RETHINKING TRANSFER IN CONTINUAL LEARNING: A REPLAY-BASED REALISATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning studies how deployed language models can continually acquire new tasks without expensive retraining from scratch. Existing methods have largely focused on preventing forgetting, leaving a complementary question underexplored: when can past experience actually help a new task? We identify three measurable conditions for realised forward transfer: the target must have remaining headroom, transferred information must persist through continued optimisation, and replay must come from compatible prior tasks. These conditions motivate Transfer-Selective Replay (TSR), which uses a training-free gradient signature to route replay toward sources predicted to help the incoming task, while distillation maintains stability. Across heterogeneous and homogeneous continual-learning streams, model scales, and backbone families, TSR improves forward transfer and final performance without sacrificing retention. More broadly, our results suggest that continual learning should treat transfer alongside forgetting as a first-class design objective.

1 INTRODUCTION

Large language models are trained once at enormous cost and then deployed into a world that does not stand still: new domains, new formats, and new skills keep arriving after release (Wang et al., 2023b; 2024a; Tankala et al., 2025; Asawa et al., 2026). Retraining from scratch for every addition is prohibitive, so a deployed model must learn tasks *sequentially*, the setting studied in continual learning (De Lange et al., 2021; Ke & Liu, 2022; Momeni et al., 2025; Chen et al., 2026). Its defining failure mode has been known for decades: *catastrophic forgetting*, where optimising for the new task overwrites what earlier tasks built (Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017). The field frames the resulting tension as a *stability–plasticity* trade-off: stability preserves the past, plasticity lets the model keep learning, and improving either naively costs the other (De Lange et al., 2021).

Existing continual-learning methods, including rehearsal-based replay and rehearsal-free regularisation or parameter isolation (Chaudhry et al., 2019b; Rebuffi et al., 2017; Kirkpatrick et al., 2017; Wang et al., 2023a), overwhelmingly address the stability side of this trade-off. The complementary objective, *forward transfer*, is the improvement in learning a new task due to knowledge acquired from previous tasks. It has been measured since GEM (Lopez-Paz & Ranzato, 2017) yet remains far less understood: on modern LLM benchmarks it is routinely reported as absent or negative. Most explicit mechanisms for forward transfer have operated through parameter or architectural *reuse* (Rusu et al., 2016; Ke et al., 2020; Qin & Joty, 2022; Huang et al., 2023; Qian et al., 2025), implicitly treating past parameters as the transferable object. Meanwhile, past-task *data* has been used almost exclusively to prevent forgetting, and the two directions have remained largely separate. And although negative transfer is documented (Wang et al., 2021; Choi & Kim, 2026), the field has no explicit formulation of *when* transfer should be expected at all.

This paper starts one step earlier: *before designing a transfer mechanism, we ask what makes transfer possible.* We study this question in *low-budget* continual learning, where each incoming task arrives with limited supervision (Zhao et al., 2024) and therefore leaves measurable room for past experience to help. Realised transfer, we argue, requires three conditions (§4, Figure 1). (1) *Headroom*: the target must leave room beyond its own limited supervision (§4.1). (2) *Persistence*: the transfer carrier must survive continued optimisation (§4.2). (3) *Source compatibility*: past tasks can help or hurt, and the right ones are predictable at zero training cost (§4.3).

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.

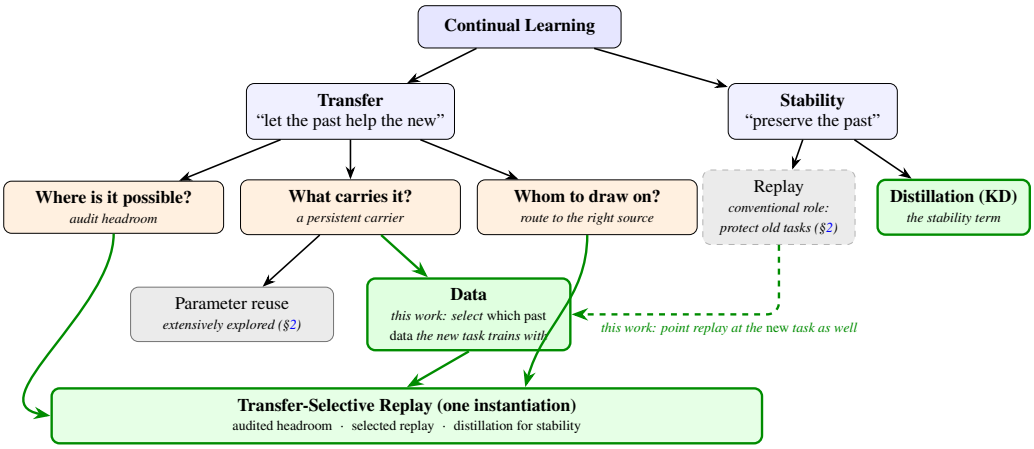


Figure 1: **Framework and instantiation.** Realised continual transfer requires headroom, a persistent carrier, and a compatible source. Transfer-Selective Replay instantiates these conditions with selected replay as the transfer carrier and distillation for stability.

Transfer-Selective Replay (TSR) instantiates these conditions: replay data serves as the persistent transfer carrier, a training-free gradient *task signature* routes each incoming task toward the past data predicted to help it, and distillation separately maintains stability (§5). Under the standard continual-learning protocol on TRACE-8 (Wang et al., 2023b) and NumGLUE-8 (Mishra et al., 2022), across three backbone families (Qwen Team, 2024; Gemma Team, 2025; Grattafiori et al., 2024) and eleven continual-learning baselines, TSR consistently improves forward transfer while maintaining positive backward transfer.

Contributions.

1. **Framework.** Three measurable conditions for continual transfer (headroom, carrier persistence, and source compatibility), each validated experimentally (§4).
2. **Mechanism.** Causal interventions show that gradient-aligned replay supplies compatible descent on the target, while distillation primarily serves as an optimisation anchor rather than a source of transferable knowledge (§4.3, §5.3).
3. **Instantiation and evidence.** TSR, which selectively replays beneficial past data and improves forward transfer without sacrificing stability across heterogeneous and homogeneous streams, scales, and backbone families (§5–6).

2 Related Work

Forward transfer. GEM formalised forward and backward transfer metrics (Lopez-Paz & Ranzato, 2017), and Progressive Networks designed for transfer (Rusu et al., 2016). GEM's metric is a zero-shot quantity, measured before training on the new task, whereas the training-time sense studied here is the improvement in learning the current task. Reusing source data for that purpose is well studied outside CL as intermediate-task training (STILTs; Phang et al., 2018; Vu et al., 2020), but rarely as a source-selective transfer mechanism within continual streams.

Parameter-side transfer. Mechanisms for training-time transfer are almost exclusively parameter-side: module reuse (Veniat et al., 2021), masks and adapters over a shared backbone (CAT/B-CL/CTR; Ke et al., 2020; 2021b;a), meta-learned representations (Javed & White, 2019), and soft-prompt reuse (Qin & Joty, 2022), a parameter-side emphasis also reflected in the NLP-CL literature (Ke & Liu, 2022). In the LLM era the same assumption drives inference-time composition (Huang et al., 2023; Ilharco et al., 2023; Chronopoulou et al., 2023; Ostapenko et al., 2024) and continual-training constraints: orthogonal adapters (O-LoRA/N-LoRA; Wang et al., 2023a; Yang et al., 2025), single-adapter merging (Qiao & Mahdavi, 2025), and gated, routed, or integrated

adapters (Zhang et al., 2025; Liang et al., 2025). These works treat parameters as the transferable object; we study the complementary role of data as a persistent transfer carrier.

Data-side transfer. Replay is classically a stability mechanism (ER, DER++, iCaRL; Chaudhry et al., 2019b; Buzzega et al., 2020; Rebuffi et al., 2017), and the replay-selection literature optimises the same objective. MIR (Aljundi et al., 2019a) picks maximally interfered samples and GSS (Aljundi et al., 2019b) gradient-diverse ones. The structurally closest works remain retention-directed: MER (Riemer et al., 2019) places a generic buffer inside a transfer-maximising meta-objective without selecting which past task helps, Adaptive Memory Replay (Smith et al., 2024) selects past-data clusters conditioned on the current task but rewards a forgetting metric, and InsCL (Wang et al., 2024b) modulates replay by task similarity in the *inverse* direction, giving dissimilar tasks more replay to protect them. Learned replay controllers frame replay as data selection, again for retention (Shihab et al., 2026). This paper inverts that objective. Doing so requires knowing which past task will help, a task-similarity question addressed by Fisher embeddings (Task2Vec; Achille et al., 2019), optimal-transport distances (OTDD; Alvarez-Melis & Fusi, 2020; Nguyen et al., 2025a), gradient-based task groupings (Fifty et al., 2021), example-level gradient selection (LESS; Xia et al., 2024), dataset traits (Krishna et al., 2025), and task orderings (Nguyen et al., 2025b). Closest inside CL, TreeLoRA (Qian et al., 2025) routes new tasks to adapter groups by gradient similarity at *evolving* weights, the same class of signal coupled to a parameter carrier.

Distillation. In continual learning, distillation is narrated as knowledge preservation, from LwF and iCaRL (Li & Hoiem, 2017; Rebuffi et al., 2017) to DER++'s stored logits (Buzzega et al., 2020) and recent self-distillation (Yang et al., 2024). Adjacent fields read KL-anchoring as regularisation, including reference-policy penalties in RL (Teh et al., 2017) and label-smoothing and regularisation accounts of supervised distillation (Yuan et al., 2020; Mobahi et al., 2020).

Negative transfer. Old knowledge can interfere with the learning of a new task (Wang et al., 2021; 2024a). MERIT (Choi & Kim, 2026) uses dataset-level gradient conflict to partition *parallel* instruction-tuning mixtures, and OPR (Chen et al., 2026) constructs on-policy replay from historical prompts and reward-filtered generations, addressing replay quality rather than source-task compatibility.

3 Problem Formulation

Continual learning. We consider task-incremental continual learning over a stream of tasks $\mathcal{T} = (T_1, \ldots, T_K)$, where each task T_t is a distribution D_t over input–output pairs and provides a training and a test sample. A single accumulating parameter state θ is trained through the stream, with no per-task parameters and no task identity at inference. When task T_t arrives, sequential learning solves

$$\theta_t \;=\; \arg\min_{\theta}\; \mathbb{E}_{(x,y)\sim D_t}\big[\,\ell(f_\theta(x),\, y)\big]\,, \tag{1}$$

optimised from θ_{t-1}, with earlier tasks accessible only through bounded retained experience (De Lange et al., 2021). Solving Eq. (1) naively degrades earlier tasks (catastrophic forgetting) and forgoes any help earlier tasks could give the current one. Throughout, we study the *low-budget* regime: each incoming task provides only a limited number of labelled examples (§6.1).

Transfer. *Forward transfer* refers to the contribution of previously acquired experience to the learning of the current task, under the fixed target-data budget. Conceptually, we decompose current-task learning into two contributions:

$$\text{Current-task learning} \;=\; \underbrace{\text{Target-data contribution}}_{\text{what the task teaches itself}} \;+\; \underbrace{\text{Transfer contribution}}_{\text{what the past supplies}}.$$

The transfer contribution is the object of this paper. It is distinct from GEM's zero-shot forward-transfer metric (Lopez-Paz & Ranzato, 2017), and it is *latent*: no standard continual-learning metric reads it off directly. §4 constructs its measurement and asks under what conditions it exists at all.

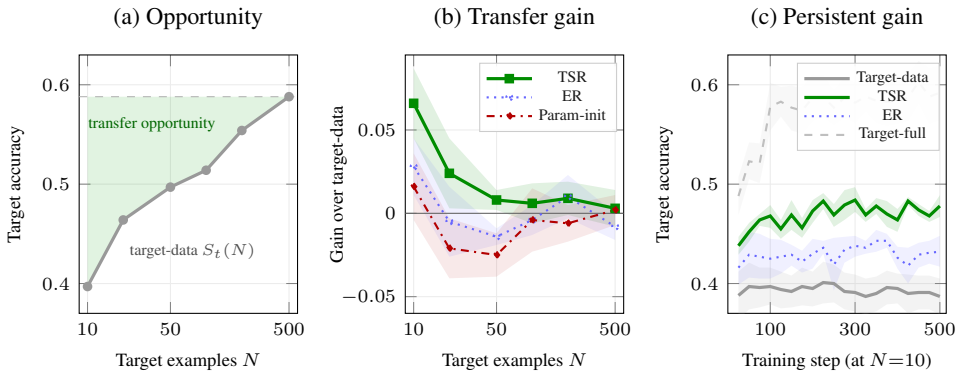


Figure 2: **Transfer gains shrink as target headroom closes.** *(a)* Headroom across target-data budgets. *(b)* Transfer gain from selected (TSR) and uniform (ER) replay. *(c)* Transfer gain over training at $N = 10$. Details are in Appendix A.2.

4 Three Conditions for Realised Transfer

Realised forward transfer in a continual stream requires

$$\text{Transfer}(j \to t) > 0 \;\text{ requires }\; \underbrace{\text{Headroom}(t)}_{\textit{Where?}} \wedge \underbrace{\text{a Persistent Carrier}}_{\textit{What?}} \wedge \underbrace{\text{the Right Source } (j)}_{\textit{Whom?}}. \quad (2)$$

4.1 Where: Headroom as Transfer Opportunity

Condition 1 *Realised transfer requires measurable headroom on the target: additional in-domain supervision must still improve performance.*

Headroom. We measure whether a target task has room to benefit from transfer by comparing its low-budget performance with a well-resourced in-domain reference. Let $S_t(N)$ denote the test accuracy obtained by training on N examples from target task t. We define

$$\text{Headroom}_t(N) = S_t(N_{\max}) - S_t(N), \quad (3)$$

where $N_{\max}$ is ten times the standard per-task budget in our experiments, giving the improvement still available from additional in-domain supervision.

Evidence. Figure 2 shows that headroom and transfer vanish together. As the target-data budget increases from $N = 10$ to $N = 500$, accuracy rises from .397 to .588 and the gain from selected source data correspondingly shrinks toward zero. At $N = 10$, the $+.066$ transfer gain is equivalent to roughly doubling the target supervision ($\approx$20 examples), recovering 35% of the available headroom. Panel (c) shows the gain is present from the first evaluation and sustained for all 500 steps.

A low-headroom benchmark provides the complementary control: on PP15 (Razdaibiedina et al., 2023), increasing target data tenfold improves accuracy by only $+.02$ on average, and none of fourteen parameter-reuse configurations yields measurable transfer (Appendix A.1). TRACE-8 (Wang et al., 2023b) retains substantial headroom at the same scale and is therefore our primary benchmark.

4.2 What: The Carrier Must Persist

Condition 2 *Realised transfer requires a carrier whose influence persists throughout target optimisation.*

We isolate persistence by holding the source, target, and training budget fixed while varying how source information is carried (Figure 3). One-shot initialisation from a source-trained adapter yields an early advantage that continued target training washes out. Keeping the same adapter present through an L2 anchor recovers roughly two-thirds of the mixing gain, while persistent replay recovers

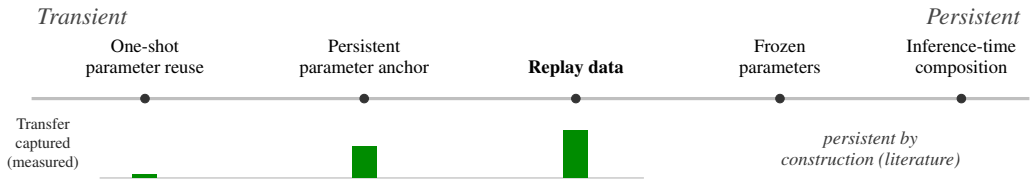


Figure 3: **Transfer follows carrier persistence.** Bars show the fraction of the data-mixing gain recovered by the three measured carriers (Appendix B.3); frozen parameters and inference-time composition are persistent by construction.

Table 1: **Pairwise transfer on TRACE-8.** Each cell reports the loss change from mixing source j with $N = 50$ target examples (negative is better; bold = column best). Green denotes positive transfer and red negative transfer.

source\target	cstn	fomc	meet	py15	sciq	ngcm	ngds	20mn
cstance	—	**−0.18**	−0.20	−0.70	+0.19	−0.01	−0.69	−0.63
fomc	**−0.07**	—	−0.24	−0.45	**−0.09**	+0.14	−0.27	−0.69
meetingbank	+0.02	−0.02	—	−0.34	+0.12	**−0.26**	−0.53	−0.45
py150	−0.03	−0.15	−0.29	—	+0.16	−0.10	−0.28	−0.46
scienceqa	−0.04	−0.09	**−0.36**	**−0.76**	—	+0.26	−0.47	−0.75
numglue-cm	+0.01	−0.04	−0.21	−0.68	−0.03	—	**−0.88**	**−0.85**
numglue-ds	+0.01	−0.11	−0.29	−0.62	−0.01	−0.20	—	−0.80
20minuten	+0.06	−0.02	−0.18	−0.35	+0.27	−0.16	+0.00	—

the full gain. Transfer therefore tracks *persistence*, not whether its carrier is parameters or data. Successful parameter-based approaches such as frozen modules and inference-time composition are consistent with this view because they persist by construction (Appendix B.2). We therefore use replay data as TSR's persistent carrier in §5.

4.3 WHOM: THE SOURCE MUST BE COMPATIBLE

Condition 3 *Given headroom and a persistent carrier, realised transfer depends on choosing a compatible source; the wrong source can instead cause negative transfer.*

Table 1 measures every ordered source–target pair on TRACE-8. Transfer is highly source-dependent: NumGLUE-cm→NumGLUE-ds yields the largest target-loss reduction ($\Delta L = -0.88$), whereas mismatched sources can increase target loss by up to $+0.27$. Continual replay therefore requires source selection rather than indiscriminate reuse.

Exhaustively measuring the matrix requires training every ordered pair, so deployment needs a training-free proxy. Among input similarity, distribution geometry, and learning-direction signatures, only gradient direction reliably identifies the best source (Table 2). For task t, we define

$$g_t = \frac{1}{m}\sum_{b=1}^{m} \nabla_B \mathcal{L}_{\mathrm{CE}}(b)\Big|_{(A_0,B_0)}, \qquad \mathrm{score}(j,t) = \cos(g_j, g_t), \tag{4}$$

where g_t averages the gradients of m probe batches at the shared initialisation (A_0, B_0). Because B is the zero-initialised LoRA factor, g_t captures the adapter's first-order learning direction before task-specific training. It identifies the oracle-best source for $4/5$ classification targets; its only miss selects the second-best. Across all eight targets, input- and distribution-based alternatives reach at most $2/8$ top-1 accuracy. Direction is the critical signal: normalising gradients preserves source selection, whereas discarding direction through a Fisher-style magnitude representation does not.

Why does gradient alignment predict transfer? A gradient signature captures what a task asks the model to *learn* rather than what its inputs look like. We test this interpretation with a local intervention at the shared initialisation. Holding the model state fixed, we add an equal-norm source update to a target update and measure the resulting change in held-out target loss (Appendix A.2). More aligned sources produce strictly greater marginal descent (Table 2, right); current-gradient and

Table 2: **Predicting and testing source compatibility.** Left: top-1 source prediction against the measured transfer matrix. Right: marginal target-loss change after an equal-norm source intervention at the shared initialisation (three classification targets). Details in Appendix A.2.

Prediction

Signature	Signal	Top-1
Proto	input similarity	$1/8$
HiddenOT	distribution geometry	$2/8$
JointOT	label-aware distribution	$0/8$
Fisher	gradient magnitude	$1/8$
GradB_last4	gradient direction (last 4 blocks)	$3/8$
GradB_norm	gradient direction (normalised)	$\mathbf{4/8}$
GradB	gradient direction (Eq. 4)	$\mathbf{4/8}$

Mechanism

Added update	marginal ΔL
Extra target batch	-1.08
Top-aligned source	$-.30$
Middle source	$-.12$
Bottom source	$-.02$

Algorithm 1 General TSR

Require: task stream $T_1, \ldots, T_K$ with training data $D_1, \ldots, D_K$; initial parameters θ_0; modules PROBE, ROUTE, REFERENCE; hyperparameters α, λ, η

$\mathcal{P} \leftarrow \{\}$; $\theta \leftarrow \theta_0$
for $t = 1, \ldots, K$ **do**
  $g_t \leftarrow \textsc{Probe}(D_t)$
  $w \leftarrow \textsc{Route}\big(g_t,\ \{g_j\}_{j<t}\big)$
  for each training step **do**
    with prob. $1-\alpha$: $b \sim D_t$; otherwise: $j \sim w$, $b \sim \mathcal{M}_j$
    $\theta^{\mathrm{ref}} \leftarrow \textsc{Reference}(b,\ \mathcal{P})$
    $\theta \leftarrow \theta - \eta \nabla_\theta \big[\, \mathcal{L}_{\mathrm{CE}}(\theta; b) + \lambda\, \mathcal{L}_{\mathrm{KD}}(\theta;\, \theta^{\mathrm{ref}}, b) \,\big]$
  end for
  $\mathcal{M}_t \leftarrow$ subsample of D_t; $\theta_t \leftarrow \theta$; $\mathcal{P} \leftarrow \mathcal{P} \cup \{(g_t,\ \mathcal{M}_t,\ \theta_t)\}$
end for

stored-signature alignment predict this reduction with $\rho = .87$ and $.85$, respectively. The top-aligned source recovers roughly 28% of the descent from an additional target batch without using additional target labels.

5 TRANSFER-SELECTIVE REPLAY

5.1 THE GENERAL ALGORITHM

The three conditions of §4 lead to a simple algorithmic structure: persistently replay past data selected for the incoming task, while handling stability separately. Algorithm 1 expresses this structure with three replaceable modules: PROBE represents the incoming task, ROUTE weights past tasks by predicted transfer, and REFERENCE supplies the stability target. Routed replay is the persistent transfer carrier shared by all instantiations. Each completed task contributes a record (g_j, M_j, θ_j) containing its signature, replay memory, and reference snapshot; routing over these records jointly determines the past data and reference used during training.

5.2 TSR INSTANTIATION

Algorithm 2 instantiates the general form using the measurements of §4. Alternative choices for PROBE, ROUTE, and REFERENCE fit the same skeleton (Appendix B.4).

Probe. TSR uses the gradient signature of Eq. 4, computed from m probe batches at the shared initialisation θ_0. This requires only forward–backward passes and no task-specific training.

Route. Past tasks are weighted by signature similarity, $w_j \propto \exp(\cos(g_j, g_t)/\tau)$. The temperature interpolates between selecting a single source ($\tau \to 0$) and uniform replay ($\tau \to \infty$).

Algorithm 2 Routed-triple TSR

Require: task stream $T_1, \ldots, T_K$ with training data $D_1, \ldots, D_K$; shared initialisation θ_0; hyperparameters m, τ, α, λ, η

$\mathcal{P} \leftarrow \{\}$; $\theta \leftarrow \theta_0$
for $t = 1, \ldots, K$ **do**
  $g_t \leftarrow \frac{1}{m} \sum_{i=1}^{m} \nabla_B \mathcal{L}_{\mathrm{CE}}(\theta_0; b_i),\ b_i \sim D_t$ ▷ PROBE
  $w_j \leftarrow \mathrm{softmax}_{j<t}\big(\cos(g_j, g_t)/\tau\big)$ ▷ ROUTE
  $j^* \leftarrow \arg\max_{j<t} w_j$
  for each training step **do**
    draw $u \sim \mathrm{Uniform}(0, 1)$
    if $u > \alpha$ **then**
      sample minibatch $b \sim D_t$; $\theta^{\mathrm{ref}} \leftarrow \theta_{j^*}$ ▷ current-task batch
    else
      draw one source $j \sim \mathrm{Categorical}(w)$ ▷ past task j chosen with prob. w_j
      sample minibatch b uniformly from $\mathcal{M}_j$
      $\theta^{\mathrm{ref}} \leftarrow \theta_j$ ▷ REFERENCE
    end if
    $\theta \leftarrow \theta - \eta \nabla_\theta \big[\mathcal{L}_{\mathrm{CE}}(\theta; b) + \lambda \mathrm{KL}\big(f_{\theta^{\mathrm{ref}}}(b) \,\|\, f_\theta(b)\big) \big]$
  end for
  $\mathcal{M}_t \leftarrow$ subsample of D_t; $\theta_t \leftarrow \theta$; $\mathcal{P} \leftarrow \mathcal{P} \cup \{(g_t, \mathcal{M}_t, \theta_t)\}$
end for

Replay. At replay steps, TSR samples a task $j \sim w$ and then a minibatch from M_j. Reintroducing routed source data throughout optimisation provides the persistent carrier of §4.2.

Reference. A replayed batch from M_j uses its corresponding snapshot θ_j as the frozen reference; current-task batches use the highest-weighted past snapshot. Section 5.3 examines the resulting distillation term and its role in stability.

5.3 DISTILLATION FOR STABILITY

Routed replay is assigned to transfer, leaving stability as a separate objective. TSR handles it with knowledge distillation: for each training batch b,

$$\mathcal{L}_{\mathrm{KD}}(\theta; \theta^{\mathrm{ref}}, b) = \mathrm{KL}(f_{\theta^{\mathrm{ref}}}(b) \,\|\, f_\theta(b)) \,, \tag{5}$$

which is added to the task loss with weight λ. The reference is tied to the task record selected by routing: a replayed batch from M_j is anchored to its corresponding era snapshot θ_j, while a current-task batch uses the highest-weighted past snapshot. Thus replay determines *what past experience to train on*, whereas distillation constrains how far the accumulating model moves from a previously learned predictor.

Why separate these roles? We audit the distillation mechanism by varying teacher identity, teacher content, loss shaping, timing, and snapshot source (Appendix B.5). Across these interventions, the gain depends much more on the presence and timing of a stable reference than on task-specific knowledge carried by the teacher. Attempts to make the teacher itself carry forward transfer do not improve transfer and can weaken stability. In this regime, distillation therefore acts primarily as an optimisation anchor, rather than as the source of transferable knowledge. The remaining teacher-specific effect is long-horizon drift: recent snapshots become increasingly biased toward later tasks, motivating the era-matched references used above.

6 EXPERIMENTS

6.1 SETUP

Setting. We evaluate Qwen2.5-0.5B/7B-Instruct with a single continually trained LoRA adapter (Hu et al., 2021) on two eight-task streams: TRACE-8 (Wang et al., 2023b) (heterogeneous) in its

Table 3: **Continual learning under the standard sequential protocol** (TRACE-8 / NumGLUE-8 at 0.5B / 7B). Mean ± std over 5 seeds; bold = best per column. Rules separate the no-mechanism baseline, the rehearsal-free family, the rehearsal-based family, and ours. Metrics, baselines, and protocol as in §6.1; per-task results in Appendix C.2.

	TRACE-8 (0.5B)			TRACE-8 (7B)			NumGLUE-8 (0.5B)			NumGLUE-8 (7B)		
Method	Overall↑	Plas↑	BWT↑	Overall↑	Plas↑	BWT↑	Overall↑	Plas↑	BWT↑	Overall↑	Plas↑	BWT↑
SeqFT	.264±.048	.398±.015	−.134±.058	.584±.033	.660±.001	−.076±.031	.284±.041	.316±.018	−.037±.053	.555±.032	**.587**±.008	−.037±.037
EWC	.245±.031	.427±.004	−.182±.031	.599±.010	.666±.011	−.067±.014	.279±.018	.320±.014	−.047±.030	.554±.035	.584±.009	−.034±.034
LwF	.291±.045	.424±.013	−.133±.033	.640±.024	.668±.001	−.029±.023	.292±.014	.313±.013	−.024±.008	.563±.015	.580±.017	−.020±.007
SDFT	.305±.070	.427±.009	−.122±.072	.605±.011	.664±.009	−.059±.019	.279±.033	**.334**±.016	−.063±.027	.553±.009	.580±.011	−.030±.011
O-LoRA	.233±.028	.401±.007	−.168±.030	.617±.010	.662±.010	−.045±.005	.281±.016	.310±.028	−.033±.014	.567±.006	.579±.014	−.013±.010
GainLoRA	.374±.018	.398±.022	−.024±.005	.654±.010	.663±.009	−.009±.005	.297±.014	.307±.017	−.011±.004	.558±.006	.563±.005	−.005±.004
DEAL	.372±.019	.401±.003	−.029±.022	.224±.153	.576±.078	−.352±.088	.253±.024	.303±.013	−.057±.023	.418±.023	.500±.006	−.094±.023
A-GEM	.310±.019	.418±.016	−.107±.021	.610±.032	.662±.007	−.052±.025	.286±.031	.325±.005	−.045±.037	.559±.025	.586±.014	−.031±.013
DER++	.220±.015	.262±.022	−.042±.037	.264±.104	.288±.096	−.023±.081	.210±.025	.212±.028	−.003±.007	.393±.054	.445±.010	−.060±.051
ER (10% replay)	.406±.019	.435±.008	−.029±.025	.638±.029	.663±.005	−.025±.026	.321±.020	.331±.013	−.012±.011	.563±.011	.583±.013	−.023±.016
ER	.451±.031	.427±.024	+.025±.020	.656±.034	.662±.018	−.006±.046	.336±.013	.314±.011	+.025±.010	.578±.012	.579±.015	−.001±.019
TSR (ours)	**.475**±.030	**.439**±.017	**+.035**±.015	**.684**±.008	**.670**±.012	**+.014**±.011	**.353**±.016	.330±.012	**+.026**±.007	**.588**±.017	.582±.012	**+.007**±.022

canonical order, and NumGLUE-8 (Mishra et al., 2022) (one numerical-reasoning core in eight formats) in type order. We follow the standard sequential protocol of the systems we compare against (Wang et al., 2023a; Qian et al., 2025): one adapter trains through all eight tasks and all seen tasks are evaluated after each task, with N=50 examples per task and n=5 seeds. All remaining configuration is in Appendix C.1.

Metrics. Let $A_{k,i}$ denote accuracy on task T_i's test set after training on task T_k. We report the three standard stream metrics (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019a),

$$\text{Overall} = \frac{1}{K}\sum_i A_{K,i}, \qquad \text{Plas} = \frac{1}{K}\sum_i A_{i,i}, \qquad \text{BWT} = \frac{1}{K-1}\sum_{i<K}\left(A_{K,i} - A_{i,i}\right),$$

where 0 BWT means perfect retention and positive BWT means old tasks improved.

Baselines. Naive sequential fine-tuning (SeqFT); *Rehearsal-free:* EWC (Kirkpatrick et al., 2017), LwF (Li & Hoiem, 2017), SDFT (Yang et al., 2024), O-LoRA (Wang et al., 2023a), GainLoRA (Liang et al., 2025), and DEAL (Han et al., 2025); *Rehearsal-based:* A-GEM (Chaudhry et al., 2019a), DER++ (Buzzega et al., 2020), and ER (Chaudhry et al., 2019b) at 50% and 10% replay ratios.

6.2 Does transfer improve low-budget continual learning?

Overall performance. TSR achieves the highest final Overall accuracy in all four settings (Table 3). Since every method receives the same 50 target examples per task, these gains come from better use of experience already available in the stream rather than additional target supervision. The margins over replay baselines are smaller than the pairwise gains in §4.1: by the time a target arrives, preceding tasks have already updated the shared adapter, reducing the additional benefit available from replaying any single source.

Transfer. We next examine Plas, where the contribution of past experience to current-task learning should appear most directly. On TRACE, TSR achieves the highest Plas at both scales (.439 at 0.5B and .670 at 7B), outperforming uniform replay under the same target-data and replay budgets. This extends the source-selection advantage of §4.1 from controlled pairs to the full continual stream. On NumGLUE, the advantage is smaller: several methods match or slightly exceed TSR in Plas, but with substantially worse BWT. High current-task performance alone therefore does not imply useful continual transfer when earlier tasks are simultaneously lost.

Stability. The improvement in current-task learning does not come at the expense of retention. TSR is the only method with positive BWT in all four settings. Selected replay continues to expose the model to useful past data, while distillation constrains drift from the corresponding source-era references. Together, these effects explain why the transfer gains translate into the strongest final performance.

Backbone robustness. The same qualitative ordering holds on Llama-3.2-1B and Gemma-3-1B (Appendix C.3): replay-based methods remain strongest, and TSR again achieves the best Overall accuracy, Plas, and BWT. The gains are therefore not specific to Qwen2.5.

6.3 What determines transfer?

The gains in Table 3 vary substantially across streams and model scales. The framework predicts two sources of this variation: task structure determines how much source selection matters, while headroom determines how much transfer is available in the first place. We examine each in turn.

Task structure. Source selection matters most when past tasks differ in how they affect the target. TRACE exhibits precisely this regime: its pairwise transfer matrix contains both strongly beneficial and harmful sources (Table 1), and routed replay outperforms uniform pooling. NumGLUE presents the complementary case. Its eight tasks share a common arithmetic core, producing a denser and more homogeneous transfer structure in which source identity matters less and uniform pooling becomes competitive (Appendix A.4.1). Gradient-signature similarities also compress within this family, reducing the selector's ability to distinguish among closely related sources. The benefit of routing therefore tracks the variation in transfer quality across candidate sources.

Headroom. Transfer opportunity is a property of the model–task regime, not of the task alone. On TRACE, stronger Qwen2.5 models increasingly saturate the target under limited supervision, so the benchmark leaves progressively less headroom and the best-source gain correspondingly decreases, becoming negligible at 7B (Figure 4). NumGLUE shows the same relationship more gradually: measurable in-domain headroom remains at 7B, while the gain supplied by other tasks shrinks from +.033 to +.003. Across benchmarks, PP15 similarly provides little measurable headroom, whereas SuperNI retains transfer opportunity but offers little source-selection advantage (Appendix A). Together, these results separate two roles of the framework: headroom determines whether there is transfer opportunity to exploit, while task structure determines whether selecting among sources is useful.

6.4 Design choices

The framework specifies *what* must be decided, not *how*. Algorithm 2 leaves implementation choices open, and we ablate these:

Routing. The routing module decides whether replay comes from a single selected source or a weighted combination, and the framework is agnostic, so we compare argmax selection, soft routing, and uniform pooling. In the stream, soft routing beats both extremes and its single-component variants (Table 3; Appendix B.4). On the controlled probe, even a softmax-weighted top-3 mixture *lowers* mean accuracy against the single selected source. The reason is the shape of the matrix: on heterogeneous streams it is sparse, the second- and third-best sources are often weak or harmful, and averaging them dilutes transfer. **Sparse transfer matrices favour selection over averaging.**

Task order. If the matrix is usable prospectively, ordering tasks so that strong sources precede their targets should help. We compare the canonical order, the greedy best-source curriculum fixed before evaluation, its reversal, and arbitrary reshuffles, at both scales (Appendix C.3). The replay family moves by at most .03 under any of them, on both streams and both scales. Replay already delivers the relevant source at training time, so a curriculum has little left to contribute. **Selection replaces curriculum.**

7 Discussion and Conclusion

Our results identify three conditions for realised continual transfer: sufficient headroom on the incoming task, a carrier that persists through optimisation, and compatibility between the source and target. These conditions give replay a role beyond preventing forgetting: when past data are selected for the incoming task, replay can directly improve new-task learning, while distillation separately maintains stability. They also have implications for evaluation: transfer evaluations should ensure measurable target headroom for the model–task regime and enough variation in source quality for source selection to matter. Our study focuses on low-budget adaptation with access to a small replay buffer. Settings where raw data cannot be stored will require alternative persistent carriers, while the reduced resolution of our gradient signature among homogeneous tasks motivates better transfer predictors. TSR is therefore one instantiation of the framework; its signature, replay policy, and stability mechanism are all replaceable. Ultimately, continual learning should evaluate transfer as systematically as it has evaluated forgetting.

AI USE STATEMENT

We used large language model (LLM) assistants to aid in drafting and polishing the presentation of this paper and for literature retrieval and discovery. All scientific claims, experimental results, and cited references were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

Appendix C provides the complete experimental configuration, evaluation protocol, per-task results, robustness analyses, and reproducibility details. The controlled probes and supporting ablations are documented in Appendices A and B. An anonymised implementation reproducing the core TRACE-8 experiments is provided in the supplementary material.

REFERENCES

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *ICCV*, 2019.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, 2019a.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, 2019b.

David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. In *Advances in Neural Information Processing Systems*, 2020.

Parth Asawa, Christopher M. Glaze, Gabriel Orlanski, et al. Continual learning bench: Evaluating frontier AI systems in real-world stateful environments. *arXiv preprint arXiv:2606.05661*, 2026.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, 2020.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations*, 2019a.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019b.

Yan Chen, Taojie Zhu, Meng Zhang, Xin Chen, Jiaqi Huang, Dongyang Xu, and Yizhi Wang. On-policy replay for continual supervised fine-tuning. *arXiv preprint arXiv:2605.29495*, 2026.

Minsik Choi and Geewook Kim. Decentralized instruction tuning: Conflict-aware splitting and weight merging. In *International Conference on Machine Learning*, 2026.

Alexandra Chronopoulou, Matthew E Peters, Alexander Fraser, and Jesse Dodge. Adaptersoup: Weight averaging to improve generalization of pretrained language models. In *Findings of EACL*, 2023.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In *Advances in Neural Information Processing Systems*, 2021.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Xiao Han, Zimo Zhao, Wanyu Wang, Maolin Wang, Zitao Liu, Yi Chang, and Xiangyu Zhao. Data efficient adaptation in large language models via continuous low-rank fine-tuning. In *Advances in Neural Information Processing Systems*, 2025.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic LoRA composition. *arXiv preprint arXiv:2307.13269*, 2023.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, 2019.

Zixuan Ke and Bing Liu. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*, 2022.

Zixuan Ke, Bing Liu, and Xingchang Huang. Continual learning of a mixed sequence of similar and dissimilar tasks. In *Advances in Neural Information Processing Systems*, 2020.

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. Achieving forgetting prevention and knowledge transfer in continual learning. In *Advances in Neural Information Processing Systems*, 2021a.

Zixuan Ke, Hu Xu, and Bing Liu. Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks. In *NAACL*, 2021b.

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

Shambhavi Krishna, Atharva Naik, Chaitali Agarwal, Sudharshan Govindan, Taesung Lee, and Haw-Shiuan Chang. Latent traits and cross-task transfer: Deconstructing dataset interactions in LLM fine-tuning. In *Proceedings of the 14th Joint Conference on Lexical and Computational Semantics (*SEM)*, pp. 225–241, 2025.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Yan-Shuo Liang, Jia-Rui Chen, and Wu-Jun Li. Gated integration of low-rank adaptation for continual learning of large language models. In *Advances in Neural Information Processing Systems*, 2025.

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

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 *Proceedings of ACL*, 2022.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in Hilbert space. In *Advances in Neural Information Processing Systems*, 2020.

Saleh Momeni, Sahisnu Mazumder, Zixuan Ke, and Bing Liu. In-context continual learning assisted by an external continual learner. In *COLING*, 2025.

Khai Nguyen, Hai Nguyen, Tuan Pham, and Nhat Ho. Lightspeed geometric dataset distance via sliced optimal transport. In *International Conference on Machine Learning*, 2025a.

Thinh Nguyen, Cuong N Nguyen, Quang Pham, Binh T Nguyen, Savitha Ramasamy, Xiaoli Li, and Cuong V Nguyen. Sequence transferability and task order selection in continual learning. *arXiv preprint arXiv:2502.06544*, 2025b.

Oleksiy Ostapenko, Zhan Su, Edoardo Maria Ponti, Laurent Charlin, Nicolas Le Roux, Matheus Pereira, Lucas Caccia, and Alessandro Sordoni. Towards modular LLMs by building and reusing a library of LoRAs. In *International Conference on Machine Learning*, 2024.

Jason Phang, Thibault Févry, and Samuel R Bowman. Sentence encoders on STILTs: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*, 2018.

Yu-Yang Qian, Yuan-Ze Xu, Zhen-Yu Zhang, Peng Zhao, and Zhi-Hua Zhou. TreeLoRA: Efficient continual learning via layer-wise LoRAs guided by a hierarchical gradient-similarity tree. In *ICML*, 2025.

Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single LoRA continual learning via continual merging. *arXiv preprint arXiv:2512.23017*, 2025.

Chengwei Qin and Shafiq Joty. LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of T5. In *International Conference on Learning Representations*, 2022.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. Progressive prompts: Continual learning for language models. In *International Conference on Learning Representations*, 2023.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Ibne Farabi Shihab, Fariya Afrin, and Anuj Sharma. Dynamic proxy-mixing: Transferring replay controllers from small to large models for continual instruction tuning. *arXiv preprint arXiv:2606.00400*, 2026.

James Seale Smith et al. Adaptive memory replay for continual learning. In *CVPR Workshops*, 2024. arXiv:2404.12526.

Pavan Kalyan Tankala, Shubhra Mishra, Satya Lokam, and Navin Goyal. CurLL: A developmental framework to evaluate continual learning in language models. In *BabyLM Workshop*, 2025.

Yee Whye Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Tom Veniat, Ludovic Denoyer, and Marc'Aurelio Ranzato. Efficient continual learning with modular networks and task-driven priors. In *International Conference on Learning Representations*, 2021.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. Exploring and predicting transferability across NLP tasks. In *EMNLP*, 2020.

Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. AFEC: Active forgetting of negative transfer in continual learning. In *Advances in Neural Information Processing Systems*, 2021.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a.

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

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, et al. TRACE: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*, 2023b.

Yifan Wang et al. InsCL: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *NAACL*, 2024b. arXiv:2403.11435.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*, 2024.

Shuo Yang, Kun-Peng Ning, Yu-Yang Liu, et al. Is parameter collision hindering continual learning in LLMs? In *COLING*, 2025.

Zhaorui Yang, Tianyu Pang, Haozhe Feng, Han Wang, Wei Chen, Minfeng Zhu, and Qian Liu. Self-distillation bridges distribution gap in language model fine-tuning. In *ACL*, 2024.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Xin Zhang, Liang Bai, Xian Yang, and Jiye Liang. C-LoRA: Continual low-rank adaptation for pre-trained models. *arXiv preprint arXiv:2502.17920*, 2025.

Zhuokai Zhao, Yibo Jiang, and Yuxin Chen. Direct acquisition optimization for low-budget active learning. *arXiv preprint arXiv:2402.06045*, 2024.

APPENDIX

Appendix guide. Appendix A collects the benchmark-level evidence for the framework of §4, spanning four regimes: PP15 (no headroom), TRACE (controlled transfer measurement), SuperNI (generic transfer), and NumGLUE (homogeneous transfer). Appendix B collects the implementation evidence behind §5: carrier, replay, routing, and distillation. Appendix C is the experimental record behind §6: setup, per-task results, robustness, and reproducibility. Unless stated otherwise, appendix results follow the probe protocol of §4.1 and the stream protocol of §6.1.

A BENCHMARK REGIMES

A.1 PP15: THE ZERO-HEADROOM REGIME

Under the audit of §4.1, PP15 with an instruction-tuned base leaves essentially no transfer opportunity: ten times more target supervision improves accuracy by only $+0.02$ on average. Consistently, the best of fourteen parameter-reuse configurations (oracle-selected source; initialisation, perturbation, moment, and projection families; n=5) improves the no-transfer baseline by $+0.005$ (within one seed standard deviation), and the mean across configurations is -0.008. No transient advantage survives past $\sim$50 steps under early-budget checks.

A.2 TRACE-8: CONTROLLED TRANSFER MEASUREMENT

This appendix records the controlled protocol behind the transfer claims of §4: the five probe arms (target-data, target-full, mixed with a chosen source, random and hindsight-oracle selection) run per target on a fresh adapter.

Budget sweep. We repeat the five probe arms of §4.1 at $N \in \{10, 20, 50, 100, 200, 500\}$ on the five TRACE classification targets, with 500 source examples and 50/50 mixing. Headroom and selected-source gains shrink together as N increases. At $N = 10$, NumGLUE-cm gains $+.066$ from its selected source, recovering 35% of the available headroom, whereas matrix-weak targets remain near zero. Parameter initialisation provides no measurable gain at any budget.

Task-signature robustness. Signatures use 10 probe batches and are evaluated by top-1 recovery of the measured best source (chance $\approx 14\%$). Using only the last four blocks reduces top-1 recovery from 4/8 to 3/8, indicating that the signal is distributed across depth. All signatures are measured at the same shared initialisation, providing a fixed pairwise representation before continual training begins.

Local intervention. We test three classification targets spanning the transfer matrix. At the shared initialisation, we apply equal-norm source and target gradient perturbations and evaluate their effect on held-out target loss, using three step scales and 30 repetitions per state. Gradient alignment strongly predicts immediate loss reduction ($\rho = .93$ source-only; .87 marginal), and the top/middle/bottom source groups preserve their ordering at all three scales. Stored-signature alignment also predicts the effect ($\rho = .85$), while the top-aligned source recovers roughly 28% of the descent of an additional target batch. This diagnostic is intentionally local: its predictive power diminishes after the small target set is largely fitted, while long-horizon effects are evaluated by the stream experiments.

A.3 SUPERNI: THE GENERIC-TRANSFER REGIME

SuperNI illustrates the generic-transfer boundary predicted by the framework (§6.3). On six held-out Super-NaturalInstructions pairs (n=5), headroom exists: additional in-domain supervision improves loss on $6/6$ pairs. PP15's zero-headroom regime, not TRACE's, is the anomaly. Source data helps: mixed training improves target loss on $5/6$ pairs. But selection has almost no value: related and unrelated sources help nearly equally ($+0.04$ loss difference). The selection premium correspondingly shrinks when source-quality variation is small. We report this boundary rather than average over it.

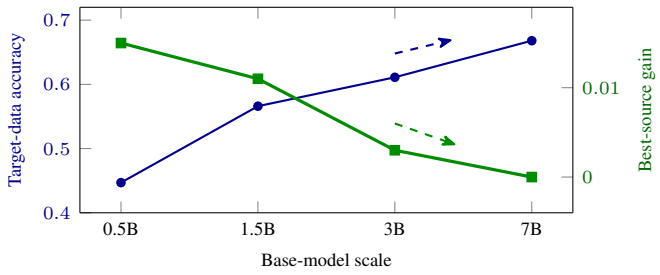


Figure 4: **Transfer follows headroom across model scale** (TRACE headroom audit per Qwen2.5 scale; error bands omitted for clarity).

A.4 NumGLUE-8: the homogeneous regime

A.4.1 Headroom audit

NumGLUE-8 casts the eight task types of the NumGLUE suite (Mishra et al., 2022) as the homogeneous regime: one numerical-reasoning core in eight formats (types 4/5 are TRACE's numglue-cm/ds). The §4.1 audit gives three regularities. Headroom exists and persists across scale: $7/8$ types retain accuracy headroom at 0.5B and at 7B (n=5). Pooling beats single-source selection on the probe at both scales (0.348 vs. 0.342 at 0.5B, 0.600 vs. 0.585 at 7B), matching the pooling prediction of §4.3. And the within-family oracle ceiling compresses from $+0.033$ to $+0.003$ across the same scales: in-domain headroom persists while the share other tasks can supply shrinks.

A.4.2 Pairwise transfer matrix

Full transfer matrix (Δacc) behind §6.3 (Δacc of $50/50$ joint training vs. target-data; n=5; green = helps, red = hurts, bold = column best; T4/T5 are TRACE's numglue-cm/ds).

Table 4: Pairwise transfer matrix on NumGLUE-8.

src\tgt	T1	T2	T3	T4	T5	T6	T7	T8
T1	—	−0.013	−0.015	+0.015	+0.008	−0.015	−0.018	+0.006
T2	**+0.060**	—	−0.015	+0.041	+0.016	−0.011	+0.004	+0.017
T3	+0.012	+0.006	—	+0.013	−0.003	−0.002	**+0.005**	−0.006
T4	+0.054	+0.001	−0.003	—	**+0.041**	−0.013	−0.015	**+0.047**
T5	+0.012	−0.007	−0.027	+0.037	—	−0.024	−0.026	−0.009
T6	+0.024	**+0.015**	−0.006	+0.000	+0.007	—	−0.005	+0.009
T7	+0.048	−0.012	**+0.018**	+0.041	+0.011	−0.007	—	−0.009
T8	+0.054	−0.009	−0.030	**+0.048**	+0.009	**+0.005**	−0.017	—

B Implementation Evidence

B.1 The TSR pipeline

Figure 5 traces one task through the evaluated instantiation. Its three panels correspond to the three replaceable modules of Algorithm 1: Probe produces the signature at the shared initialisation, Route matches it against the task memory to pick both the replay data and the reference snapshot, and training combines the task loss with the distillation term against that reference.

B.2 The carrier

The carrier results of §4.2 concern one-shot parameter reuse: initialising from, perturbing toward, or injecting optimiser state from a past adapter before continued target training. They do not imply that all parameter-side transfer is transient. Frozen modules and inference-time composition remain present throughout optimisation or inference and are therefore persistent by construction. Consistent

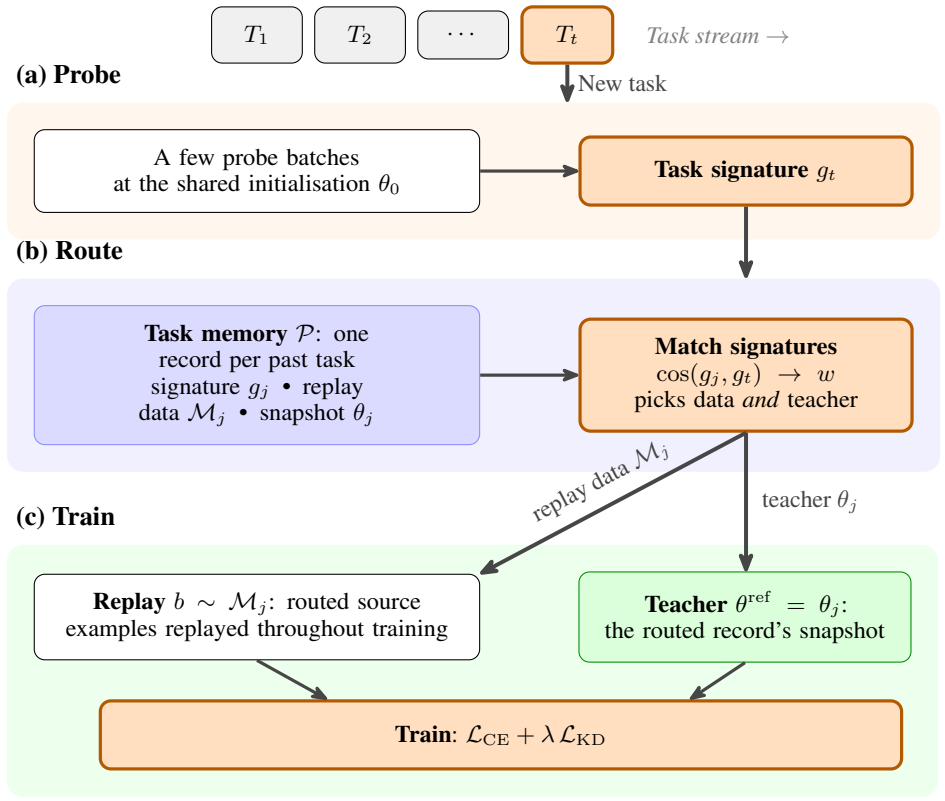


Figure 5: **The Transfer-Selective Replay pipeline**.

with this distinction, continuously anchoring the target adapter toward a source adapter recovers part of the gain from data mixing, whereas one-shot initialisation recovers none. Thus the relevant distinction is not parameters versus data, but whether the transferred signal remains available during target optimisation.

B.3 Replay and anchor ablations

Persistent parameter anchoring (L2-toward-source). Training the target with $L = \text{CE} + \mu\|\theta - \theta_{\text{source}}\|^2$ keeps the source adapter present at every step: a persistent signal whose carrier is *parameters*. On the five oracle pairs (n=5): mean accuracy 0.4526 at μ=0.1 and 0.4410 at μ=1.0, vs. 0.4472 target-data and 0.4547 for data mixing. Persistence is necessary (the one-shot init captures nothing) but not sufficient: the data carrier remains stronger at every μ we probed.

B.4 Routing-policy variants

The routed triple has two knobs: the routing temperature τ (argmax at $\tau \to 0$, uniform pooling at $\tau \to \infty$) and the teacher binding (recent snapshot vs. the routed batch's era snapshot). All single-component variants, same protocol and budgets as Table 3:

Table 5: Routing-policy variants of TSR under the standard stream protocol.

	TRACE-8 (0.5B)			TRACE-8 (7B)			NumGLUE-8 (0.5B)			NumGLUE-8 (7B)		
Method	Overall	Plas	BWT	Overall	Plas	BWT	Overall	Plas	BWT	Overall	Plas	BWT
ER (uniform pooling; $\tau \to \infty$)	$.451_{\pm.031}$	$.427_{\pm.024}$	$+.025_{\pm.020}$	$.656_{\pm.034}$	$.662_{\pm.018}$	$-.006_{\pm.046}$	$.336_{\pm.013}$	$.314_{\pm.011}$	$+.025_{\pm.010}$	$.578_{\pm.012}$	$.579_{\pm.015}$	$-.001_{\pm.019}$
argmax selection ($\tau \to 0$, recent teacher)	$.439_{\pm.033}$	$.430_{\pm.014}$	$+.010_{\pm.019}$	$.675_{\pm.010}$	$.668_{\pm.015}$	$+.007_{\pm.009}$	$.308_{\pm.013}$	$.308_{\pm.012}$	$+.000_{\pm.016}$	$.572_{\pm.011}$	$.577_{\pm.011}$	$-.005_{\pm.007}$
soft routing (recent teacher)	$.459_{\pm.011}$	$.436_{\pm.015}$	$+.023_{\pm.006}$	$.664_{\pm.026}$	$.668_{\pm.010}$	$-.004_{\pm.028}$	$.338_{\pm.010}$	$.316_{\pm.011}$	$+.025_{\pm.018}$	$.572_{\pm.014}$	$.577_{\pm.012}$	$-.005_{\pm.026}$
pool-hybrid (recent teacher)	$.459_{\pm.020}$	$.435_{\pm.007}$	$+.024_{\pm.014}$	$.669_{\pm.014}$	$.670_{\pm.017}$	$-.001_{\pm.019}$	$.333_{\pm.009}$	$.318_{\pm.014}$	$+.018_{\pm.015}$	$.566_{\pm.040}$	$.575_{\pm.016}$	$-.010_{\pm.030}$
routed triple (full TSR)	$\mathbf{.475}_{\pm.030}$	$\mathbf{.439}_{\pm.017}$	$\mathbf{+.035}_{\pm.015}$	$\mathbf{.684}_{\pm.008}$	$\mathbf{.670}_{\pm.012}$	$\mathbf{+.014}_{\pm.011}$	$\mathbf{.353}_{\pm.016}$	$\mathbf{.330}_{\pm.012}$	$\mathbf{+.026}_{\pm.007}$	$\mathbf{.588}_{\pm.017}$	$\mathbf{.582}_{\pm.012}$	$\mathbf{+.007}_{\pm.022}$

B.5 DISTILLATION: THE FULL ABLATION RECORD

This appendix records the distillation audit referenced in §6.3. All experiments: PP15, sequential CL with a single shared LoRA, q, k, v, o projections, 500 steps/task, n=5 seeds, final mean accuracy (AVG) unless noted. The cumulative conclusion: KD in this setting is a stability mechanism, a trajectory regulariser anchored to the shared base. It cannot be converted into a transfer channel by teacher choice, loss reshaping, timing, weighting, or snapshot source.

B.5.1 TEACHER IDENTITY IS IRRELEVANT (ROUTING CEILING)

If KD transferred teacher-specific knowledge, teacher choice should matter. It does not:

Table 6: Teacher-identity ablation for the distillation audit.

Teacher selection	AVG	note
cosine prototype (default)	0.724 ± 0.018	
random teacher per batch	0.715 ± 0.022	ties cosine
per-batch oracle (min-CE teacher)	0.715 ± 0.004	*ceiling*; ties random
latest snapshot only	0.703 ± 0.001	systematically biased, -0.02
anchor to frozen base	0.659 ± 0.008	no snapshot at all
no distillation (sequential FT)	0.584 ± 0.022	

Even the per-batch oracle (the tightest upper bound on routing benefit) adds nothing over a random teacher. The gain over no distillation ($+0.13$) is therefore attributable to having *an* anchor, not to choosing the right one.

B.5.2 SIX LOSS-RESHAPING INTERVENTIONS ALL FAIL

Each intervention was designed to amplify or isolate teacher-specific signal; every one hurts:

Table 7: Loss-reshaping interventions on the distillation term.

Variant	AVG	vs. plain KD (0.724)
amplify teacher LoRA scale (γ=4)	0.493	-0.231
distill on residual (teacher $-$ base)	0.516	-0.208
push student away from base (λ=10^{-3})	0.234	-0.490
selective KL (gate by teacher correctness)	0.602	-0.122
mixup CE	collapse	—
struggle-gate / margin transfer / hidden-MSE	0.638/0.600/0.681	all negative

Removing the base-shared component or repelling the student from the base is catastrophic: direct evidence that the value of KD lies in the base-shared anchor, not in task-specific teacher content.

B.5.3 TIMING AND WEIGHT: REGULARISER SIGNATURES

KD interleaved with CE beats KD applied as a post-hoc phase (endpoint correction), across two initialisation modes:

Table 8: Distillation timing: interleaved versus post-hoc.

Init	KD during CE	KD after CE
QR	0.673 ± 0.015	0.651 ± 0.010
routed mix	0.627 ± 0.016	0.614 ± 0.023

The weight curve is shallow over an order of magnitude (AVG 0.673–0.683 for $\lambda \in [0.1, 2]$), with no narrow optimum and no collapse at either end: the signature of a soft regulariser.

B.5.4 SNAPSHOT SOURCE IS IRRELEVANT

Anchoring to an accumulated merged snapshot versus a routed per-task snapshot makes no measurable difference (QR init: 0.673 vs. 0.673; routed-mix init: 0.627 vs. 0.607, within 1σ): both pull the student toward the same base-shared manifold.

B.5.5 TRANSFER CHANNELS THROUGH DISTILLATION: LOCAL SIGNAL EXISTS, NONE SURVIVES

Beyond loss reshaping, we tested whether snapshots carry extractable transfer along three channels, plus a vision-imported learnable bridge. A zero-step diagnostic first confirmed a real *local* signal: perturbing parameters along a snapshot's update direction predicts loss change with Pearson $r = 0.93$ at small step sizes. None of it survives training:

Table 9: Transfer channels through distillation.

Channel	Method	best AVG
logit	gated multi-teacher KD (top-K by gradient alignment)	0.707
parameter	continuous push along aligned update directions	0.635
feature	relational distillation on batch similarity matrices	0.611
feature	learnable projected bridge ($L = \|h_S - P(h_T)\|^2$)	0.624
—	plain cosine KD baseline	0.724

A control worth recording: under the projected-bridge framework, a *random* teacher scores $+0.014$ *above* the real teacher. This is the spectral-regularisation signature: the bridge's gain, where present, is regularisation rather than transfer.

B.5.6 TEACHER CHOICE IN THE STREAM

Three-task stream numglue_cm $\rightarrow$ cstance $\rightarrow$ numglue_ds (target, N=50, $50/50$ selected replay of cm; n=5). Arms differ only in the KD teacher. "Forgetting" is accuracy change from the pre-target reference.

Table 10: Teacher choice in the three-task stream.

teacher	target acc	cm forget	cstance forget
none	0.151	+0.013	−0.041
recent (pre-task; default)	**0.167**	0.000	−0.037
matched (selected source's era)	0.163	+0.004	−0.043
base (initial adapter)	0.151	−0.017	−0.068
hybrid (matched on replay / recent on target)	0.163	−0.009	**−0.024**

The replayed task (cm) is protected in every arm, because replay data itself does that work. In this *short* stream the recent snapshot is best or tied. **In the full eight-task stream the ranking reverses**: with seven candidate snapshots and substantial drift accumulated in the recent one, matched-era teachers win (TRACE-0.5B, n=5: matched $0.458_{\pm.028}$ > random $0.445_{\pm.024}$ > recent $0.439_{\pm.033}$). The regime dependence is mechanistic: the recent snapshot is the most complete teacher but also the most drifted. As the stream lengthens, drift costs more than completeness buys. This motivates the routed-triple design of §5.

B.5.7 INTERPRETATION AND SCOPE

Distillation for stability has a long lineage in continual learning, from LwF (Li & Hoiem, 2017) and iCaRL (Rebuffi et al., 2017) to replay-based methods such as DER++ (Buzzega et al., 2020). DER++ is particularly related: it anchors replayed examples to logits stored when they enter the buffer, whereas TSR uses a queryable task-era snapshot bound to each routed replay record.

The ablations above sharpen the role of distillation in our regime. Ground-truth supervision is available on every batch, and all teachers are low-rank perturbations of the same frozen base. Teacher-

specific information is locally measurable (r=0.93) but does not survive continued optimisation, whereas repeatedly applying a stable reference consistently improves retention. We therefore interpret distillation primarily as an optimisation anchor in the sequential, frozen-base, low-rank setting studied here. This complements the carrier analysis of §4.2: replay supplies the persistent transfer signal, while distillation supplies stability.

C Experimental Details

C.1 Full-stream setup

Configuration. All methods train a single LoRA adapter (r=16, α=32 on the q, k, v, o projections; AdamW, lr 10^{-4}, batch size 4 at 0.5B / 2 at 7B, max length 384) for 500 steps per task on the N=50 target examples, with a replay buffer of 100 examples per completed task ($N_{\max}$=500 in the §4 audits). TSR: m=10 probe batches, routing temperature τ=0.1, replay ratio α=0.5, distillation weight λ=0.5 at temperature 2. The scale audit of §6.3 uses n=5 seeds per scale (per-point seed s.e.m. .002–.010).

Evaluation. Evaluation uses 500-example per-task test sets; classification tasks report exact-match accuracy, generation tasks evaluation loss.

Stream construction. NumGLUE-8 arranges the eight task types of the established NumGLUE suite (Mishra et al., 2022) as a continual stream: the same construction by which Split-CIFAR derives from CIFAR and TRACE itself from eight existing datasets. Two of its types (numglue-cm/ds) already appear inside TRACE.

Cost of TSR. Signatures are m=10 forward-backward passes per task (all eight of a benchmark's signatures take ∼30 seconds at 0.5B, and picks are scale-stable, so a small proxy model suffices); routing is an inner product; training cost matches standard replay; the buffer stores 100 examples per task (∼3 MB for 15 tasks, versus ∼130 MB for per-task adapter libraries). We regard this routable task record (signature, data subsample, snapshot), rather than the example-level buffer of rehearsal methods or the adapter-only library of parameter-isolation methods, as the right unit of continual memory.

C.2 Per-task results

End-of-stream accuracy per classification task, TRACE-8 at 0.5B (canonical order, mean over 5 seeds; the per-task view behind Table 3).

Table 11: Per-task end-of-stream accuracy on TRACE-8 at 0.5B.

Task	SeqFT	EWC	LwF	SDFT	A-GEM	O-LoRA	DER++	ER	TSR_{arg}	TSR_{soft}	TSR_{hyb}	TSR_{tri}
C-STANCE	.273	.178	.303	.260	.265	.290	.456	.463	.481	.522	.510	.521
FOMC	.214	.215	.167	.202	.390	.178	.075	.580	.502	.553	.536	.541
ScienceQA	.491	.473	.533	.538	.533	.426	.000	.593	.579	.567	.612	.600
NumGLUE-cm	.225	.255	.320	.383	.262	.188	.411	.474	.483	.506	.485	**.552**
NumGLUE-ds	.117	.107	.132	.144	.102	.084	.157	.147	.152	.147	.151	**.159**
AVG	.264	.245	.291	.305	.310	.233	.220	.451	.439	.459	.459	**.475**

C.3 Stream robustness

Task order. Across the greedy best-source curriculum, its reversal, two deterministic reshufflings, and arbitrary reorderings, at both scales varies by at most .03 in Overall accuracy. No-memory baselines vary substantially, tracking the order's *tail* (recency, not the matrix): at 0.5B they swing by +.05 to +.09 under the curriculum, and an arbitrary reshuffle with a similar tail lifts SeqFT even more (+.12). At 7B, where the base barely forgets, the same swings shrink below .04, and on NumGLUE all order effects are within seed noise (§6.3). The one mild sensitivity is argmax selection when a

target precedes its best sources: the transfer matrix is directional, and soft routing removes this by construction.

Backbone. We further evaluate whether the framework generalises across backbone families. Following §4, we first audit headroom on Llama-3.2-1B-Instruct (Grattafiori et al., 2024) and Gemma-3-1B-it (Gemma Team, 2025). The TRACE-8 audit finds open headroom on every task (target-full minus target-data, n=2: $+.045$ to $+.135$ accuracy on Llama and $+.045$ to $+.269$ on Gemma across the five classification targets; the three generation targets improve by 1.8–4.6 in loss), so transfer is measurable on both. Table 12 reports the stream under the exact configuration of Appendix C.1. The qualitative picture of Table 3 is unchanged: replay-based methods dominate; TSR attains the best overall accuracy, plasticity, and backward transfer on both backbones; and positive backward transfer remains confined to the replay-based methods: every rehearsal-free baseline, including the expansion-based GainLoRA, forgets.

Table 12: **Backbone robustness of TSR on TRACE-8** under the standard sequential protocol (Llama-3.2-1B and Gemma-3-1B; mean $\pm$ std over 5 seeds).

	TRACE-8 (Llama-3.2-1B)			TRACE-8 (Gemma-3-1B)		
Method	Overall	Plas	BWT	Overall	Plas	BWT
SeqFT	$.420_{\pm.028}$	$.458_{\pm.018}$	$-.038_{\pm.022}$	$.154_{\pm.018}$	$.319_{\pm.012}$	$-.164_{\pm.023}$
LwF	$.422_{\pm.027}$	$.459_{\pm.014}$	$-.038_{\pm.015}$	$.241_{\pm.028}$	$.319_{\pm.012}$	$-.078_{\pm.020}$
GainLoRA	$.430_{\pm.019}$	$.438_{\pm.019}$	$-.008_{\pm.009}$	$.279_{\pm.013}$	$.301_{\pm.010}$	$-.022_{\pm.012}$
ER	$.494_{\pm.019}$	$.470_{\pm.012}$	$+.024_{\pm.020}$	$.354_{\pm.011}$	$.337_{\pm.012}$	$+.018_{\pm.017}$
TSR	$\mathbf{.519}_{\pm.015}$	$\mathbf{.481}_{\pm.012}$	$\mathbf{+.037}_{\pm.011}$	$\mathbf{.381}_{\pm.014}$	$\mathbf{.351}_{\pm.012}$	$\mathbf{+.030}_{\pm.010}$

C.4 Reproducibility

All experiments use publicly available benchmarks and base models. Complete training configurations are given in Appendix C.1, with experiment-specific protocols, budgets, and seed counts reported alongside the corresponding results. All reported statistics use the complete set of designated seeds for each experiment. An anonymised implementation reproducing the core TRACE-8 experiments is provided in the supplementary material.