP (representation MSE) under $K = 8$, $\rho = 0.01$, and original task order. Both losses use student-generated responses and identical checkpoint selection. Reverse KL $\mathrm{KL}(\pi_\theta \| \pi_\phi)$ uses full vocabulary at temperature 1. Bold indicates best results (↑).

Loss	λ	Avg ↑	BWT ↑	Loss	λ	Avg ↑	BWT ↑
Reverse KL	0.01	60.35	-6.28	Reverse KL	20	62.70	-5.03
Reverse KL	0.1	61.02	-7.03	Reverse KL	50	63.35	-2.81
Reverse KL	10	64.01	-2.61	ReTAP (MSE)	10	**66.66**	**-0.21**

5 Conclusion

Continual instruction tuning requires a single model to learn new tasks while retaining prior capabilities. ReTAP addresses the mismatch between stored replay responses and the evolving student's behavior by separating trajectory generation from preservation targets: the student generates responses to historical prompts, while frozen historical checkpoints provide task-specific representation targets on the same responses. Across eight TRACE tasks and three model backbones, ReTAP achieves higher final average performance and backward transfer than the evaluated continual-learning baselines, approaching the performance of independently trained single-task models. These results suggest that matching task-specific reference representations on current student responses can reduce forgetting while supporting new-task learning.

AI USE DISCLOSURE

OpenAI Codex provided technical and editorial assistance during this study. The authors reviewed its outputs and take responsibility for the analyses, reported results, references, and final manuscript.

REPRODUCIBILITY STATEMENT

Section 3 describes ReTAP, and Appendix A details training configurations, historical prompt sampling, and evaluation procedures. Appendix B presents the representation analysis, and Appendix C provides detailed hyperparameter and layer-ablation scores. The accompanying source package contains the tables and figures used in this manuscript.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning, 2019. URL https://arxiv.org/abs/1902.10486.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting, 2026a. URL https://arxiv.org/abs/2510.18874.

Yan Chen, Taojie Zhu, Meng Zhang, Xin Chen, Jiaqi Huang, Dongyang Xu, and Yizhi Wang. On-policy replay for continual supervised fine-tuning, 2026b. URL https://arxiv.org/abs/2605.29495.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. MeetingBank: A benchmark dataset for meeting summarization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16409–16423, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.906. URL https://aclanthology.org/2023.acl-long.906/.

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

Zhizhong Li and Derek Hoiem. Learning without forgetting, 2017. URL https://arxiv.org/abs/1606.09282.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning, 2022. URL https://arxiv.org/abs/1706.08840.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022. URL https://arxiv.org/abs/2209.09513.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. Codexglue: A machine learning benchmark dataset for code understanding and generation, 2021. URL https://arxiv.org/abs/2102.04664.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3505–3523, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.246. URL https://aclanthology.org/2022.acl-long.246/.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Annette Rios, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. A new dataset and efficient baselines for document-level text simplification in German. In Giuseppe Carenini, Jackie Chi Kit Cheung, Yue Dong, Fei Liu, and Lu Wang (eds.), *Proceedings of the Third Workshop on New Frontiers in Summarization*, pp. 152–161, Online and in Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.newsum-1.16. URL https://aclanthology.org/2021.newsum-1.16/.

Agam Shah, Suvan Paturi, and Sudheer Chava. Trillion dollar words: A new financial dataset, task & market analysis. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6664–6679, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.368. URL https://aclanthology.org/2023.acl-long.368/.

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

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. Lamol: Language modeling for lifelong language learning, 2019. URL https://arxiv.org/abs/1909.03329.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. Trace: A comprehensive benchmark for continual learning in large language models, 2023. URL https://arxiv.org/abs/2310.06762.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415, 2016. doi: 10.1162/tacl_a_00107. URL https://aclanthology.org/Q16-1029/.

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.

Shenzhi Yang, Guangcheng Zhu, Bowen Song, Haobo Wang, Mingxuan Xia, Xing Zheng, Yingfan Ma, Zhongqi Chen, Weiqiang Wang, Junbo Zhao, and Gang Chen. Oprd: On-policy representation distillation, 2026. URL https://arxiv.org/abs/2606.06021.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. C-STANCE: A large dataset for Chinese zero-shot stance detection. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13369–13385, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.747. URL https://aclanthology.org/2023.acl-long.747/.

A TRAINING AND EVALUATION DETAILS

A.1 COMMON CONFIGURATION

The main backbone is Qwen2.5-7B-Instruct, pinned to revision `a09a35458c702b33eeacc393d103063234e8bc28`; cross-model runs use Qwen3-1.7B and Qwen3-4B. Unless varied in the capacity study, adapters use LoRA rank 256, scale $\alpha = 512$, and dropout 0, targeting the attention query/key/value/output projections and the MLP gate/up/down projections. Embeddings, the LM head, and normalization parameters are frozen. The optimizer is AdamW with learning rate 10^{-5}, cosine decay, zero warmup, and zero weight decay. Training uses BF16, FlashAttention-2, and gradient checkpointing. The main table and rollout-count sweep aggregate three runs; other ablations, resource studies, and original-order ReTAP results use seed 1234. Qwen3 runs use non-thinking mode for training, rollouts, teacher selection, and evaluation. Optimizer and scheduler are initialized per stage; the learned student adapter is carried forward.

The ordered epoch counts are $(5, 3, 7, 5, 3, 5, 5, 7)$. The nominal global batch is 128: two GPUs, eight items per device per microbatch, and eight accumulation steps. Historical prompts count as items before expansion to sampled responses. Each single-task evaluation model resets the base and adapter for its task; the completed 20Minuten run used one GPU and 16 accumulation steps, retaining global batch 128. This is a batch-layout difference and does not imply bitwise equivalent optimization. The Historical loss sweeps use the same first-stage SFT configuration. Stage one uses the ordinary response-token mean CE trainer; prompt-mean joint training begins at later stages. Partial final batches and any distributed sampler padding follow the training framework; "once per epoch" refers to nominal dataset membership.

A.2 HISTORICAL SAMPLING AND BASELINE ADAPTATIONS

At stage $t > 1$, task quotas partition 50 historical items by default as evenly as possible among $t - 1$ tasks; remainder items are assigned in task order. The default replay ratio is $\rho = 0.01$; Section 4.5 additionally evaluates 0.005 and 0.02 under the same allocation rule. Replay and ReTAP sample from historical training pools without replacement within each task allocation, using a deterministic seed derived from the main seed, stage index, and task index. The selected items are mixed with the current data and revisited across the stage's epochs. They are not repeated in every optimizer step. The next stage draws a new set from the available pools. This protocol does not provide a strict 50-record lifelong storage guarantee.

OPR-RU and OPR-SC build historical replay targets using student generation and task-wise selection. RU ranks candidates with the task metric against gold answers. SC ranks them by mean log probability of the generated tokens, including EOS, and does not use gold reward for selection. The implementation corrects the upstream SC aggregation to average the selected tokens' log probabilities rather than all returned top-k alternatives. Candidate generation uses temperature 0.1; the generation and selection workload may access substantially more historical prompts than the 50 replayed items. Ordinary replay baselines use response-token cross-entropy on the mixed batch.

Our LwF adaptation (Li & Hoiem, 2017) uses a frozen previous-stage model to compute forward KL, $\mathrm{KL}(p^{(\tau)}_{\text{teacher}} \| p^{(\tau)}_{\text{student}})$, on the current gold-response predictive positions, with temperature $\tau = 2$, the τ^2 scaling factor (Hinton et al., 2015), and coefficient 1, alongside current-task CE. It uses no historical examples. This is a specified language-model adaptation of LwF, not the original vision architecture or a teacher-matched isolation of our representation loss.

ReTAP uses $K = 8$, temperature 1, top-$p = 1$, top-$k = 0$, and fresh on-access sampling. C-STANCE and FOMC have a one-token response cap; all other tasks have a 512-token cap, further bounded by $2048 - |\text{prompt}|$. EOS ends a response. No explicit reasoning-format prompt is added. The frozen teacher forwards each generated sequence; its detached states are temporarily stored in FP16, and the MSE subtraction and reduction use FP32. Student and teacher model forwards use BF16. References are implemented as named adapters over a shared frozen base. ReTAP stores completed student adapters as Historical references.

For historical task j before stage t, ReTAP selects a reference from

$$\mathcal{C}_{j,t}^{\text{Hist}} = \{\theta_j, \ldots, \theta_{t-1}\}, \qquad \phi_{j,t} \in \mathcal{C}_{j,t}^{\text{Hist}}. \tag{7}$$

The selected reference remains frozen throughout stage t. No independently trained expert belongs to this candidate pool.

The supervised-layer, rollout-count, preservation-strength, and output-distillation studies all use Historical references selected from each run's completed student checkpoints. No independent task-expert checkpoint is used as a ReTAP training reference.

A.3 Repeated runs and statistical summaries

Table 1 and the K sweep report arithmetic means and sample standard deviations across seeds 1234, 1235, and 1236. Each seed has a prescribed task order, shared by every compared method:

- 1234: C-STANCE, FOMC, MeetingBank, Py150, ScienceQA, NumGLUE-cm, NumGLUE-ds, 20Minuten.
- 1235: MeetingBank, FOMC, C-STANCE, NumGLUE-cm, Py150, 20Minuten, ScienceQA, NumGLUE-ds.
- 1236: C-STANCE, FOMC, ScienceQA, MeetingBank, Py150, NumGLUE-ds, NumGLUE-cm, 20Minuten.

Epoch counts stay attached to their tasks when the order changes. Each completed eight-stage run is one replicate. For a metric z, we report $\bar{z} = \frac{1}{3}\sum_{s=1}^{3} z_s$ and $\mathrm{SD}(z) = \sqrt{\frac{1}{2}\sum_{s=1}^{3}(z_s - \bar{z})^2}$. Avg and BWT are computed separately within each run before averaging; BWT follows that run's task order. Per-task statistics align tasks by name. Thus, the reported SD describes variation across the three seed-and-order settings, with comparisons paired by setting.

A.4 Evaluation and metric extraction

Each completed stage is evaluated on all tasks seen so far. There is exactly one greedy generation per example (temperature 0), with the one-token caps for C-STANCE/FOMC and 512 otherwise. The same decoding and scoring rules apply to all continual-learning rows in Table 1. Evaluation uses the prompt as input and never the gold answer. This protocol differs from the upstream OPR evaluator's repeated stochastic decoding; published cross-paper numbers are not mixed into our tables.

The TRACE tasks draw on C-STANCE (Zhao et al., 2023), FOMC (Shah et al., 2023), MeetingBank (Hu et al., 2023), Py150 as used in CodeXGLUE (Lu et al., 2021), ScienceQA (Lu et al., 2022), NumGLUE (Mishra et al., 2022), and 20Minuten (Rios et al., 2021). These citations identify the source datasets; the following rules specify the evaluator used in our experiments.

In our evaluator, C-STANCE, FOMC, and ScienceQA receive 100 if the first output character matches the first gold-answer character, otherwise 0. MeetingBank uses ROUGE-L F-measure (Lin, 2004) multiplied by 100. Py150 uses a fuzzy edit-ratio score after replacing the benchmark's literal placeholders. For NumGLUE-cm/ds, our evaluator extracts the final numeric substring and compares it to the gold-answer string. For 20Minuten, we use SARI (Xu et al., 2016). These extraction rules are held constant across methods, although scores can be sensitive to output formatting.

B Representation analysis

B.1 Representation agreement and next-token behavior

Representation matching has a direct behavioral interpretation when student and reference share a frozen output head, as in our LoRA setting. At a supervised prefix s, let $h_\theta(s)$ and $h_\phi(s)$ denote the final normalized hidden states, and write $p_\theta(\cdot \mid s) = \mathrm{softmax}(W h_\theta(s) + b)$, with an analogous definition for p_ϕ.

Proposition 1 (Agreement at supervised states)**.** *For any distribution μ over supervised prefixes, define $e_L(s) = \|h_\theta(s) - h_\phi(s)\|_2^2/d$. With a common frozen head (W, b),*

$$\mathbb{E}_{s\sim\mu}\,\mathrm{TV}(p_\theta(\cdot \mid s), p_\phi(\cdot \mid s)) \le \min\left\{1, \frac{\|W\|_2}{2\sqrt{2}}\sqrt{d\,\mathbb{E}_\mu e_L}\right\}. \tag{8}$$

If the loss averages over L layers including this final state, then $\mathbb{E}_\mu e_L \le L\mathbb{E}_\mu e_{\mathrm{all}}$.

The result connects final-state agreement to next-token agreement at supervised prefixes. The proof below follows from softmax smoothness and Pinsker's inequality. Task references specify the targets, student trajectories determine the supervised prefixes, and representation matching controls agreement at those prefixes.

B.2 Proof of Proposition 1

For $F(z) = \log\sum_i \exp z_i$, the Hessian is $\nabla^2 F(z) = \mathrm{diag}(p) - pp^\top$. Its spectral norm is at most $1/2$: in row i, the diagonal is $p_i(1-p_i)$ and the sum of absolute off-diagonal entries is the same, so the Gershgorin upper bound is $2p_i(1-p_i) \le 1/2$. Consequently its Bregman divergence satisfies

$$D_F(a, b) \le \tfrac{1}{4}\|a - b\|_2^2. \tag{9}$$

The softmax identity gives $D_F(a, b) = \mathrm{KL}(\mathrm{softmax}(b)\|\,\mathrm{softmax}(a))$. Set $a = Wh_\theta + b_0$ and $b = Wh_\phi + b_0$ for the common output bias b_0; then $a - b = W(h_\theta - h_\phi)$. Pinsker's inequality then yields

$$\mathrm{TV}(p_\theta, p_\phi) \le \sqrt{\tfrac{1}{2}\,\mathrm{KL}(p_\phi\|p_\theta)} \le \frac{\|W\|_2}{2\sqrt{2}}\|h_\theta - h_\phi\|_2. \tag{10}$$

Take expectations and apply Cauchy–Schwarz, along with $\mathrm{TV} \le 1$, to obtain Equation (8). Define $e_{\mathrm{all}}(s) = (Ld)^{-1}\sum_{\ell\in\mathcal{L}}\|h_\theta^\ell(s) - h_\phi^\ell(s)\|_2^2$. The layer-average statement follows because all squared errors are nonnegative, so the final-layer error cannot exceed L times this uniform mean. This proof concerns exact real-valued states; finite-precision target storage adds a numerical approximation.

Scope. The proposition explains how final-state matching constrains next-token distributions at supervised prefixes under a common frozen head. It is a local mechanism statement, not a guarantee of sequence-level task performance or a comparison between MSE and KL. It does not apply directly to full fine-tuning when the output head changes.

C Full ablation results

C.1 Rollout count and preservation strength

Tables 3 and 4 provide the eight final task scores underlying Figure 3. Both sweeps use Qwen2.5-7B-Instruct, raw representation MSE, all layers, and all supervised response prefixes. The K sweep reports means and sample SD across three seed-and-order settings; the λ sweep reports seed 1234 over 0.01–20. Thus, the default $K = 8$ row aggregates three runs, whereas the default $\lambda = 10$ row reports only the seed-1234 run.

Table 3: Rollout-count sweep for ReTAP on Qwen2.5-7B-Instruct. Rows vary K with $\lambda = 10$ and all-layer supervision. Values are means with sample SD beneath across seeds 1234/1235/1236 and their matched task orders. Bold marks the highest mean Avg and BWT.

K	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.	Avg.	BWT
					Qwen2.5-7B-Instruct					
1	59.23 ±0.46	71.24 ±0.12	64.86 ±0.78	67.48 ±0.09	88.20 ±0.35	72.84 ±2.47	69.54 ±0.92	41.21 ±0.14	66.83 ±0.24	−0.28 ±0.63
2	59.52 ±0.58	71.84 ±0.65	65.19 ±0.79	67.38 ±0.21	88.15 ±0.39	72.43 ±0.71	69.44 ±2.31	41.25 ±0.16	**66.90** ±0.42	**−0.15** ±0.80
4	58.82 ±0.46	71.30 ±0.51	64.53 ±0.32	67.47 ±0.20	88.08 ±0.80	69.96 ±3.56	70.56 ±0.64	41.08 ±0.14	66.48 ±0.21	−0.61 ±0.36
8	59.18 ±0.25	71.51 ±0.62	64.88 ±0.15	67.57 ±0.41	87.97 ±0.89	71.19 ±0.71	70.26 ±0.47	41.07 ±0.09	66.70 ±0.32	−0.40 ±0.34
16	58.83 ±0.46	71.57 ±1.05	64.87 ±0.59	67.40 ±0.63	88.22 ±0.76	72.84 ±3.70	70.26 ±0.89	40.91 ±0.26	66.86 ±0.37	−0.29 ±0.92

Table 4: Preservation-strength sweep for ReTAP on Qwen2.5-7B-Instruct. Rows vary $\lambda \in \{0.01, 0.1, 1, 5, 10, 20\}$ with $K = 8$ and all-layer supervision. Each row uses seed 1234 and the original task order. Bold marks the highest Avg and BWT in the displayed range.

Setting	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.	Avg.	BWT
Qwen2.5-7B-Instruct										
0.01	54.30	67.74	31.78	61.30	88.85	65.43	67.08	41.06	59.69	-7.61
0.1	55.40	66.94	39.78	65.18	87.40	67.90	68.62	40.69	61.49	-6.08
1	58.25	69.96	54.84	65.23	88.50	70.37	70.15	41.21	64.81	-1.75
5	59.30	71.98	65.48	67.48	88.20	70.37	70.46	41.13	**66.80**	-0.41
10	58.95	71.37	64.97	67.60	87.75	71.60	69.85	41.18	66.66	-0.21
20	59.85	71.77	65.65	67.99	87.35	64.20	69.85	40.99	65.96	**+0.31**

C.2 Loss aggregation

To rule out loss aggregation as a confounding factor, we compare different aggregation schemes on Qwen2.5-7B-Instruct using seed 1234, the original task order, LoRA rank 256, and 50 historical items per stage. ReTAP uses Historical references, $K = 8$, $\lambda = 10$, and all-layer representation MSE. In prompt-mean OPR-RU, each current or historical answer contributes its mean response-token CE, with equal weight per item within each microbatch.

Table 5: Loss aggregation comparison on Qwen2.5-7B-Instruct. Each row reports one eight-stage run. OPR-RU supervises selected historical answers with CE; ReTAP uses representation MSE with Historical references. Higher Avg and BWT are better.

Method	Aggregation	Avg ↑	BWT ↑
OPR-RU	Token-mean (original)	63.88	−3.46
OPR-RU	Prompt-mean	63.75	−3.39
ReTAP	Token-weighted	65.33	−1.53
ReTAP	Prompt-mean (default)	**66.66**	**−0.21**

ReTAP achieves higher Avg and BWT than OPR-RU when both methods use prompt-level averaging. ReTAP with token-weighted aggregation also outperforms the original OPR-RU baseline on both metrics in this run. These results suggest that the observed performance gap is not solely attributable to the aggregation scheme.

C.3 Supervised layers

Table 6 gives the full task scores for the layer comparison in Section 4.4. The all-layer row reports the default configuration ($K = 8$, $\lambda = 10$).

Table 6: Supervised-layer ablation on Qwen2.5-7B-Instruct. All rows use $\lambda = 10$, $K = 8$, seed 1234, and the original task order. Rows select individual layers, four-layer groups, or all non-embedding hidden states. The all-layer default includes the final normalized state. Bold marks the highest Avg and BWT.

Setting	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.	Avg.	BWT
Qwen2.5-7B-Instruct										
first	53.85	65.93	38.07	60.89	89.00	64.20	68.00	41.12	60.13	-6.94
first4	54.15	66.53	29.82	61.45	88.80	66.67	70.77	41.59	59.97	-8.01
middle	55.20	68.35	22.90	59.76	87.60	71.61	67.39	41.38	59.27	-7.42
middle4	56.65	69.76	29.11	59.77	87.75	71.61	68.00	41.24	60.49	-6.54
last	60.65	70.97	64.90	67.55	87.85	65.43	68.92	40.46	65.84	-0.46
last4	60.75	70.97	65.28	67.94	87.30	71.61	67.08	40.92	66.48	-1.08
all (default)	58.95	71.37	64.97	67.60	87.75	71.60	69.85	41.18	**66.66**	**-0.21**

D Additional ReTAP results

Independently trained single-task models are used only for performance evaluation; they do not provide ReTAP training targets.

D.1 Original-order ReTAP results

ReTAP reached 66.66, 60.15, and 65.38 Avg on Qwen2.5-7B-Instruct, Qwen3-1.7B, and Qwen3-4B, respectively. These individual-run results correspond to the original-order setting of the main comparison.

Budget and capacity studies. Sections 4.5 and 4.6 report replay-ratio, LoRA-rank, and full-fine-tuning results. Independently trained single-task scores in the capacity study serve only as evaluation benchmarks.

E STAGE-WISE TASK PERFORMANCE

Tables 7–24 show one complete run per model and method: the highest final Avg among seeds 1234, 1235, and 1236. Each table retains that run's actual task order and all stage scores; these selected trajectories complement the three-run means in Table 1. Rows show scores after each fine-tuning stage; adjacent rows track changes on earlier tasks. Bold diagonal entries mark initial acquisition, and the shaded last row shows final performance. Dashes denote tasks not yet evaluated.

E.1 QWEN2.5-7B-INSTRUCT

Table 7: Sequential SFT (seed 1235). Final Avg 63.45; BWT −3.63.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**64.92**	–	–	–	–	–	–	–
2: FOMC	64.35	**68.35**	–	–	–	–	–	–
3: C-St.	62.92	68.75	**59.75**	–	–	–	–	–
4: NG-cm	63.57	68.75	61.00	**74.07**	–	–	–	–
5: Py150	52.82	70.16	59.65	71.60	**67.75**	–	–	–
6: 20Min.	50.76	70.56	60.00	70.37	64.33	**40.71**	–	–
7: SciQA	38.80	66.94	57.60	61.73	63.56	40.32	**89.15**	–
8: NG-ds	44.64	69.56	58.20	70.37	66.37	40.94	89.20	**68.31**

Table 8: Replay (seed 1235). Final Avg 64.93; BWT −1.81.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**65.02**	–	–	–	–	–	–	–
2: FOMC	62.10	**69.76**	–	–	–	–	–	–
3: C-St.	59.37	69.76	**59.50**	–	–	–	–	–
4: NG-cm	60.51	66.94	60.10	**71.60**	–	–	–	–
5: Py150	56.64	67.14	59.30	70.37	**68.35**	–	–	–
6: 20Min.	55.32	67.74	59.10	70.37	67.02	**40.97**	–	–
7: SciQA	52.32	66.94	57.70	75.31	65.38	40.40	**88.30**	–
8: NG-ds	57.68	71.37	58.05	69.14	65.46	40.48	88.65	**68.62**

Table 9: OPR-RU (seed 1235). Final Avg 65.51; BWT −2.17.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**65.21**	–	–	–	–	–	–	–
2: FOMC	64.35	**70.97**	–	–	–	–	–	–
3: C-St.	62.89	69.15	**59.45**	–	–	–	–	–
4: NG-cm	62.63	70.56	57.50	**75.31**	–	–	–	–
5: Py150	57.30	69.15	58.70	70.37	**67.93**	–	–	–
6: 20Min.	57.86	69.15	58.15	71.60	67.13	**40.46**	–	–
7: SciQA	54.83	66.73	55.95	69.14	62.64	40.84	**89.75**	–
8: NG-ds	57.86	68.55	57.70	72.84	66.74	41.23	89.00	**70.15**

QWEN2.5-7B-INSTRUCT (CONTINUED)

Table 10: OPR-SC (seed 1235). Final Avg 63.36; BWT -4.53.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**64.45**	–	–	–	–	–	–	–
2: FOMC	59.51	**70.16**	–	–	–	–	–	–
3: C-St.	59.82	69.56	**59.00**	–	–	–	–	–
4: NG-cm	60.19	70.16	59.60	**75.31**	–	–	–	–
5: Py150	50.13	68.55	58.45	71.60	**68.09**	–	–	–
6: 20Min.	52.15	64.31	57.45	74.07	67.41	**41.15**	–	–
7: SciQA	50.32	64.52	56.45	72.84	65.72	40.47	**89.05**	–
8: NG-ds	51.54	63.91	56.40	70.37	66.96	37.73	88.60	**71.38**

Table 11: LwF (seed 1235). Final Avg 62.01; BWT -0.76.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**64.85**	–	–	–	–	–	–	–
2: FOMC	65.27	**67.54**	–	–	–	–	–	–
3: C-St.	64.24	66.13	**60.85**	–	–	–	–	–
4: NG-cm	64.23	65.32	61.35	**64.20**	–	–	–	–
5: Py150	62.68	64.92	61.55	65.43	**62.37**	–	–	–
6: 20Min.	61.97	65.32	61.90	65.43	59.95	**38.68**	–	–
7: SciQA	60.22	65.12	62.25	65.43	46.88	39.16	**87.55**	–
8: NG-ds	59.93	66.73	61.95	65.43	59.98	39.15	87.55	**55.38**

Table 12: ReTAP (seed 1235). Final Avg 67.04; BWT -0.20.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**64.94**	–	–	–	–	–	–	–
2: FOMC	65.13	**71.57**	–	–	–	–	–	–
3: C-St.	64.52	71.77	**59.25**	–	–	–	–	–
4: NG-cm	65.18	71.98	59.55	**74.07**	–	–	–	–
5: Py150	65.14	70.56	59.20	74.07	**67.36**	–	–	–
6: 20Min.	64.79	71.57	59.45	71.60	67.50	**40.90**	–	–
7: SciQA	65.17	71.98	59.35	71.60	67.47	40.94	**88.85**	–
8: NG-ds	64.71	72.18	59.15	71.60	67.96	41.02	88.95	**70.77**

E.2 QWEN3-1.7B

Table 13: Sequential SFT (seed 1235). Final Avg 54.41; BWT −7.02.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**61.00**	–	–	–	–	–	–	–
2: FOMC	57.39	**65.52**	–	–	–	–	–	–
3: C-St.	54.44	61.69	**55.05**	–	–	–	–	–
4: NG-cm	46.57	63.71	52.05	**61.73**	–	–	–	–
5: Py150	32.12	63.51	50.40	55.56	**64.04**	–	–	–
6: 20Min.	26.46	60.08	55.80	56.79	52.74	**39.83**	–	–
7: SciQA	23.78	58.67	53.70	39.51	41.17	38.02	**77.85**	–
8: NG-ds	40.71	60.08	49.75	50.62	57.27	39.44	78.00	**59.38**

Table 14: Replay (seed 1234). Final Avg 56.94; BWT −4.76.

After learning	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.
1: C-St.	**54.20**	–	–	–	–	–	–	–
2: FOMC	52.85	**68.55**	–	–	–	–	–	–
3: Meet.	52.40	64.72	**61.59**	–	–	–	–	–
4: Py150	50.55	64.52	53.44	**63.81**	–	–	–	–
5: SciQA	51.75	61.69	48.63	52.24	**79.05**	–	–	–
6: NG-cm	53.90	61.49	54.11	59.10	76.45	**64.20**	–	–
7: NG-ds	52.95	61.90	50.56	61.73	78.45	48.15	**57.54**	–
8: 20Min.	56.15	63.10	47.31	59.53	79.20	54.32	56.00	**39.91**

Table 15: OPR-RU (seed 1236). Final Avg 57.38; BWT −2.77.

After learning	C-St.	FOMC	SciQA	Meet.	Py150	NG-ds	NG-cm	20Min.
1: C-St.	**54.10**	–	–	–	–	–	–	–
2: FOMC	52.80	**68.35**	–	–	–	–	–	–
3: SciQA	53.50	64.72	**78.60**	–	–	–	–	–
4: Meet.	52.50	64.92	76.00	**61.67**	–	–	–	–
5: Py150	53.45	63.91	75.70	56.74	**63.60**	–	–	–
6: NG-ds	50.40	66.13	77.15	56.38	62.77	**56.62**	–	–
7: NG-cm	48.65	62.10	78.15	55.91	62.75	53.23	**55.56**	–
8: 20Min.	53.85	66.33	77.80	52.61	60.37	51.38	56.79	**39.89**

QWEN3-1.7B (CONTINUED)

Table 16: OPR-SC (seed 1234). Final Avg 55.94; BWT −5.59.

After learning	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.
1: C-St.	**54.20**	–	–	–	–	–	–	–
2: FOMC	50.30	**68.35**	–	–	–	–	–	–
3: Meet.	47.95	63.10	**61.57**	–	–	–	–	–
4: Py150	45.20	61.09	51.95	**63.96**	–	–	–	–
5: SciQA	47.90	55.44	50.84	52.98	**78.45**	–	–	–
6: NG-cm	47.30	45.16	50.41	59.31	76.30	**60.49**	–	–
7: NG-ds	47.35	42.94	46.09	56.14	75.95	50.62	**59.69**	–
8: 20Min.	50.25	62.70	44.64	56.44	75.25	60.49	57.85	**39.86**

Table 17: LwF (seed 1235). Final Avg 51.94; BWT −3.71.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**61.40**	–	–	–	–	–	–	–
2: FOMC	60.92	**63.51**	–	–	–	–	–	–
3: C-St.	59.10	53.02	**55.50**	–	–	–	–	–
4: NG-cm	57.47	54.84	55.75	**34.57**	–	–	–	–
5: Py150	55.10	52.02	55.65	35.80	**56.88**	–	–	–
6: 20Min.	51.32	52.02	55.15	37.04	54.39	**37.26**	–	–
7: SciQA	52.09	54.03	56.75	39.51	36.18	38.06	**79.80**	–
8: NG-ds	49.08	57.86	55.25	35.80	53.26	38.29	73.40	**52.62**

Table 18: ReTAP (seed 1234). Final Avg 60.15; BWT +0.07.

After learning	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.
1: C-St.	**54.00**	–	–	–	–	–	–	–
2: FOMC	53.85	**68.35**	–	–	–	–	–	–
3: Meet.	52.20	66.94	**54.75**	–	–	–	–	–
4: Py150	52.40	66.13	53.87	**62.49**	–	–	–	–
5: SciQA	53.95	65.32	53.39	62.86	**78.60**	–	–	–
6: NG-cm	54.35	68.35	54.32	62.37	78.75	**65.43**	–	–
7: NG-ds	54.65	67.94	53.64	62.17	79.55	61.73	**57.85**	–
8: 20Min.	54.65	66.53	54.07	62.18	79.70	66.67	58.15	**39.21**

E.3 QWEN3-4B

Table 19: Sequential SFT (seed 1236). Final Avg 60.86; BWT −4.99.

After learning	C-St.	FOMC	SciQA	Meet.	Py150	NG-ds	NG-cm	20Min.
1: C-St.	**59.25**	–	–	–	–	–	–	–
2: FOMC	56.40	**68.55**	–	–	–	–	–	–
3: SciQA	55.15	68.95	**88.40**	–	–	–	–	–
4: Meet.	55.15	61.90	46.45	**64.27**	–	–	–	–
5: Py150	54.20	66.94	81.25	57.15	**66.34**	–	–	–
6: NG-ds	53.45	65.12	87.05	58.50	65.56	**67.69**	–	–
7: NG-cm	53.05	64.72	85.85	57.80	64.82	64.92	**66.67**	–
8: 20Min.	56.15	66.94	86.25	41.19	64.15	63.69	67.90	**40.61**

Table 20: Replay (seed 1234). Final Avg 62.07; BWT −2.74.

After learning	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.
1: C-St.	**59.75**	–	–	–	–	–	–	–
2: FOMC	56.35	**69.76**	–	–	–	–	–	–
3: Meet.	56.25	69.35	**64.35**	–	–	–	–	–
4: Py150	55.35	68.95	54.57	**65.70**	–	–	–	–
5: SciQA	56.60	66.73	54.20	62.24	**89.40**	–	–	–
6: NG-cm	56.95	67.14	52.68	64.58	89.20	**59.26**	–	–
7: NG-ds	56.45	66.53	56.07	64.24	89.25	60.49	**67.38**	–
8: 20Min.	57.20	64.92	51.99	63.62	88.05	64.20	66.46	**40.16**

Table 21: OPR-RU (seed 1234). Final Avg 63.05; BWT −2.04.

After learning	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.
1: C-St.	**59.75**	–	–	–	–	–	–	–
2: FOMC	56.25	**69.35**	–	–	–	–	–	–
3: Meet.	55.20	69.96	**64.22**	–	–	–	–	–
4: Py150	56.75	67.54	59.94	**65.86**	–	–	–	–
5: SciQA	55.60	66.94	59.39	61.56	**89.10**	–	–	–
6: NG-cm	54.10	68.15	58.43	64.34	88.50	**64.20**	–	–
7: NG-ds	53.05	62.90	59.35	64.40	88.30	65.43	**65.54**	–
8: 20Min.	55.95	65.52	58.03	64.27	88.40	66.67	64.92	**40.67**

QWEN3-4B (CONTINUED)

Table 22: OPR-SC (seed 1236). Final Avg 62.29; BWT −3.12.

After learning	C-St.	FOMC	SciQA	Meet.	Py150	NG-ds	NG-cm	20Min.
1: C-St.	**58.95**	–	–	–	–	–	–	–
2: FOMC	56.10	**69.35**	–	–	–	–	–	–
3: SciQA	54.60	69.76	**87.60**	–	–	–	–	–
4: Meet.	54.45	66.33	84.65	**65.28**	–	–	–	–
5: Py150	53.15	66.33	84.65	58.83	**66.35**	–	–	–
6: NG-ds	53.00	64.72	86.60	59.48	65.36	**66.46**	–	–
7: NG-cm	52.90	66.73	86.45	59.52	65.10	64.31	**65.43**	–
8: 20Min.	55.00	65.12	86.70	56.15	64.24	64.92	65.43	**40.72**

Table 23: LwF (seed 1235). Final Avg 58.64; BWT −4.98.

After learning	Meet.	FOMC	C-St.	NG-cm	Py150	20Min.	SciQA	NG-ds
1: Meet.	**64.17**	–	–	–	–	–	–	–
2: FOMC	61.14	**68.35**	–	–	–	–	–	–
3: C-St.	60.42	58.06	**59.75**	–	–	–	–	–
4: NG-cm	56.94	58.27	60.20	**59.26**	–	–	–	–
5: Py150	48.79	55.04	59.90	55.56	**63.21**	–	–	–
6: 20Min.	44.62	56.85	59.60	54.32	60.42	**38.21**	–	–
7: SciQA	53.36	57.86	58.15	53.09	45.33	38.37	**89.45**	–
8: NG-ds	51.62	57.26	58.40	54.32	59.75	38.70	87.50	**61.54**

Table 24: ReTAP (seed 1234). Final Avg 65.38; BWT −0.44.

After learning	C-St.	FOMC	Meet.	Py150	SciQA	NG-cm	NG-ds	20Min.
1: C-St.	**59.95**	–	–	–	–	–	–	–
2: FOMC	58.60	**70.56**	–	–	–	–	–	–
3: Meet.	58.05	69.76	**64.57**	–	–	–	–	–
4: Py150	58.55	69.15	62.60	**65.46**	–	–	–	–
5: SciQA	57.95	68.75	62.75	64.86	**88.50**	–	–	–
6: NG-cm	58.65	67.14	63.44	65.08	88.05	**67.90**	–	–
7: NG-ds	58.50	67.54	64.11	65.28	88.70	69.14	**68.31**	–
8: 20Min.	59.60	68.15	63.55	65.24	88.50	69.14	68.00	**40.88**