On-Policy Replay for Continual Supervised Fine-Tuning

Anonymous ACL submission

Abstract

Continual supervised fine-tuning (SFT) is the de facto recipe for adapting large language models (LLMs) to a stream of downstream tasks, but it suffers from catastrophic forgetting of earlier capabilities. Recent work shows that on-policy signals—training on the model's own outputs—reduce forgetting more reliably than off-policy supervision. Existing on-policy methods route this signal through a new *training objective* (e.g., self-distillation losses with a teacher copy), inheriting an extra forward pass, schedule sensitivity, and stylistic drift from the teacher. We instead route the on-policy signal through the *training data source*. Our method, **On-Policy Replay** (OPR), rolls out the most recent checkpoint on a small budget of historical prompts, filters the generations by a task reward, and replays the surviving (prompt, model response) pairs as ordinary SFT examples. There is no teacher, no auxiliary loss, and no on-the-fly distillation. Across three 7–8B instruction-tuned backbones (Qwen2.5-7B-Instruct, Qwen3-8B, Llama3.1-8B-Instruct) on the TRACE continual-learning benchmark, OPR consistently reduces forgetting; on the sharpest stress test (Qwen2.5-7B-Instruct, Sequential SFT BWT -13.93), OPR lifts BWT to -0.65 at a 10% replay budget and to -2.29 at a 1% budget—a 46% reduction in $|\mathrm{BWT}|$ over a tuned Vanilla Replay baseline, with 42–46% reductions observed across all three backbones. We give a KL-shrinkage interpretation that places OPR and prior on-policy distillation methods on a single axis, and we present a counterintuitive finding that explains why Vanilla Replay is already a strong baseline: *low-score* replay is uniformly worse than Vanilla Replay, demonstrating that the active ingredient in OPR is the on-policy distribution, not the response quality alone.

1 Introduction

Modern large language models (LLMs) are rarely deployed as one-shot artifacts. A single instruction-tuned checkpoint is repeatedly adapted to new domains, formats, and skills as they emerge after deployment (Chung et al., 2022; Ouyang et al., 2022; Wu et al., 2024), and the default tool for each such update is supervised fine-tuning (SFT). Sequential SFT, however, suffers severe *catastrophic forgetting*: accuracy on previously learned tasks degrades by double-digit margins as new tasks are fine-tuned in sequence (Luo et al., 2023; Wang et al., 2023b). Recent diagnoses converge on a single thread (Shenfeld et al., 2025; Chen et al., 2025; Chu et al., 2025; Zhu et al., 2026): SFT is *off-policy*, and the per-update KL between successive policies on historical prompts, $\mathbb{E}_{x\sim p_{\text{hist}}}\,\mathrm{KL}\big(\pi_{\theta_t}\,\|\,\pi_{\theta_{t-1}}\big)$, upper-bounds how much the model forgets at each step—training on samples from $\pi_{\theta_{t-1}}$ shrinks this KL by construction. The diagnosis sharpens "mitigate forgetting" into a concrete question: where in the pipeline should the on-policy signal be delivered?

Existing remedies all answer this question on the *objective* side. SDFT (Shenfeld et al., 2026), OPSD (Zhao et al., 2026), and GKD (Agarwal et al., 2024) add an explicit per-token KL term against a teacher copy of the model, paying a teacher forward pass on every gradient step and inheriting a schedule (EMA decay, KL coefficient) that competes with the SFT optimizer for tuning bandwidth. RazorSFT (Shenfeld et al., 2025) keeps plain SFT but replaces the current task's gold target with a self-sampled response that passes a verifier, avoiding the teacher at the price of being confined to the current task's targets. Either way, the on-policy signal lives in the loss or in the target it is computed against. We observe that the underlying constraint is not on the loss but on the *data distribution* the loss sees—and that every replay-based continual learner already has a piece of state with that property: the cross-task replay buffer. The buffer is the one object in the pipeline that carries data across task boundaries; in standard practice it is filled with stale, off-policy gold targets and contributes noth-

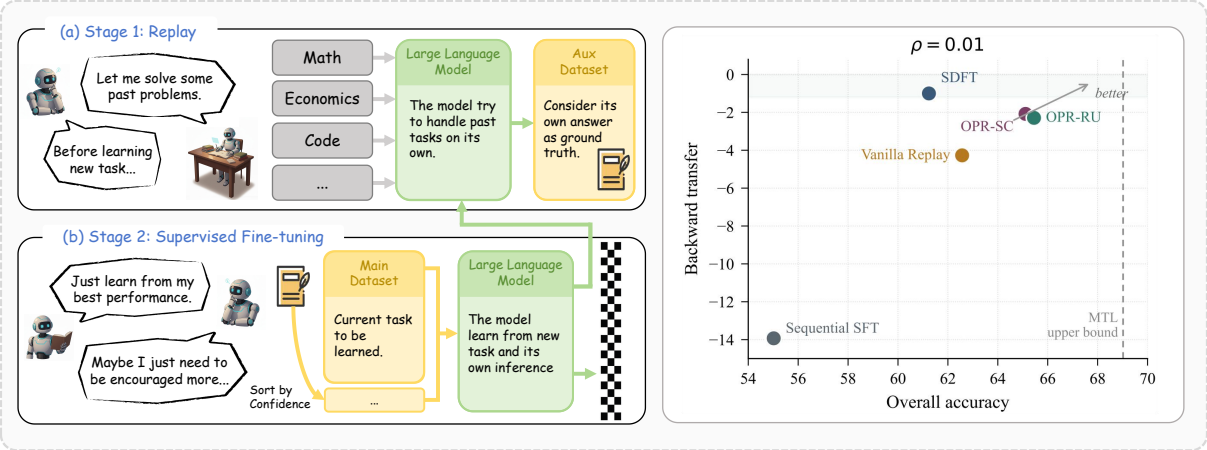

Figure 1: **On-Policy Replay (OPR) at a glance.** *Left:* after each task, OPR rolls out on past-task prompts, keeps the top-confidence (prompt, response) pairs as a replay buffer, and mixes it with the next-task data under plain SFT—no teacher, no auxiliary loss. *Right:* BWT vs. overall accuracy on TRACE/Qwen2.5-7B-Instruct at ρ=0.01; OPR-RU and OPR-SC move the frontier toward the MTL upper bound.

ing to the on-policy goal. Making the buffer itself on-policy turns out to be the cheapest way to enforce the KL constraint, and it leaves the training loop untouched.

We introduce **On-Policy Replay** (OPR): after each task, OPR rolls out the current checkpoint on a small budget of historical prompts, scores each rollout, and keeps only the top-quantile (prompt, response) pairs as the buffer for the next stage of plain cross-entropy SFT—no teacher, no auxiliary loss, no current-target rewrite (Figure 2). The default variant, OPR-RU (*rule-based*), scores rollouts with each task's own evaluation metric; because each replayed pair has high $\pi_{\theta_{t-1}}(\tilde{y} \mid x)$ by construction, the SFT gradient on the replay portion is small and points back toward $\pi_{\theta_{t-1}}$, so KL shrinkage emerges as a side effect of an unchanged training loop. We additionally introduce a label-free variant, OPR-SC (*self-confidence*), which scores rollouts by the model's own length-normalized log-probability—returned by the sampler at no additional cost—removing the dependence on task labels for buffer construction.

Across three 7–8B instruction-tuned backbones (Qwen2.5-7B-Instruct (Qwen Team, 2025), Qwen3-8B (Yang et al., 2025), Llama3.1-8B-Instruct (Grattafiori et al., 2024)), OPR consistently reduces backward transfer. On the sharpest stress test, Qwen2.5-7B-Instruct (Sequential SFT BWT -13.93), OPR-RU lifts BWT to -2.29 at a 1% replay budget—a 46% cut in $|\text{BWT}|$ over a tuned Vanilla Replay baseline—and beats SDFT by 4.2 overall-accuracy points without a teacher forward pass. A falsifying ablation supports the KL-shrinkage account: replacing the high-reward rollouts in the buffer with low-reward ones from the same pool drives BWT *below* Vanilla Replay, ruling out "replay-of-anything" explanations.

Contributions.

- **OPR**, a continual-SFT method that enforces on-policy KL shrinkage through the cross-task replay buffer rather than through an auxiliary loss term or current-target rewrite, leaving the SFT training loop untouched.
- **Label-free scoring via OPR-SC.** Replacing the rule-based scorer with the model's own length-normalized log-probability of the rollout removes the dependence on ground-truth labels for buffer construction, making continual SFT applicable in settings where reliable task annotations are unavailable.

2 Related Work

On-policy continual fine-tuning. A growing line of work argues that the gradient signal in continual SFT should come from data that lies on the model's own output distribution (Chen et al., 2025; Chu et al., 2025; Zhu et al., 2026), since on-policy data minimizes the per-update KL that bounds forgetting (Shenfeld et al., 2025). Existing realizations of this principle deliver the signal on the objective side: SDFT (Shenfeld et al., 2026), OPSD (Zhao et al., 2026), and GKD (Agarwal et al., 2024) add a per-token KL term against a teacher copy of the model on every gradient step; RazorSFT (Shenfeld

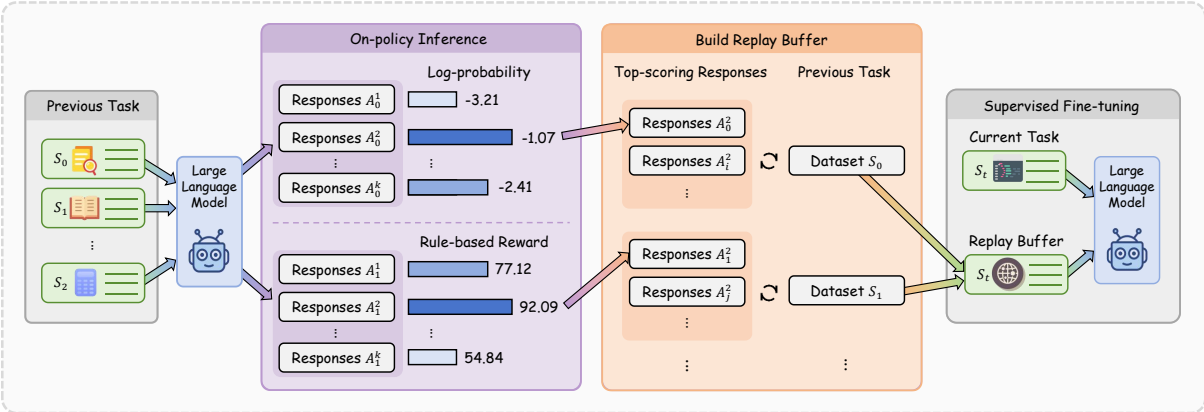

Figure 2: **OPR pipeline.** After training on task T_i (dataset S_i in the figure), the checkpoint π_{θ_i} rolls out on historical prompts; a scorer (rule reward or length-normalized log-likelihood) ranks the responses, the top-scoring (prompt, response) pairs form the replay buffer, and the buffer is mixed with task-T_{i+1} data for plain supervised fine-tuning.

et al., 2025) keeps plain SFT but rewrites the current task's gold target with a self-sampled, verifier-passing response. We place a detailed mechanistic comparison in §3.2; OPR differs from these methods in that it routes the on-policy signal through the cross-task replay buffer, leaving the SFT loss and target untouched.

Replay-based continual learning for LLMs. Among classical anti-forgetting families—regularization (Kirkpatrick et al., 2017), parameter isolation (Wang et al., 2023a; Ren et al., 2024), and rehearsal (Lopez-Paz and Ranzato, 2017)—rehearsal scales most gracefully at LLM scale (Wu et al., 2024; Luo et al., 2023), and TRACE (Wang et al., 2023b) consolidates eight downstream tasks as a standard testbed. Prior LLM-targeted replay methods refine *which prompts* to anchor on: Vanilla Replay mixes prior tasks' gold data into the current batch (Wu et al., 2024), GeRe (Zhang et al., 2025) anchors pretraining-time capability with general samples, I-LoRA (Ren et al., 2024) replays through interpolated LoRA adapters, the *spurious forgetting* hypothesis (Zheng et al., 2025) shows that ten anchor instances suffice to recover format alignment, and SuRe (Hazard et al., 2025) selects surprise-prioritised prompts. We refine the orthogonal axis—*which targets* to attach to historical prompts—replacing stale gold answers with reward-filtered on-policy rollouts; OPR-SC additionally removes the dependence on gold labels by scoring rollouts under the model's own length-normalized log-probability.

3 Method

3.1 Preliminaries

A continual SFT learner observes a sequence of tasks $T_1, \ldots, T_N$, each with a labeled dataset $\mathcal{D}_i = \{(x_{ij}, y_{ij})\}_{j=1}^{n_i}$, and at stage i fine-tunes $\pi_{\theta_{i-1}}$ on $\mathcal{D}_i$ together with an optional replay buffer $\mathcal{B}_i$ of size $|\mathcal{B}_i| \leq \rho \cdot n_i$. Let $a_{i,j}$ denote the accuracy of π_{θ_j} on task T_i. Backward transfer (Lopez-Paz and Ranzato, 2017) is $\mathrm{BWT} = \frac{1}{N-1}\sum_{i<N}(a_{i,N} - a_{i,i})$, and overall accuracy is the mean of $a_{i,N}$. A negative BWT measures forgetting.

3.2 Forgetting as Per-Update KL on Historical Prompts

We begin from the bound established by Shenfeld et al. (2025), which converts "mitigate forgetting" into a concrete scalar minimisation problem. For any held-out historical task T_i with prompt distribution p_i and any sufficiently small SFT update from θ_{t-1} to θ_t during stage i, the cross-entropy loss change on T_i satisfies, to leading order,

$$\Delta_i(\theta_{t-1} \to \theta_t) \;\leq\; C \cdot \mathbb{E}_{x\sim p_i}\big[\mathrm{KL}(\pi_{\theta_t} \,\|\, \pi_{\theta_{t-1}})\big] + o(\|\theta_t - \theta_{t-1}\|^2), \qquad (1)$$

for a constant $C > 0$ that depends on the smoothness of the loss. Aggregating over $j < i$ yields the same bound on the historical pool $p_{\text{hist}} = \frac{1}{i-1}\sum_{j<i} p_j$. Forgetting at stage i is therefore upper-bounded by a single scalar—the *per-update KL* between consecutive policies, evaluated on historical prompts—and minimising this scalar minimises the leading-order forgetting term.

The bound is mechanism-agnostic—any algorithm that keeps the per-update KL small mitigates

Algorithm 1 On-Policy Replay (one stage)

Require: Checkpoint π_{θ_i}; budget b; historical prompts $P_{\leq i}$; scorer r.

$\mathcal{R} \leftarrow \emptyset$
for each $x \in P_{\leq i}$ **do**
  $y \sim \pi_{\theta_i}(\cdot \mid x)$; $\mathcal{R} \leftarrow \mathcal{R} \cup \{(x, y, r(x, y))\}$
end for
Sort $\mathcal{R}$ by score; $\mathcal{B}_{i+1} \leftarrow$ top b entries.
Stage $i{+}1$: train π_{θ_i} on $\mathcal{D}_{i+1} \cup \mathcal{B}_{i+1}$ with cross-entropy SFT.
return $\pi_{\theta_{i+1}}$

forgetting. We show in §3.4 that OPR achieves this as a property of the training data rather than through an explicit loss term: on every replay sample the cross-entropy loss is small at θ_{t-1} by construction, and the expected SFT gradient over those samples descends $\mathrm{KL}(q_{t-1} \,\|\, \pi_\theta)$, pulling π_θ back toward the previous checkpoint.

3.3 On-Policy Replay

OPR constructs $\mathcal{B}_{i+1}$ from self-generated rollouts of the most recent checkpoint π_{θ_i} on historical prompts (Algorithm 1, Figure 2). At the end of stage i, π_{θ_i} samples one response per prompt $x \in P_{\leq i}$ under low-temperature decoding (sampling details in Appendix C); a scorer r ranks the resulting (prompt, response) pairs; the top-b entries, with $b = \rho\, n_{i+1}$ distributed equally across the i prior tasks (b/i entries each, with the remainder rounded forward), form $\mathcal{B}_{i+1}$, which is concatenated with $\mathcal{D}_{i+1}$ and trained under the standard token-level cross-entropy. The rollout is amortised once per stage and dominated by training compute in our setting.

3.4 KL-Shrinkage Analysis

We now show that the replay portion of every OPR batch satisfies Eq. (1) *by construction*. Throughout this subsection, i indexes the continual-learning stage and t indexes SGD steps within stage $i+1$; the buffer $\mathcal{B}_{i+1}$ is constructed once at the end of stage i using π_{θ_i}, so $\theta_{t-1} = \theta_i$ at the first step of stage $i+1$. Let a buffer item $(x, \tilde{y}) \in \mathcal{B}_{i+1}$ be drawn from the high-reward, on-policy distribution

$$q_{t-1}(y \mid x) \;=\; \frac{\pi_{\theta_{t-1}}(y \mid x)\, \mathbf{1}[r(x,y) \geq \tau]}{Z_{t-1}(x)}, \quad (2)$$

where τ is the per-task top-quantile threshold induced by Algorithm 1 and $Z_{t-1}(x)$ is the normaliser. Denote the per-sample cross-entropy loss by $\ell(\theta; x, \tilde{y}) = -\log \pi_\theta(\tilde{y} \mid x)$.

The filter in Algorithm 1 acts directly on the loss at θ_{t-1}. For the self-confidence scorer r_{SC} (§3.5, Eq. (7)), the chain rule yields the identity $r_{\mathrm{SC}}(x, \tilde{y}) = -\ell(\theta_{t-1}; x, \tilde{y})/|\tilde{y}|$, so top-$b$ selection at the empirical b-quantile τ is equivalent to a per-sample upper bound on the loss at θ_{t-1}:

$$\ell(\theta_{t-1}; x, \tilde{y}) \;=\; -\log \pi_{\theta_{t-1}}(\tilde{y} \mid x) \;\leq\; \ell_{\max}, \quad (3)$$

where $\ell_{\max} := -|\tilde{y}|\,\tau$; every buffer item sits in a high-likelihood region of $\pi_{\theta_{t-1}}$ by construction. Eq. (3) is tight at the first SGD step of stage $i+1$ (where $\theta_{t-1} = \theta_i$, the policy used to build $\mathcal{B}_{i+1}$) and loosens as $\|\theta_{t-1} - \theta_i\|$ grows, but refreshing the buffer at each stage boundary bounds this drift within a single stage. For the rule-based variant r_{RU} the identity no longer holds; we adopt the empirical assumption that r_{RU} is positively correlated with $\log \pi_{\theta_{t-1}}(\tilde{y} \mid x)$ on the rollout pool—task-correct rollouts tend to be high-likelihood ones—so that filtering by r_{RU} retains rollouts with per-sample loss bounded by an empirical $\ell_{\max}$.

The per-sample bound propagates to the mini-batch gradient. Under L-smoothness of ℓ, the self-bounding inequality $\|\nabla_\theta \ell(\theta_{t-1}; x, \tilde{y})\|^2 \leq 2L\,\ell(\theta_{t-1}; x, \tilde{y})$ combined with Jensen's $\|g\|^2 \leq \mathbb{E}_{q_{t-1}}[\|\nabla_\theta \ell\|^2]$ and Eq. (3) yields

$$\|g\|^2 \;\leq\; 2L\,\mathbb{E}_{q_{t-1}}[\ell] \;\leq\; 2L\,\ell_{\max}, \quad (4)$$

with $g := \mathbb{E}_{q_{t-1}}[\nabla_\theta \ell]$ the gradient on the replay portion of the batch. This gradient also points in a meaningful direction: because $\mathbb{E}_q[\nabla_\theta \log \pi_\theta(y \mid x)] = -\nabla_\theta \mathrm{KL}(q \,\|\, \pi_\theta)$ and the entropy of q is independent of θ,

$$\mathbb{E}_{q_{t-1}}\big[\nabla_\theta \ell\big] \;=\; \nabla_\theta\, \mathbb{E}_x\big[\mathrm{KL}(q_{t-1} \,\|\, \pi_\theta)\big], \quad (5)$$

so an SGD step against g decreases $\mathrm{KL}(q_{t-1} \,\|\, \pi_\theta)$ and pulls π_θ toward q_{t-1}—a restriction of the previous checkpoint to its high-reward region.

Combining these ingredients yields the forgetting bound. Under (a) non-negativity and L-smoothness of ℓ in θ on a neighbourhood of θ_{t-1}; (b) bounded spectral norm $\sigma_{\max}$ of the Fisher information $F(\theta_{t-1})$; and (c) step size η small enough that $O(\eta^3)$ terms are negligible, one SGD step of size η against the replay gradient g satisfies

$$\mathbb{E}_{p_{\mathrm{hist}}}\big[\mathrm{KL}(\pi_{\theta_t} \,\|\, \pi_{\theta_{t-1}})\big] \;\leq\; L\,\sigma_{\max}\,\eta^2\,\ell_{\max} + o(\eta^2), \quad (6)$$

where $\ell_{\max}$ is the per-sample loss bound at θ_{t-1} enforced by the high-score filter (Eq. (3)). The filter therefore directly controls the leading-order forgetting term. The bound follows from Taylor-expanding $\mathrm{KL}(\pi_{\theta_t} \,\|\, \pi_{\theta_{t-1}})$ in θ_t around $\theta_t = \theta_{t-1}$, with displacement $\Delta := \theta_t - \theta_{t-1} = -\eta g$. The zeroth-order term is $\mathrm{KL}(\pi_{\theta_{t-1}} \,\|\, \pi_{\theta_{t-1}}) = 0$, and the first-order term vanishes because $\theta_t = \theta_{t-1}$ is the unique minimiser of $\mathrm{KL}(\pi_{\theta_t} \,\|\, \pi_{\theta_{t-1}})$ in θ_t. The leading non-zero term is the quadratic

$$\begin{aligned}\tfrac{1}{2}\,\Delta^\top F(\theta_{t-1})\,\Delta &= \tfrac{1}{2}\,\eta^2\, g^\top F(\theta_{t-1})\, g \\ &\le \tfrac{1}{2}\,\sigma_{\max}\,\eta^2\,\|g\|^2,\end{aligned}$$

where the Hessian of $\mathrm{KL}(\pi_{\theta_t} \| \pi_{\theta_{t-1}})$ in θ_t at $\theta_t = \theta_{t-1}$ coincides with the Fisher information $F(\theta_{t-1}) = \mathbb{E}\big[\nabla \log \pi_{\theta_{t-1}} \nabla \log \pi_{\theta_{t-1}}^\top\big]$. Substituting Eq. (4) for $\|g\|^2$ yields Eq. (6).

The bound in Eq. (6) rests on *both* ingredients of q_{t-1}, which makes a sharp falsifying prediction. Removing high-reward selection while preserving on-policy sampling—i.e., replacing the indicator $\mathbf{1}[r \ge \tau]$ in Eq. (2) with $\mathbf{1}[r \le \tau]$—keeps Eq. (5) intact but breaks Eq. (3): low-score rollouts lie in low-likelihood regions of $\pi_{\theta_{t-1}}$, so $\ell_{\max}$ becomes large and Eq. (6) becomes loose. The theory therefore predicts a degraded BWT relative to high-score OPR, and predicts that the resulting buffer can be *worse* than naive off-policy replay—since the latter at least uses correct gold targets. We verify both predictions in §4.3.

Loss-side methods such as SDFT minimise an upper bound on Eq. (1) by adding an explicit term $\beta \cdot \mathrm{KL}(\pi_\theta \,\|\, \pi_\phi)$ to the training loss, where π_ϕ is a teacher copy of the model. The two routes share the surrogate but encode it differently: as a per-step term in the loss vs. as a property of the replay data distribution. The data route is intrinsically *one-sided*—Eq. (6) bounds the per-update KL only on the replay portion of each batch, leaving the new-task gradient unconstrained.

3.5 Choice of Scorer

The scorer r in Algorithm 1 is the only component that requires task-level knowledge. We consider two instantiations.

Rule-based (OPR-RU). The default variant scores each rollout with the task's own evaluation metric: exact-match for classification (C-STANCE, FOMC, NumGLUE-cm/ds, ScienceQA), token-level fuzz similarity for Py150, ROUGE-L F1 for MeetingBank, SARI for 20Minuten. Because the same scorer is used to filter the buffer and to evaluate the model, the buffer's filter is by construction aligned with the metric the model is judged on.

Self-confidence (OPR-SC). Eq. (3) in the derivation above only requires that $\tilde{y}$ lie in a high-likelihood region of $\pi_{\theta_{t-1}}$; the model's own length-normalised log-probability is exactly that quantity:

$$r_{\mathrm{SC}}(x, y) \;=\; \frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi_{\theta_i}(y_t \mid x, y_{<t}). \qquad (7)$$

Two consequences. First, r_{SC} is a free by-product of the rollout: the log-probabilities are returned by the sampler at no additional cost. Second, no ground-truth answer, external scorer, or teacher enters the buffer-construction loop—making OPR-SC applicable in production settings where reliable task annotations are unavailable. §4.3 reports its head-to-head gap to OPR-RU.

4 Experiments

4.1 Setup

Benchmark. We conduct our evaluation on TRACE (Wang et al., 2023b), a widely adopted continual learning benchmark for LLMs. TRACE comprises eight diverse tasks arranged in a fixed canonical order: C-STANCE (*C-STA*; stance classification), FOMC (Federal Open Market Committee classification), MeetingBank (*MeBa*; meeting summarization), Py150 (Python code completion), ScienceQA (*SciQA*; multi-step science QA), NumGLUE-cm and NumGLUE-ds (*NG-cm*, *NG-ds*; numerical reasoning), and 20Minuten (*20Min*; German news simplification). For each task, we subsample 5,000 training examples and adopt the official evaluation set.

Models. We report the main results on Qwen2.5-7B-Instruct, Qwen3-8B, and Llama3.1-8B-Instruct. All three are instruction-tuned checkpoints obtained through RLHF, and we regard the preservation of instruction-following ability across the eight-stage sequence as the practically meaningful objective.

Training. All methods use AdamW (learning rate 1×10^{-5}, cosine schedule, no warmup, weight decay 0) with a global batch size of 128, max sequence length 2048, and bf16 precision. We run all experiments with three different random seeds and report the averaged results. Per-task epochs follow

Method	ACC	BWT	C-STA	FOMC	MeBa	Py150	SciQA	NG-cm	NG-ds	20Min
Qwen2.5-7B-Instruct										
SFT	55.02	−13.93	46.23	28.41	59.23	63.09	85.74	61.64	77.15	39.93
Replay	62.57	−4.27	53.51	69.91	59.74	64.78	**93.42**	65.03	77.08	40.72
SDFT	61.23	**−1.00**	54.45	69.43	27.78	58.30	92.80	**76.38**	69.19	41.35
OPR-RU	**65.45**	−2.29	**54.65**	**72.65**	**63.76**	64.07	86.56	67.67	**77.56**	**41.89**
OPR-SC	65.11	−2.08	51.86	70.69	63.60	**66.85**	91.68	59.57	75.31	41.33
Qwen3-8B										
SFT	58.25	−10.85	46.62	28.81	59.63	63.49	86.14	62.04	77.96	41.33
Replay	65.89	−3.37	53.51	70.31	60.14	65.18	93.82	65.43	77.58	41.12
SDFT	64.56	**−1.20**	54.65	69.83	37.18	58.70	93.20	**76.78**	69.69	41.75
OPR-RU	**68.10**	−1.97	**56.53**	**74.73**	65.84	66.15	88.64	69.75	**79.23**	42.97
OPR-SC	67.81	−1.77	54.44	73.27	**66.18**	**69.43**	**94.26**	62.15	77.89	**43.91**
Llama3.1-8B-Instruct										
SFT	50.78	−17.07	23.26	43.69	60.59	64.27	72.83	28.09	72.15	41.32
Replay	58.33	−5.36	37.29	69.00	61.18	64.22	82.83	39.82	71.15	41.21
SDFT	57.01	**−1.34**	**38.83**	70.11	30.51	59.32	80.49	41.15	70.84	42.72
OPR-RU	**61.89**	−2.89	36.90	**71.62**	**62.53**	64.26	82.14	**41.61**	**72.26**	**42.82**
OPR-SC	61.52	−2.67	35.53	70.91	62.36	**65.77**	**84.50**	39.27	71.70	42.43

Table 1: **Main results on TRACE at ρ=0.01.** Overall accuracy (ACC, %), backward transfer (BWT, pp), and per-task accuracy. **Bold**: best within backbone; underline: second-best.

$[5, 3, 7, 5, 3, 5, 5, 7]$ (chosen to roughly equalize the number of gradient steps across tasks). The same hyperparameters are applied to every method and every backbone, isolating the replay-data source as the only varying factor. See Appendix B for the full infrastructure setup.

Methods compared. **Sequential SFT** (*SFT* in Table 1) fine-tunes all model parameters on tasks one after another with no replay. **Multi-Task Learning** (*MTL* in Figure 1) jointly trains on all eight task datasets in a single run; it has access to data that the continual learner does not and serves only as an upper bound. **Vanilla Replay** (*Replay* in Table 1) is a standard replay-based sequential fine-tuning baseline: at each stage it mixes a ρ fraction of each prior task's gold training data into the current task's batch. **Self-Distillation Fine-Tuning** (*SDFT* in Table 1) (Shenfeld et al., 2026) augments the SFT objective with an explicit per-token KL term against a teacher copy of the model, anchoring the student's output distribution to the teacher on every gradient step. **OPR-RU (ours)** is Algorithm 1 with task-specific rule-based scoring (§3.5). **OPR-SC (ours)** replaces the rule-based scorer with the length-normalized log-probability under π_{θ_i} (Eq. (7)).

Metrics. We report two metrics. The overall accuracy (*ACC* in Table 1) is the mean of the per-task accuracies $a_{i,N}$ at the end of the full eight-stage sequence, and reflects final capability. Backward transfer (*BWT* in Table 1) is the average per-task accuracy change between end-of-task and end-of-sequence, $\mathrm{BWT} = \frac{1}{N-1}\sum_{i<N}(a_{i,N} - a_{i,i})$; a more negative value indicates more forgetting.

4.2 Main Results

Table 1 summarizes performance at the strictest ρ=0.01 budget. The trends are stable across all three backbones.

Sequential SFT collapses on every backbone (BWT −10.85 to −17.07 pp), consistent with prior reports of forgetting at LLM scale (Wang et al., 2023b; Luo et al., 2023). A 1% Vanilla Replay budget recovers most of the degradation (Wu et al., 2024) but leaves a 3–5 pp BWT gap.

OPR-RU consistently improves over Vanilla Replay. It cuts $|\mathrm{BWT}|$ by 42–46% across the three backbones at ρ=0.01, and the pattern holds across all budgets we test (Table 2).

OPR approaches SDFT without a teacher. SDFT retains a marginal BWT edge (within ∼1.5 pp on every backbone) but loses 3–5 pp of overall accuracy, driven almost entirely by a collapse on MeetingBank (Figure 3). In long-form summarization the self-distilled teacher inherits its own hallucinations; lacking any external reward signal, the student inherits them too. We make this mechanistic in §4.4.

4.3 Ablations Analyses

We sweep three axes: the replay budget ρ, a falsifying low-score variant predicted by the falsifying argument in §3.4, and the choice of scorer.

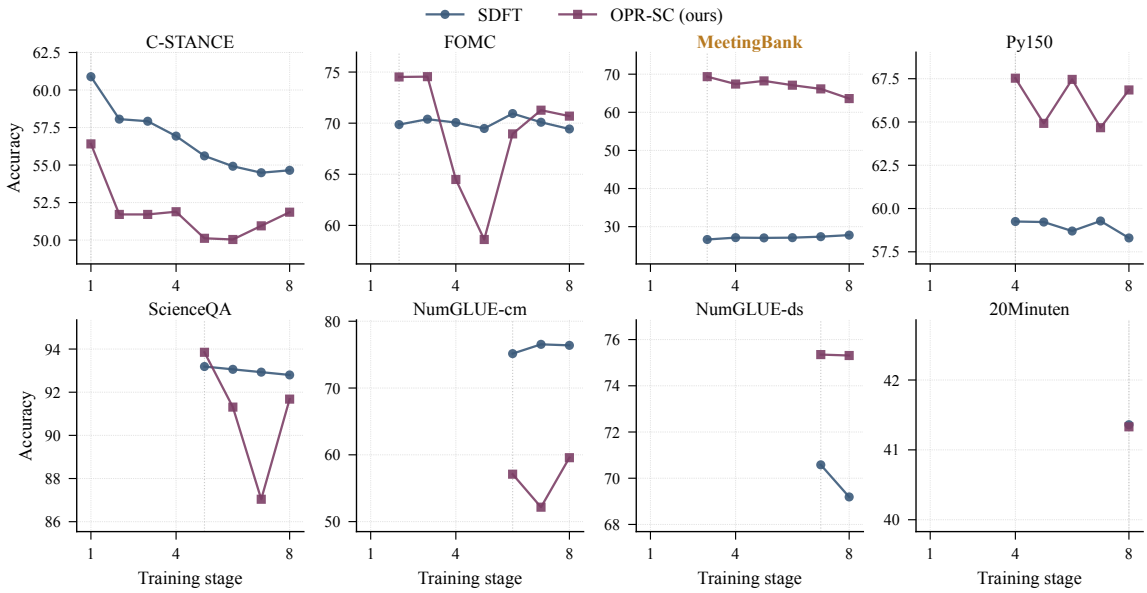

Figure 3: **Per-task accuracy across the 8 TRACE stages, SDFT vs. OPR-SC at ρ=0.01 on Qwen2.5-7B-Instruct.** SDFT plateaus on MeetingBank; OPR-SC underperforms on NumGLUE-cm, where the model is confidently wrong.

Method	ρ	ACC ↑	BWT ↑
Vanilla Replay	0.01	62.57	−4.27
	0.02	63.68	−3.89
	0.05	64.50	−3.53
	0.10	66.08	−1.37
OPR-RU (ours)	0.01	65.45	−2.29
	0.02	65.59	−1.71
	0.05	65.78	−1.43
	0.10	**65.92**	**−0.65**

Table 2: **Budget sensitivity on Qwen2.5-7B-Instruct.** ACC and BWT for Vanilla Replay vs. OPR-RU across $\rho \in \{0.01, 0.02, 0.05, 0.10\}$.

Method (ρ=0.10)	Overall ↑	BWT ↑
Vanilla Replay	66.08	−1.37
OPR-RU (high-score)	**65.92**	**−0.65**
OPR-RU (low-score)	65.01	−3.47

Table 3: **Low-score ablation on Qwen2.5-7B-Instruct (ρ=0.10).** Replacing the high-reward rollouts in the OPR buffer with low-reward ones from the same pool (prompts, sampler, code held fixed).

Budget sensitivity. We sweep $\rho \in \{0.01, 0.02, 0.05, 0.10\}$ (Table 2). OPR-RU cuts $|\mathrm{BWT}|$ by 46–60% over Vanilla Replay at every budget, and its overall accuracy is nearly flat (range 0.47 pp) while Vanilla Replay swings by 3.51 pp—once the buffer is on-policy, replay *quality* dominates over replay *quantity*.

Low-score replay. To test the falsifying prediction in §3.4, we hold the prompt set, sampler, and code fixed, and swap the high-score rollouts for low-score ones from the same on-policy batch (Table 3, ρ=0.10). Low-score replay is both (i) much worse than high-score OPR-RU (BWT degrades 5.3×, from −0.65 to −3.47) and (ii) worse than off-policy Vanilla Replay (−3.47 vs. −1.37). On-policy sampling alone is insufficient—the high-reward filter is the decisive factor, exactly as predicted when $\pi_{\theta_{t-1}}(\tilde{y} \mid x)$ is small and the bound in Eq. (6) becomes loose.

Scorer choice. At ρ=0.01 on Qwen2.5-7B-Instruct, OPR-SC matches OPR-RU on aggregate (65.11 vs. 65.45 ACC; −2.08 vs. −2.29 BWT), confirming the design rationale of §3.5: the on-policy plus filter structure is the principal driver of the gains, not the specific scorer. The per-task breakdown in Figure 3 adds nuance: OPR-SC excels where the model is well-calibrated (FOMC, ScienceQA) but pays a cost on NumGLUE-cm, where fluently-stated incorrect rollouts pass the confidence filter—a known design boundary, since the buffer mechanism anchors what the model is already confident in rather than manufacturing new capability.

4.4 Loss Dynamics

Figure 4 plots per-stage training loss for SDFT and OPR-SC, sharing every hyperparameter except the

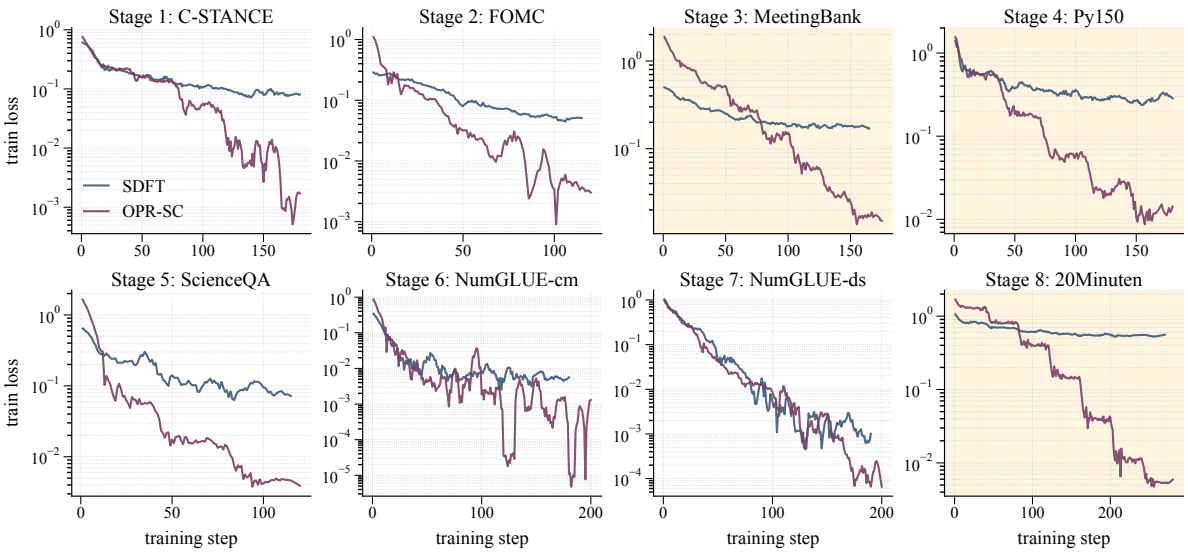

Figure 4: **Per-stage training loss, SDFT vs. OPR-SC at ρ=0.01 on Qwen2.5-7B-Instruct.** On MeetingBank (Stage 3), Py150 (Stage 4), and 20Minuten (Stage 8) SDFT's loss plateaus while OPR-SC's descends to the same level it reaches on the classification tasks.

supervision route. On four stages (C-STANCE, FOMC, ScienceQA, NumGLUE-ds) the curves are nearly indistinguishable. On three stages—MeetingBank, Py150, and 20Minuten—SDFT's loss plateaus an order of magnitude above OPR-SC's and never descends, while OPR-SC's loss drops to the same low neighbourhood it reaches elsewhere.

The three plateau tasks all demand large distributional shifts from the instruction-tuned base—summarization, code completion, German simplification—whereas the four matched tasks are classification, whose surface form (selecting from labeled options) sits close to the base distribution. SDFT's teacher reverse-KL anchor keeps the cross-entropy from descending precisely on the tasks that need it to descend most, as the bound in §3.4 predicts.

4.5 Per-Task Breakdown

Figure 5 decomposes Table 1 into per-task gains and losses relative to Vanilla Replay. SDFT exhibits a single substantial deficit on MeetingBank (-23 to -32 pp across backbones), plus smaller consistent deficits on Py150 and NG-ds. Neither OPR-RU nor OPR-SC produces a comparable failure mode—deviations stay within a few points of zero on most tasks. Consistent with §4.4: SDFT's anchor pulls the model away from the new-task optimum when a large distributional shift is required, whereas OPR preserves both retention and acquisition.

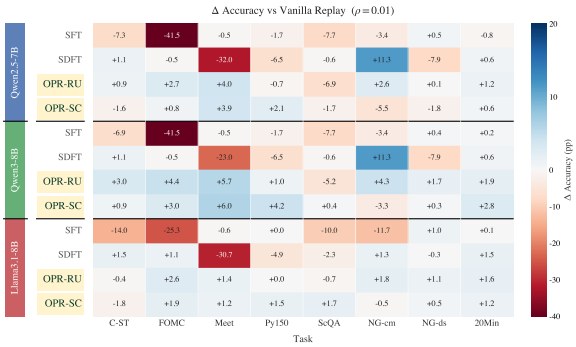

Figure 5: **Per-task Δ accuracy vs. Vanilla Replay at ρ=0.01.**

5 Conclusion

Forgetting in continual SFT is bounded by the per-update KL on historical prompts, and prior work has uniformly enforced that bound on the loss side—via a teacher network, an auxiliary KL term, or a rewritten target. We showed it can be enforced through the cross-task replay buffer alone: drawing the buffer from the previous checkpoint and filtering by reward upper-bounds the per-sample loss at θ_{t-1} and descends $\mathrm{KL}(q_{t-1} \,\|\, \pi_\theta)$ as an identity, with the SFT loop untouched. Loss-side and data-side methods become two encodings of the same surrogate, and a low-reward ablation falsifies any "replay-of-anything" account of the gains. The natural next step is to learn the scorer end-to-end, in place of handcrafting it from evaluation metrics or per-token likelihoods.

Limitations

Our claims rest on three 7–8B instruction-tuned backbones and TRACE's canonical eight-task English sequence; we have not validated the KL-shrinkage account on base checkpoints, on models above ∼10B parameters, on alternative task orderings, or on non-English continual streams. The headline numbers of OPR-RU also depend on a task-specific evaluator, and although the fallback OPR-SC removes that dependence and matches OPR-RU in aggregate, it fails on a predictable class of inputs—those on which the model is *confidently wrong*—where fluent but incorrect rollouts pass the self-confidence filter and degrade per-task accuracy.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. 2025. Retaining by doing: The role of on-policy data in mitigating forgetting. *arXiv preprint arXiv:2510.18874*.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

Hyung Won Chung, Le Hou, Shayne Longpre, and 1 others. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Aaron Grattafiori and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hugo Hazard, Zafeirios Fountas, Martin A. Benfeghoul, Adnan Oomerjee, Jun Wang, and Haitham Bou-Ammar. 2025. Sure: Surprise-driven prioritised replay for continual llm learning. *arXiv preprint arXiv:2511.22367*.

Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. MeetingBank: A benchmark dataset for meeting summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yun Luo, Zhicheng Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Long Ouyang, Jeff Wu, Xu Jiang, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Qwen Team. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Veselin Raychev, Pavol Bielik, and Martin Vechev. 2016. Probabilistic model for code with decision trees. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA)*.

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

Annette Rios, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. 2021. A new dataset and efficient baselines for document-level text simplification in German. In *Proceedings of the Third Workshop on New Frontiers in Summarization (NewSum@EMNLP)*.

Agam Shah, Suvan Paturi, and Sudheer Chava. 2023. Trillion dollar words: A new financial dataset, task & market analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. 2026. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. 2025. RI's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023a. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics (EMNLP Findings)*.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjia Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023b. Trace: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*.

Tongtong Wu, Linhai Luo, Yifan Wen, Liwei Zhao, Yuhao Zhang, Wei Chen, Liang Cui, Bing Liu, Bolin Ding, and Jingren Zhou. 2024. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364*.

An Yang and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yunan Zhang, Shuoran Jiang, Mengchen Zhao, Yuefeng Li, Yang Fan, Xiangping Wu, and Qingcai Chen. 2025. Gere: Towards efficient anti-forgetting in continual learning of llm via general samples replay. *arXiv preprint arXiv:2508.04676*.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. 2023. C-STANCE: A large dataset for Chinese zero-shot stance detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. 2026. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. 2025. Spurious forgetting in continual learning of language models. In *International Conference on Learning Representations (ICLR)*.

Taojie Zhu, Dongyang Xu, Ding Zou, Sen Zhao, Qiaobo Hao, Zhiguo Yang, and Yonghong He. 2026. Bridging sft and rl: Dynamic policy optimization for robust reasoning. *arXiv preprint arXiv:2604.08926*.

A TRACE Tasks, Templates, and Metrics

This section documents the eight TRACE tasks at the level of detail needed to reproduce our results. It complements the high-level description in §4.1. The original sources of the eight tasks are C-STANCE (Zhao et al., 2023), FOMC (Shah et al., 2023), MeetingBank (Hu et al., 2023), Py150 (Raychev et al., 2016), ScienceQA (Lu et al., 2022), NumGLUE (Mishra et al., 2022) (from which TRACE's NumGLUE-cm and NumGLUE-ds splits are taken), and 20Minuten (Rios et al., 2021).

A.1 Task Summary

Two properties of this composition matter for our experiments. First, average input length varies by almost two orders of magnitude, from 39 tokens (NumGLUE-ds) to 3,550 tokens (MeetingBank), so a per-task replay budget of even $\rho = 1\%$ corresponds to very different token counts per task. Second, the metric set is heterogeneous: ROUGE-L and SARI are bounded surface-form metrics, edit similarity is character-level, and accuracy is binary, which means the per-task scores in Table 4 are not directly commensurable. This is the same heterogeneity we flag in the Limitations section and is the reason OPR-RU averages a normalized scalar reward across tasks rather than the raw metric.

A.2 Input/Output Examples

To make the input/output conventions of each TRACE task fully unambiguous, we provide one representative *(prompt, gold response)* pair per task below. Each example is reproduced *verbatim* in the same chat-templated form that is fed to the model at both training and evaluation time, including instruction wording, answer-format constraints, and any in-context formatting (e.g., option letters for multiple-choice tasks). The examples follow the same order as Table 4: C-STANCE, FOMC, MeetingBank, Py150, ScienceQA, NumGLUE-cm, NumGLUE-ds, and 20Minuten. Note that C-STANCE is natively a Chinese dataset; the example shown below has been translated into English for presentation only, while the actual training and evaluation are performed on the original Chinese text. For readability, long contexts (e.g., the MeetingBank transcripts and 20Minuten articles) are shown in full without truncation, so that the average-length statistics in Table 4 can be cross-checked against the samples.

Example for C-STANCE

Query: Determine the stance of the following text toward the given target. Choose one of: A. Support, B. Oppose, C. Neutral. Output only A, B, or C.
Text: Starting bilingual language training for kids from an early age is actually feasible. They can at least handle daily conversations with ease. However, sufficient language input and output must be guaranteed. For example, one parent speaks one language while the other uses the second, and children should respond in the corresponding language. Otherwise, they will end up understanding the

Dataset	Source	Avg len	Metric	Language	Size
C-STANCE	Social media	109	Accuracy	Chinese	5000
FOMC	Finance	76	Accuracy	English	5000
MeetingBank	Meeting	3550	ROUGE-L	English	5000
Py150	Github	814	Edim similarity	English	5000
ScienceQA	Science	285	Accuracy	English	5000
NumGLUE-cm	Math	50	Accuracy	English	5000
NumGLUE-ds	Math	39	Accuracy	English	5000
20Minuten	News	739	SARI	German	5000

Table 4: An overview of the dataset statistics in TRACE. "Source" denotes the origin of the context, "Avg len" indicates the average number of tokens per dataset, and "SARI" refers to a metric specific to text simplification.

language but being unable to speak it.
Target: bilingual parenting for early language training
Answer: A

Example for FOMC

Query: What is the monetary policy stance for the following text? A. dovish, B. hawkish, C. neutral. Choose one from A, B and C.
Text: First, tightening will in all likelihood occur in the context of a more firmly established economic recovery in the United States so that any adverse effects on EME financial conditions should be buffered by the beneficial effects of higher external demand.
Answer: B

Example for MeetingBank

Query: Write a summary of the following meeting transcripts.
Meeting transcripts:Councilmember Clark, would you please put resolution 2a1 on the floor for adoption.As council president of the council, resolution 201 be adopted.Thank you. We've got a mover in a second. Comments and questions by members of Council on Resolution 201. Councilmember CdeBaca.Thank you. I just want to go on record voting no for this tonight. Pepsi is not necessarily a group that needs $1,000,000 incentive to stay in Denver, and they are well on their way to completing their move with or without this incentive.So definitely just want to make sure that I'm clear in not supporting this tonight. Thank you.Thank you. And I'll go ahead. And since this is in my council district, District 11, we as community members, along with some of the original steering committee members from the far northeast neighborhood plan, reconvened and talked through all of the pros and possible challenges of welcoming a large scale manufacturer into District 11.And the community saw fit that this was a great use for this area and this will bring close to 500 jobs to the region as well. And I would ask my colleagues to support it tonight. Madam Secretary, roll call on Council Resolution 22, Dash 201, please.See tobacco? No. Clark.I.Swen.I. Herndon, I.Hi. All right. Cashman. Kenny Ortega. Sandoval.I.Sawyer. I. Torres. I. Black. Madam President, I.Madam Secretary, please close the voting and announce the results.One day, 11 eyes.11 Eyes. Council Resolution 22, dash 201 has been adopted. Madam Secretary, please put the next item up on our screens. Councilmember Flynn, go ahead with your comments on resolution 159, please.
Answer: A resolution approving a proposed Business Incentive Fund (BIF) contract with Bottling Group, LLC for an amount not to exceed $1,000,000 for 7 years and for performance based criteria that include establishment of a state-of-the-art manufacturing facility and associated significant investments, as well as job retention and creation and associated investments into upskilling and training of the manufacturing facility's Denver-based workforce.Approves a Business Incentive Fund (BIF) contract with Bottling Group, LLC for an amount not to exceed $1,000,000 and for 7 years through 12-31-2028 for performance based criteria that include establishment of a state-of-the-art manufacturing facility and associated significant investments, as well as job retention and creation and associated investments into upskilling and training of the manufacturing facility's Denver-based workforce (OEDEV-202160849-00). The last regularly scheduled Council meeting within the 30-day review period is on 4-11-2022. The Committee approved filing this item at its meeting on 3-9-2022.

Example for Py150

Query: code
Answer: completion

Example for ScienceQA

Query: Choose an answer for the following question and give your reasons.
Question: What information supports the conclusion that Trent acquired this trait?
Hint: Read the description of a trait. Trent knows how to fly a kite.
Choices: A. Trent's neighbor taught him how to repair a kite. B. Trent likes to fly a kite with his younger brother. C. Trent's friend taught him how to fly a kite.
Answer: C. Organisms, including people, have both inherited and acquired traits.
Inherited and acquired traits are gained in different ways. Inherited traits are passed down from biological parents to their offspring through genes. Genes are pieces of hereditary material that contain the instructions that affect inherited traits. Offspring receive their genes, and therefore gain their inherited traits, from their biological parents. Inherited traits do not need to be learned.
Acquired traits are gained during a person's life. Some acquired traits, such as riding a bicycle, are gained by learning. Other acquired traits, such as scars, are caused by the environment. Parents do not pass acquired traits down to their offspring.

Example for NumGLUE-cm

Query: Solve the following math problem.
Question: In a car parking lot, Greyson observed that there are 48 car wheels. Find out the number of cars in the parking lot.
Answer: 12

Example for NumGLUE-ds

Query: Solve the following math problem.
Question: How many moles of Ammonium iodide are required to react with 3 moles of Potassium hydroxide to form 3 moles of Ammonia, 3 moles of Potassium iodide and 3 moles of Water.
Answer: 3

Example for 20Minuten

Query: Provide a simplified version of the following paragraph in German.
Paragraph: In Lyss (Bern), police raided an illegal gambling den and arrested two women aged 30 and 41. Around 40 people were checked on Friday evening on suspicion of illegal gambling and betting. According to the Bernese Jura-Seeland Regional Public Prosecutor's Office and Bern Cantonal Police, the two women are believed to have worked in the premises without a valid work permit and will be charged with violations of the Foreign Nationals and Integration Act. Authorities confirmed that prohibited gambling and betting services were offered at the site. During the inspection and a subsequent house search, officers seized various electronic devices, a slot machine, tens of thousands of Swiss francs in cash and suspected gold items. The operation followed extensive prior investigations. Public prosecutors are still looking into the operators of the premises. Illegal gambling and sports betting raids have taken place in Lyss multiple times before. Back in 2018, 25 people were checked at another venue, with cash, betting computers and slot machines confiscated. Similar operations were also carried out in 2016 and 2015.
Answer: In Lyss, the Bern Cantonal Police conducted a targeted inspection. Cash was seized and two women were arrested during the operation. Similar inspections had already taken place in previous years.

B Training Details

All runs use the `swift sft` entry point of ms-swift with full-parameter fine-tuning, on 8×H100 (80 GB) GPUs under DeepSpeed ZeRO-2 and Liger Kernel, in bf16. We apply AdamW with learning rate 1×10^{-5}, cosine schedule, no warmup, weight decay 0, per-device batch size 16 (global 128), and max sequence length 2048. Per-task epochs follow $[5, 3, 7, 5, 3, 5, 5, 7]$. Only the final checkpoint of each stage is retained and used as the initialization of the next stage. The same hyperparameters are applied to every method and every backbone, so that the replay-data source is the only varying factor.The total training cost for a single model across all 8 datasets amounts to 32 GPU-hours.

C Rollout for OPR

Between consecutive stages, the most recent checkpoint π_{θ_i} generates one response per historical prompt using **vLLM**. The tensor parallel size is 4, sampling temperature $\tau = 0.1$, and nucleus sampling parameter top-p at its default 1, with a maximum of 512 newly generated tokens. The rollout runs as a separate subprocess from the training step, which uses 8-way data parallelism, so the rollout GPUs need not hold the optimizer state.

For each prior task T_j, the prompts are identical to the 5,000 training prompts used at stage j, with the gold labels *discarded*. Prompts whose chat-templated length exceeds 2048 tokens are removed by an overlong filter, and the same filter is consistently applied during both training and evaluation.

Given a global budget $b = \rho \cdot 5000$, the buffer at stage $i{+}1$ is divided among the i prior tasks as evenly as possible, subject to the constraint that the allocation for any single task does not exceed the size of its corresponding source buffer.

D Detailed Experiment Results

In this section, we report a portion of the detailed experimental results from our paper. The model employed is Qwen2.5-7B-Instruct, and the corresponding results are presented in the following table.

Task ID	C-STA	FOMC	MeBa	Py150	SciQA	NG-cm	NG-ds	20Min
1	55.89							
2	54.37	73.69						
3	49.88	65.80	68.30					
4	47.70	33.72	42.26	66.80				
5	17.40	0.00	51.93	62.83	95.09			
6	50.74	65.78	47.89	66.27	92.40	62.96		
7	50.95	67.09	44.98	63.19	80.41	63.58	77.92	
8	46.23	28.41	59.23	63.09	85.74	61.64	77.15	39.93

Table 5: Detailed results of Qwen2.5-7B-Instruct under sequential SFT.

Task ID	C-STA	FOMC	MeBa	Py150	SciQA	NG-cm	NG-ds	20Min
1	55.82							
2	54.51	73.54						
3	50.93	69.76	69.04					
4	50.92	69.25	58.29	67.02				
5	51.36	65.50	61.92	65.10	94.99			
6	52.64	68.78	60.55	66.30	92.72	71.60		
7	52.74	68.20	60.09	66.28	92.12	65.28	77.54	
8	53.51	69.91	59.74	64.78	93.42	65.03	77.08	40.72

Table 6: Detailed results of Qwen2.5-7B-Instruct under vanilla replay with $\rho = 0.01$.

Task ID	C-STA	FOMC	MeBa	Py150	SciQA	NG-cm	NG-ds	20Min
1	60.89							
2	58.06	69.86						
3	57.92	70.39	26.63					
4	56.93	70.06	27.12	59.25				
5	55.61	69.48	27.05	59.22	93.19			
6	54.93	70.94	27.11	58.70	93.06	75.15		
7	54.49	70.09	27.37	59.28	92.93	76.54	70.58	
8	54.65	69.43	27.78	58.30	92.80	76.39	69.19	41.36

Table 7: Detailed results of Qwen2.5-7B-Instruct using SDFT

Task ID	C-STA	FOMC	MeBa	Py150	SciQA	NG-cm	NG-ds	20Min
1	55.79							
2	53.71	74.72						
3	50.43	71.12	69.34					
4	52.25	66.73	59.38	66.03				
5	52.06	63.26	61.29	63.94	94.60			
6	52.31	70.36	64.18	65.87	93.18	69.29		
7	54.46	74.32	63.20	62.30	82.01	66.20	77.96	
8	54.65	72.65	63.76	64.07	86.56	67.67	77.56	41.89

Table 8: Detailed results of Qwen2.5-7B-Instruct under OPR-RU with $\rho = 0.01$.

Task ID	C-STA	FOMC	MeBa	Py150	SciQA	NG-cm	NG-ds	20Min
1	56.41							
2	51.71	74.52						
3	51.71	74.55	69.35					
4	51.89	64.49	67.41	67.53				
5	50.12	58.62	68.24	64.92	93.85			
6	50.04	68.95	67.11	67.46	91.31	57.10		
7	50.95	71.27	66.14	64.67	87.04	52.16	75.35	
8	51.86	70.69	63.60	66.85	91.68	59.57	75.31	41.33

Table 9: Detailed results of Qwen2.5-7B-Instruct under OPR-SC with $\rho = 0.01$.