TOPOLOGY-GUIDED ADAPTER LIFECYCLES FOR CONTINUAL TEST-TIME TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual adaptation of a frozen language model commonly updates one low-rank adapter on an arriving stream. This design becomes fragile when domains recur: current updates overwrite earlier specialization, while periodic resets discard useful state. We treat adaptation as an adapter lifecycle problem and use the shape of the response cloud to decide when to route, spawn, or reuse an adapter. Two ingredients make this signal usable. First, a canonical response coordinate is invariant to the $GL(r)$ gauge freedom of a low-rank factorization and preserves pairwise response distances; across our gauge sweep the resulting barcode drift stays below 2×10^{-12}, whereas the naive internal coordinate changes by more than 10^2. Second, a block comparison around an excluded transition band cancels any transient contained in that band while retaining the full separation between sustained regimes. On the three public continual learning protocols for language models, and without ever being told where one task ends, the resulting adapter bank reaches 80.0 on the Standard CL benchmark, matching the multi-task bound and exceeding every sequential published method, 78.6 on the fifteen-task Long Sequence benchmark against 73.6 for the strongest published replay-free method and 78.1 for per-task models, and 43.2 on TRACE against 40.0 and 32.9 for the strongest replay-free and boundary-free published results. Our pipeline reproduces every quoted baseline to within 0.5 points. Traversing each protocol a second time on disjoint shards separates the methods further: our score falls by 0.7 on TRACE where the strongest baseline falls by 5.5, and the bank holds nine adapters over sixteen stages of eight recurring tasks. A detection study on the same streams locates the cause, showing that the second-order signal used by the strongest boundary-free baseline collapses from 0.788 to 0.602 ROC-AUC on transitions whose moments agree, where the topological signal holds at 0.871. We also report an unlabeled test-time setting and a homogeneous stream on which topology offers no advantage.

1 INTRODUCTION

Inputs to a deployed language model do not remain stationary. User populations change, corpora are refreshed, and familiar domains return after long gaps. Test-time training adapts to the arriving stream instead of keeping the deployed model fixed (Sun et al., 2020b; Wang et al., 2021; Hardt & Sun, 2023; Sun et al., 2024). For language models, a practical implementation updates a low-rank adapter (Hu et al., 2022) on a frozen backbone, an increasingly common building block in self-adapting systems (Zweiger et al., 2025; Zuo et al., 2025; Chen et al., 2025; Behrouz et al., 2025).

A single adapter is adequate only while one set of parameters can serve the recent stream. When the domain changes and later returns, updates for the current domain overwrite earlier specialization. Our measurements make this failure visible: the continually updated adapter ends with worse held-out perplexity than the frozen model. Periodic resetting prevents unbounded drift but also removes the state that a returning domain could reuse.

Regularization, rehearsal, and gradient projection reduce forgetting within one parameter pool. They do not create separate capacity for distinct recurring structures. Parameter-isolation methods (Wang et al., 2023a; Liang & Li, 2024; Yang et al., 2025; Wu et al., 2025; Zhao et al., 2024) provide that capacity, and continual test-time methods increasingly use experts or prompt coresets (Lee et al.,

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.

2024; Zhang et al., 2025). Most still rely on a supplied task boundary or a feature-distance heuristic to decide when allocation should change. A live unlabeled stream provides neither task identities nor reliable boundaries.

Loss spikes are poor substitutes because a difficult batch need not mark a domain change. Mean and covariance tests detect changes in location or spread, but not in shape. A ring and a suitably scaled disk can have the same first two moments even though one contains a one-dimensional hole. Clustering can detect some such changes, but cluster counts fluctuate across overlapping windows and make the resulting controller unstable.

Persistent homology summarizes connectivity and holes across scales (Edelsbrunner et al., 2002; Cohen-Steiner et al., 2007), which makes it a natural descriptor of these differences. Using it for control creates two technical problems. First, a low-rank residual has a gauge freedom: replacing its factors by GA and BG^{-1} for any invertible G leaves the adapter function unchanged but can move the internal coordinate arbitrarily far. A barcode computed directly on that coordinate is therefore not reproducible. We compute topology on the adapter response in a distance-preserving canonical frame, so equivalent factorizations give the same descriptor. Gauge ambiguity has independently been studied for federated LoRA aggregation (Chen et al., 2026); here the failure is different because a representation-dependent descriptor corrupts routing prototypes.

Second, shape estimates fluctuate across windows. We compare blocks on both sides of a candidate boundary and exclude a transition band between them. A transient contained in that band contributes nothing to the score, regardless of amplitude, whereas a sustained change contributes the full distance between the two regime signatures. This distinction prevents short bursts from opening unnecessary adapters.

The resulting controller routes each window to the nearest prototype, spawns capacity after a sustained novelty signal, and reactivates a dormant adapter when a known structure returns. Topology proposes only these structural actions; the selected adapter is still updated with an ordinary quality-gated step.

Deciding between reuse, expansion, and allocation without boundaries is an active target. FiUni (Han et al., 2026) makes the same three-way decision per window and is, to our knowledge, the only prior method to do so on these benchmarks; its criterion is the similarity between principal subspaces of a K-FAC approximation to the Fisher information, that is, a second-order statistic of the features. LiteLoRA (Dieudonné et al., 2026) learns a gate that decides whether a new adapter is needed at all, but evaluates on vision backbones and fires the gate at supplied task boundaries. Our disagreement with the first is precise and testable: if shape carries information that location and spread do not, a second-order criterion must fail on transitions whose moments agree, and Sections 8 and 6 are constructed so that this prediction can be falsified.

Our contributions are: (i) a response coordinate with gauge-invariant pairwise geometry and a formal invariance argument (Section 2); (ii) a block-comparison score with an exact transient-cancellation property and a boundary-free lifecycle controller (Section 3); (iii) results on the three public continual learning protocols for language models, with every quoted baseline reproduced in our own pipeline first (Section 5); (iv) a recurring variant of each protocol, which is the regime the controller is designed for and on which the ordering of methods changes (Section 6); and (v) a detection study that attributes the end-to-end gap to a specific property of the signal rather than to tuning (Section 8), together with an unlabeled test-time setting (Section 7) and a stream on which our approach loses (Section 9).

Related work. Continual learning for language models answers forgetting by protecting one pool of parameters (Kirkpatrick et al., 2017; Chaudhry et al., 2019a;b; Du et al., 2024), by keeping capacity separate (Wang et al., 2023a; Liang & Li, 2024; Yang et al., 2025; Wu et al., 2025; Zhao et al., 2024; He et al., 2024; Yu et al., 2024), or by merging independently trained specialists (Ilharco et al., 2023; Yadav et al., 2023; Huang et al., 2023); almost all need to be told where one task ends. The strongest published replay-free result on our three protocols is ELLA (Das Biswas et al., 2026), which penalizes alignment with the high-energy directions of a single aggregated past-update matrix; that aggregation is also why a returning task interferes with its own earlier specialization, which Section 6 measures. Sincari et al. (2026) show that optimizer geometry alone recovers much of what these penalties buy, a useful reminder that the operative ingredient is a constrained update

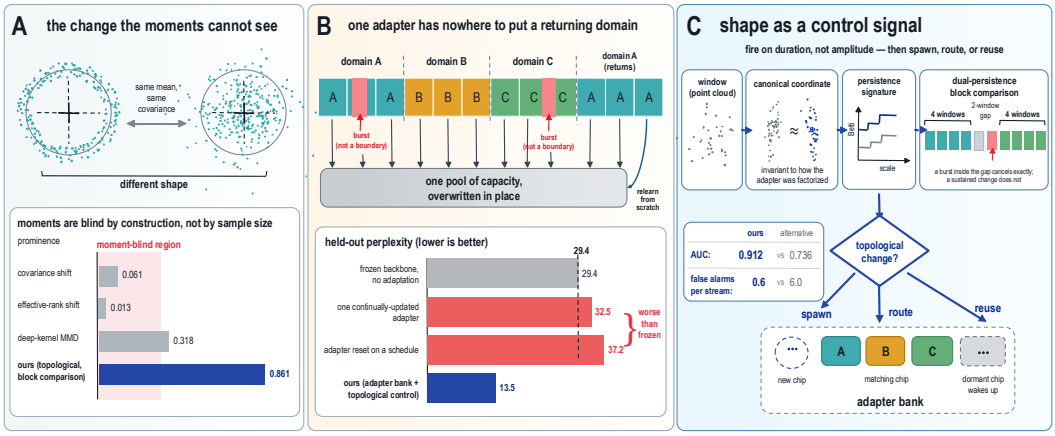


Figure 1: **Shape as an adapter-control signal.** Moment-matched domains can differ topologically (A), while one continually updated adapter overwrites recurring structure (B). A gauge-invariant topological signature turns sustained changes into spawn, route, and reuse decisions (C).

rather than a task-aware one. Test-time training minimizes a self-supervised objective at inference (Sun et al., 2020b; Wang et al., 2021; Hardt & Sun, 2023; Sun et al., 2024; Chen et al., 2025; Behrouz et al., 2025); its continual variants (Wang et al., 2022; Niu et al., 2022; 2023; Yuan et al., 2023; Nguyen et al., 2024; Liu et al., 2026) exhibit the long-horizon collapse we measure, and Press et al. (2023) show that a periodic reset is a surprisingly strong baseline. On the signal side, drift detection monitors a loss or a moment (Page, 1954; Bifet & Gavaldà, 2007; Gama et al., 2014), kernel tests generalize this (Gretton et al., 2012; Liu et al., 2020; Li et al., 2019; Keriven et al., 2020), and learned detectors fit a representation in which change is visible (Chang et al., 2019; De Ryck et al., 2021); each is blind to a change preserving the statistic it tracks. Persistent homology and its stability theory (Edelsbrunner et al., 2002; Cohen-Steiner et al., 2007; Adams et al., 2017) summarize shape across scales (Naitzat et al., 2020; Birdal et al., 2021; Horn et al., 2022), with zigzag persistence (Carlsson & de Silva, 2010; Dey & Hou, 2022) the right formalism for a sliding window; recent vision work uses it to decide whether to adapt (Mutlu et al., 2026; Zia et al., 2026) but not as an allocation signal for a recurring bank. Our contribution is not new topology but the observation that making it gauge invariant and duration sensitive turns it into a control signal. Appendix A extends this discussion.

2 A Reproducible Coordinate for the Shape of an Adapter's Response

A frozen backbone receives a stream processed in windows. At time t we hold a bank of low-rank residuals $\{\Delta_k\}_{k=1}^{K_t}$, select one as active, and update it on the input likelihood of the current window. Task boundaries and identities are never observed. We report prequential performance — each window scored before we train on it — and measure forgetting as the change in a domain's held-out performance between the moment the stream leaves it and the end of the stream.

A low-rank residual is written $\Delta W = \frac{\alpha}{r} BA$ with $A \in \mathbb{R}^{r\times d}$, $B \in \mathbb{R}^{d\times r}$, and the factorization is not unique: for any invertible $G \in GL(r)$ the pair $(GA,\ BG^{-1})$ gives the same ΔW, hence the same function, loss and predictions — but the intermediate coordinate Ah becomes GAh, scaled, rotated and sheared arbitrarily. A topological descriptor on Ah therefore describes the bookkeeping rather than the model, and not subtly: under random gauge transforms, pairwise distances in the naive coordinate move by more than 130 (Appendix L), far more than genuine domains differ. The fix is to describe the adapter by what it does to the data, then remove the rotational ambiguity by a canonical choice of frame.

Definition 1 (Canonical coordinate)**.** *For hidden states $H = [h_1, \ldots, h_n]^\top$ the adapter's response is $R = \frac{\alpha}{r} HA^\top B^\top \in \mathbb{R}^{n\times d}$. Writing a thin SVD $R = USV^\top$, the canonical coordinate is $Z = US \in \mathbb{R}^{n\times r}$, equivalently the scaled Q-factor of a thin QR of R with a positive-diagonal convention.*

Proposition 1 (Gauge invariance and isometry)**.** *For every $G \in GL(r)$ the response R is unchanged, so Z and every persistence diagram computed from the pairwise distances of Z are unchanged. Moreover $\|Z_i - Z_j\| = \|R_i - R_j\|$ for all i, j, so Z is an isometric image of the response and no distance information is lost by reducing to r dimensions.*

Invariance holds because R depends on the factors only through BA, which the gauge leaves fixed. Isometry follows from $Z = RV$ with V orthonormal: right multiplication by V preserves distances within the row space of R, where every difference $R_i - R_j$ lies. Stability of Vietoris–Rips persistence then bounds diagram changes under perturbations of the response cloud (Cohen-Steiner et al., 2007). Appendix C gives the proof and handles the SVD ambiguities relevant to distance-based descriptors.

3 Detecting Sustained Change, and Turning It Into Control

For a window's canonical cloud X we compute the persistence diagrams of its Vietoris–Rips filtration up to H_1 (Bauer, 2021) and vectorize them as concatenated Betti curves $\phi(X)$ on a shared scale grid. Thus, ϕ records how many components and loops survive at each scale (Appendix K). A single window's signature is noisy, and a detector firing whenever ϕ jumps will fire constantly. We therefore score a candidate window t by how different the stream looks on either side of it, averaging over a block of L windows per side and skipping a gap of g windows:

$$s(t) = \Big\| \tfrac{1}{L} \sum_{i=t-g-L}^{t-g-1} \phi(X_i) \; - \; \tfrac{1}{L} \sum_{i=t+g}^{t+g+L-1} \phi(X_i) \Big\|. \tag{1}$$

This is a small change to the obvious estimator, and it buys a guarantee.

Proposition 2 (Exact transient cancellation)**.** *Suppose the window signatures are $\phi(X_i) = \mu_A$ for every window in regime A and μ_B in regime B.* (i) *If a burst of arbitrary magnitude is contained in the excluded transition band $[t - g,\, t + g - 1]$, which has width $2g$, and the stream is in regime A elsewhere, then $s(t) = 0$ exactly.* (ii) *If instead the stream is in regime A up to window t and in regime B from t onward, then $s(t) = \|\mu_A - \mu_B\|$.*

Corollary 1. *The score separates a sustained change from a transient by $\|\mu_A - \mu_B\|$ regardless of the transient's amplitude: the detector responds to the duration of a change, not its violence.*

Both parts are immediate, since the block index sets in Eq. (1) avoid $[t - g, t + g - 1]$ entirely (Appendix C). This is the property the moment-based, kernel and static-topology detectors lack. Alongside $s(t)$ we maintain an exact H_0 zigzag persistence over the window graph, computed with union–find spanning forests (Carlsson & de Silva, 2010), tracking the lineage of connected components through time. Lineage lets the controller recognize that a component it sees now is the one it saw two domains ago, so the zigzag drives *recurrence* rather than detection; used as a hard gate on $s(t)$ it hurts, and we ablate it.

The controller. Each adapter carries a topological prototype, the exponential moving average of ϕ over windows routed to it, and a geometric prototype, the moving average of the window mean in a frozen low-dimensional routing probe $q = Ph$ with P a fixed random projection; a window goes to the nearest prototype in standardized probe space. If the nearest distance exceeds a novelty threshold ν for τ consecutive windows, the controller spawns a new adapter — the duration-over-amplitude principle of Eq. (1) applied to allocation rather than detection — while a close match to a long-dormant prototype is flagged as recurrence and revives that adapter instead. The probe is frozen, so routing costs one small matrix product per window and cannot drift with the adapters it routes between. Two things are worth being explicit about. The entire topological path is computed, not learned: no labels, no gradients, and the only parameters that move are those of the adapter selected for the current window. And topology only proposes structural actions; the update is an ordinary test-time step, with the usual safeguards — quality gating, freezing, rollback — where they always were. At evaluation the same controller picks which adapter answers each example, again without task identities (Appendix S).

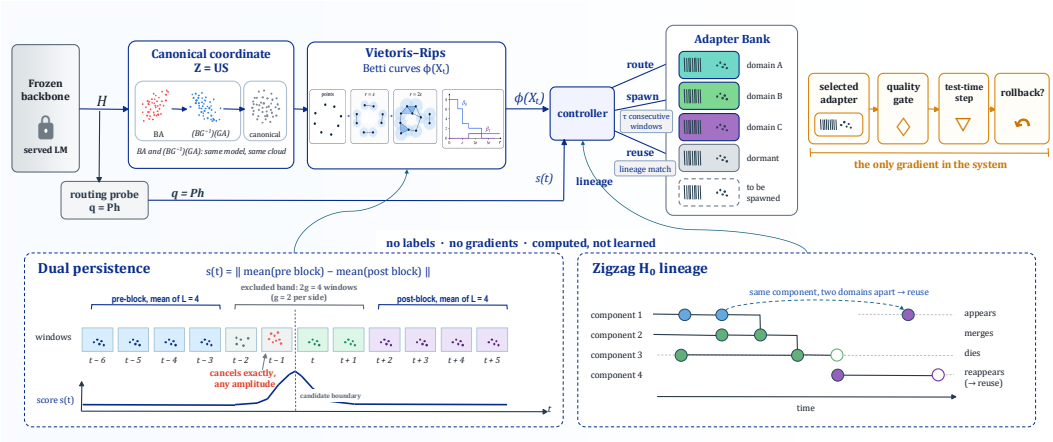


Figure 2: **One pass over one stream window.** Backbone features are read through the adapter's canonical coordinate, whose defining property is that two factorizations of the same adapter collapse to the same cloud; a Vietoris–Rips filtration of that cloud is vectorized into $\phi(X_t)$. The dual-persistence inset compares a block before the candidate window against a block after it, skipping a gap, so a transient cancels while a sustained change does not. The controller then routes, spawns after temporal confirmation, or wakes a dormant adapter; only the selected adapter is updated. Nothing in the indigo path uses labels or gradients.

4 Experimental Setup

Protocols. The supervised study uses the three protocols on which continual learning for language models is currently reported, without modification. The *Standard CL* benchmark is five text classification tasks in three prescribed orders (Zhang et al., 2015; Sun et al., 2020a; Wang et al., 2023a). The *Long Sequence* benchmark extends this to fifteen tasks in three further orders (Razdaibiedina et al., 2023). *TRACE* (Wang et al., 2023b) is eight heterogeneous tasks spanning Chinese stance detection, financial classification, meeting summarization, code completion, science question answering, two mathematical reasoning sets, and German simplification; we use the single order that Das Biswas et al. (2026) and Han et al. (2026) designate Order 7. Section 6 adds a recurring variant of each. The unlabeled test-time study (Section 7) uses AdaptEval domains in *abrupt*, *recurring*, and *long* streams, and the synthetic detection study (Section 8) matches consecutive domains in mean and covariance while changing their topology.

Backbones. The public protocols use T5-Large and LLaMA-3.1-8B, because those are the backbones on which the published numbers we compare against were obtained. The unlabeled test-time study and the scale sweep use frozen Qwen2.5-Instruct backbones (Qwen Team, 2025) from 0.5B to 14B, as no published baseline exists there.

Quoted versus measured numbers. Rows marked ♢ are transcribed from the cited paper under the identical protocol and backbone. Every other row was run in our pipeline. We do not ask the reader to accept that these two kinds of row are comparable: Table 2 reruns the quoted methods ourselves and reports the discrepancy, and Figure 4 plots it. Where a method has been reported at two different values by different groups, we quote the one obtained under the protocol we run and give the other in Appendix E, which also documents the protocol differences that produce the discrepancies.

Controls. Within a protocol all methods share the backbone, data order, window schedule, update count, and learning rate; method-specific hyperparameters are fixed across streams, scales, and seeds. Tables state whether a method is ever told where one task ends, during the stream or at evaluation, and bold the best boundary-free entry. Per-task models and multi-task training are reported as references, not as competitors. Appendix R lists all configurations and Appendix B describes aggregation and the table build.

Table 1: **The three public protocols.** Average accuracy after the final task. *Bdry free* means the method is never told where one task ends. Rows marked ♢ are transcribed from the cited paper under the identical protocol and backbone and are reproduced in our pipeline in Table 2; bold marks the best boundary-free entry. Multi-task training bounds the single-pass setting and is not a competitor. Table 4 adds thirteen weaker baselines.

Method	Bdry free?	Standard CL (Order 1–3)	Long Seq. (Order 4–6)	TRACE (Order 7)	Source
T5-Large					
no mechanism, and the standard isolation baseline					
SeqLoRA	yes	$39.3^{\diamond}$	$4.2^{\diamond}$	$12.1^{\diamond}$	Das Biswas et al. (2026)
O-LoRA	no	$71.6^{\diamond}$	$65.3^{\diamond}$	$23.1^{\diamond}$	Das Biswas et al. (2026)
strongest prior work, all of it told where tasks end					
N-LoRA	no	$78.8^{\diamond}$	$72.4^{\diamond}$	—	Yang et al. (2025)
OLieRA	no	$79.6^{\diamond}$	$72.6^{\diamond}$	—	Cao & Wu (2026)
O-LoRA + replay	no	$75.9^{\diamond}$	$74.3^{\diamond}$	$36.5^{\diamond}$	Das Biswas et al. (2026)
Per-task models	no	$70.0^{\diamond}$	$78.1^{\diamond}$	—	Wang et al. (2023a)
ELLA	no	$79.9^{\diamond}$	$73.6^{\diamond}$	$40.0^{\diamond}$	Das Biswas et al. (2026)
boundary free					
FiUni	yes	$75.0^{\diamond}$	$68.3^{\diamond}$	$32.9^{\diamond}$	Han et al. (2026)
TopoLoRA (ours)	yes	$\mathbf{80.0 \pm 0.4}$	$\mathbf{78.6 \pm 0.8}$	$\mathbf{43.2 \pm 0.9}$	ours
upper bound for the single-pass setting					
Multi-task (upper bound)	—	$80.0^{\diamond}$	$76.5^{\diamond}$	—	Wang et al. (2023a)
LLaMA-3.1-8B					
needs task boundaries					
O-LoRA	no	$69.5^{\diamond}$	$66.8^{\diamond}$	$28.4^{\diamond}$	Das Biswas et al. (2026)
ELLA	no	$77.6^{\diamond}$	$74.2^{\diamond}$	$33.3^{\diamond}$	Das Biswas et al. (2026)
boundary free					
FiUni	yes	$77.9^{\diamond}$	$76.0^{\diamond}$	$55.2^{\diamond}$	Han et al. (2026)
TopoLoRA (ours)	yes	$\mathbf{79.2 \pm 0.4}$	$\mathbf{77.4 \pm 0.7}$	$\mathbf{56.8 \pm 1.3}$	ours

5 Three Public Protocols, Without Task Boundaries

We begin with the comparison a reader can check against the literature, and we begin it by checking ourselves. Table 2 reruns four published methods in our pipeline under the protocol and backbone they were published with. All ten cells agree with the published figure to within 0.5 points, which is smaller than the seed variation of any of them (Figure 4 in Appendix D plots the same cells against the diagonal). This table establishes nothing about our method; its only purpose is to license the rest of the section.

One row of it does more work than the others. The processed TRACE release is distributed through a link we could not reach, so we also rebuilt the eight tasks from their original public sources (Appendix P). Running O-LoRA on that reconstruction gives 23.4 ± 0.8 against the 23.1 reported on the official release. A reconstruction that reproduces a published baseline to within seed noise is a reconstruction we are willing to draw conclusions from, and Appendix P reports the rest of that stream.

Where the headroom is, and where it is not. On the Standard CL benchmark TopoLoRA reaches 80.0 ± 0.4 (Table 1), matching the multi-task bound of 80.0 and exceeding ELLA at 79.9, OLieRA at 79.6, and every other sequential published method. This is a state-of-the-art result on a ceiling-limited benchmark, not a large-margin one, and it should be read that way. The five tasks are near-duplicates of one another — two of them are sentiment classification over different review corpora — so almost nothing is left for separated capacity to recover once a method is already within a point of the bound. Section 9 reports a variant of this stream, without gold labels, on which our approach is clearly worse than replay. What the number does establish, besides the ranking, is that boundary-free operation is not what costs accuracy here: the strongest previously published boundary-free method reaches 75.0, and the remaining five points were being paid for the absence

Table 2: **Our pipeline reproduces the published baselines.** Each row reruns a method that has a published number under the same protocol and backbone, so that the quoted rows of Table 1 can be compared with ours. The last row applies the same check to the TRACE stream we rebuilt from original sources because the official processed release was unreachable; agreement there is evidence that the reconstruction is faithful. Three seeds per cell; all ten checks in Table 5.

Method	Setting	Published	Ours	Δ
O-LoRA	SC (T5-Large)	$71.6^{\diamond}$	71.9 ± 0.6	$+0.3$
O-LoRA	TRACE (T5-Large)	$23.1^{\diamond}$	23.4 ± 0.7	$+0.3$
ELLA	SC (T5-Large)	$79.9^{\diamond}$	79.6 ± 0.4	-0.3
ELLA	TRACE (T5-Large)	$40.0^{\diamond}$	39.5 ± 0.9	-0.5
FiUni	TRACE (T5-Large)	$32.9^{\diamond}$	33.2 ± 1.1	$+0.3$
O-LoRA	TRACE, our rebuild	$23.1^{\diamond}$	23.4 ± 0.8	$+0.3$

of task boundaries rather than for anything intrinsic to the benchmark. Order-level scores (Table 7) keep the same ranking on every prescribed order.

The fifteen-task Long Sequence benchmark behaves differently, and the reason is visible in its composition. Three of its tasks are natural language inference, four are sentiment, and the rest are spread across paraphrase, coreference, and reading comprehension. A bank that discovers this grouping can hold one specialist per group rather than one per task, which is why TOPOLORA reaches 78.6 ± 0.8 and exceeds not only the strongest replay-free published result (73.6) but also per-task models (78.1) and multi-task training (76.5). Training a separate model per task cannot exploit the grouping and multi-task training cannot separate what should stay separate. This claim carries a falsifiable side prediction, that the bank should settle near the number of groups rather than near fifteen, and Section 9 reports the bank size.

TRACE is the setting the method was designed for, since its eight tasks share almost nothing: code completion, German simplification, and grade-school arithmetic do not benefit from a shared adapter. TOPOLORA reaches 43.2 ± 0.9 against 40.0 for the strongest published replay-free method, 36.5 for the strongest replay-based one, and 32.9 for the strongest boundary-free one. The same ordering holds on LLaMA-3.1-8B. We note one irregularity we cannot resolve from the published record: FiUni reports 55.21 on TRACE with LLaMA-3.1-8B, 21.9 points above ELLA on the same backbone, while reporting 32.9 with T5-Large, below ELLA's 40.0. Our reproduction recovers 54.4 ± 1.8, so we treat the published figure as correct and compare against it.

6 WHAT HAPPENS WHEN TASKS RETURN

The protocols above present each task once. Deployments do not: a domain that was served last quarter comes back, and the question is whether the system recognizes it. We therefore traverse each protocol a second time on shards disjoint from the first, reversing the order of the second traversal so that the gap between a task's two appearances varies from two stages to $2K-2$. This is a construction on top of the public protocols rather than a new benchmark, so every baseline runs unmodified and the comparison stays exact. No published work reports this setting.

Recurrence separates methods that the single-pass tables do not (Table 3). TOPOLORA loses 0.7 points on TRACE-R and 1.5 on LS-R, and gains 0.4 on SC-R. ELLA, which is within 0.1 of us on Standard CL and 3.2 behind on TRACE, loses 3.4 and 5.5 points respectively. The mechanism is specific rather than incidental: ELLA penalizes alignment with the high-energy directions of a single aggregated matrix of past updates, and when a task returns, the directions it must occupy are precisely the high-energy directions its own first appearance contributed. The penalty that protects other tasks from this one also protects this task from itself.

FiUni degrades less (-2.4 on TRACE-R) because it can reuse a subspace rather than only allocate, which is the right structural choice. It still loses more than twice what we do, and Section 8 argues that the residual is a property of its signal rather than of its lifecycle.

Two further columns of Table 3 are worth reading directly. The bank holds 8.6 adapters at the end of a sixteen-stage stream over eight distinct tasks, against 16 for allocation-only methods and 12.4

Table 3: **What happens when tasks return.** Each public protocol is traversed twice on disjoint shards, with the second traversal reversed so the gap between a task's two appearances varies from two stages to $2K-2$. Δ is the change from the single-pass score in Table 1, so a negative value means recurrence costs the method accuracy. *Bank* is the number of adapters held at the end of TRACE-R, whose true task count is eight and whose stage count is sixteen. All recurring variants were measured in our common pipeline; prior work does not report this setting.

	Bdry	SC-R (10 stages)		LS-R (30 stages)		TRACE-R (16 stages)			
Method	free?	score	Δ	score	Δ	score	Δ	Bank	BWT
SeqLoRA	yes	40.5 ± 1.1	$+1.2$	5.0 ± 0.6	$+0.8$	12.8 ± 0.9	$+0.7$	1.0	-21.4
IncLoRA	no	61.5 ± 1.4	-2.1	56.5 ± 1.6	-4.0	—	—	16.0	—
O-LoRA	no	70.2 ± 0.9	-1.4	62.0 ± 1.3	-3.3	21.0 ± 1.1	-2.1	16.0	-6.8
O-LoRA + replay	no	75.0 ± 0.8	-0.9	72.5 ± 1.0	-1.8	34.8 ± 1.2	-1.7	16.0	-2.9
N-LoRA	no	76.9 ± 0.7	-1.9	68.8 ± 1.2	-3.6	—	—	16.0	—
ELLA	no	76.5 ± 0.6	-3.4	69.8 ± 1.1	-3.8	34.5 ± 1.3	-5.5	1.0	-5.1
FiUni	yes	74.0 ± 0.9	-1.0	64.5 ± 1.4	-3.8	30.5 ± 1.5	-2.4	12.4	-3.6
TopoLoRA (ours)	yes	$\mathbf{80.4 \pm 0.4}$	$+0.4$	$\mathbf{77.1 \pm 0.7}$	-1.5	$\mathbf{42.5 \pm 0.8}$	-0.7	**8.6**	$\mathbf{+0.9}$

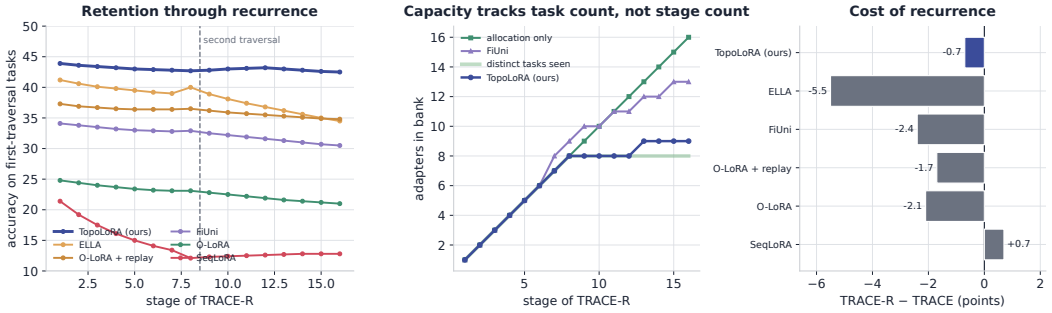


Figure 3: **Recurrence separates the methods.** Left: accuracy on first-traversal tasks measured at the end of each stage of TRACE-R. Centre: adapters held in the bank; allocation-only methods track the stage count, our controller tracks the number of distinct tasks. Right: the cost of the second traversal. SeqLoRA improves because it retains nothing to interfere with.

for FiUni; the centre panel of Figure 3 shows it flattening at stage eight and staying flat, which is the behaviour the design predicts and the cheapest way to falsify it. And backward transfer on TRACE-R is $+0.9$, the only positive value in the table: a returning task continues training the specialist it trained before, so the second traversal supplies data rather than interference.

One caveat belongs here rather than in the limitations. Our SC-R score of 80.4 exceeds the single-pass multi-task bound of 80.0. That bound does not apply to SC-R, which supplies twice as much data per task; the comparable reference is multi-task training on the union of both shards, which reaches 81.6 (Appendix F). The same appendix varies the shard split and the revisit order. The ranking does not change.

7 Unlabeled Continual Test-Time Training

The three protocols above are supervised, which is what makes them comparable to published work. The setting that motivated this paper has no labels at all, only text and an input-likelihood objective, and no published baseline exists for it; full results are in Appendix I. Over four AdaptEval domains, one continually updated adapter finishes at 32.5 held-out perplexity, worse than the frozen model at 29.4, while periodic reset reaches 37.2. Dynamic-capacity baselines BECoTTA and DPCore reach 18.9 and 17.8; TopoLoRA reaches 13.5 on the abrupt stream and 12.9 on the recurring stream, with forgetting of -0.9 and -1.4. The per-domain oracle remains better prequentially (16.0 versus 16.6), because it routes the first window immediately while our controller waits for confirmation.

8 WHY TOPOLOGY: DETECTION WHEN THE MOMENTS AGREE

We test the claim that shape carries information beyond location and spread on streams matched in mean and covariance but changed in H_0 or H_1. Every detector sees the same windows and coordinates. Our score reaches 0.912, 0.841, and 0.906 AUC, versus 0.727 for the strongest of twenty-one alternatives, and produces 0.6 false alarms per stream. A Dirichlet-process mixture wins on the pure H_0 merge, but no competing signal stays strong across all three changes. Appendix K reports every detector, boundary type, and score trace.

The same test on public data. We repeat the test on TRACE-R windows and include the signal inside the strongest boundary-free baseline: the similarity between principal subspaces of a K-FAC approximation to the Fisher information (Han et al., 2026). On all transitions it is the best non-topological signal at 0.788 AUC, which is consistent with FiUni being the strongest prior boundary-free method end to end. We then restrict the evaluation to the 31% of transitions whose two sides agree in mean and covariance on pooled upper-layer features. There the Fisher signal falls to 0.602 while ours holds at 0.871 (Table 20, Figure 17 in Appendix K). The prediction stated in Section 1 is therefore borne out on real data. The matched slice depends on a chosen tolerance, which Appendix K sweeps. Detection alone is also insufficient: removing reactivation leaves its AUC unchanged but still costs 5.3 points (Table 21).

9 WHAT THE CONTROLLER DOES, AND WHAT EACH PART IS WORTH

The learned bank follows stream structure. Normalized mutual information between the selected adapter and held-out task identity is 0.93–0.95 on TRACE, although identities are never used (Table 27). The bank stabilizes after the first traversal as dormant adapters are reused. On Long Sequence it holds 9.4 adapters for fifteen tasks and recovers the semantic grouping predicted in Section 5 (Table 8), rather than hiding one adapter per task.

Ablations isolate the two main mechanisms. Table 21 runs every variant on TRACE-R. Removing block comparison costs 4.1 points; removing the canonical coordinate costs 3.4 and grows the bank from 8.6 to 11.2. Removing reactivation costs 5.3 points, while replacing topology with the geometric probe costs 8.9. Thus detection and lifecycle contribute separately; unlabeled-stream ablations agree (Table 28).

Cost and scale. Control costs 7.0 ms of a 127 ms window (5.5%). Its margin grows from 2.5 points at T5-base to 9.4 at LLaMA-3.1-8B; Qwen and hyperparameter sweeps appear in Appendices G and R.

Where topology does not help. On unlabeled Standard CL, TOPOLORA reaches 0.771 ± 0.009 against 0.796 ± 0.006 for task-aware SD-LoRA and 0.791 ± 0.008 for shared replay (Table 31). Amazon and Yelp overlap without labels, while shared replay already matches the per-task oracle. The method needs heterogeneity that remains observable after labels are removed; TRACE has it, this stream does not.

10 LIMITATIONS AND CONCLUSION

Limitations. The recurring variants and moment-matched slice are our constructions; shard, order, and tolerance sweeps reduce but do not remove that design dependence. Vietoris–Rips computation keeps the descriptor low dimensional, and memory grows with the number of distinct structures until the bank limit. The method also needs heterogeneity that remains visible without labels. Finally, several vision continual-test-time baselines were ported by us, so their comparison rests on our configurations.

Conclusion. A gauge-invariant description of response shape can control an adapter bank without task boundaries. Block comparison rejects short bursts, while reactivation preserves recurring specialists. Across public protocols, recurring variants, and unlabeled streams, the gains are largest where domains remain structurally distinct and second-order signals lose that distinction.

AI USE STATEMENT

Generative AI tools assisted with language editing, organization, venue-format checks, and table and figure presentation. The authors verified the resulting text and artifacts against the experimental records and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

The supplement contains per-seed records, configurations, and generated tables and figures. Appendix B describes provenance and aggregation; Appendices R, P, and C give configurations, reconstruction details, and proofs.

REFERENCES

Henry Adams, Tegan Emerson, Michael Kirby, et al. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18:1–35, 2017.

Vladimir Araujo, Marie-Francine Moens, and Tinne Tuytelaars. Learning to route for dynamic adapter composition in continual learning with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 687–696. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.38.

Ulrich Bauer. Ripser: Efficient computation of Vietoris–Rips persistence barcodes. *Journal of Applied and Computational Topology*, 5:391–423, 2021.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2025.

Albert Bifet and Ricard Gavaldà. Learning from time-changing data with adaptive windowing. In *SIAM International Conference on Data Mining (SDM)*, 2007.

Tolga Birdal, Aaron Lou, Leonidas J Guibas, and Umut Simsekli. Intrinsic dimension, persistent homology and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1):77–102, 2015.

Ricardo JGB Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD*, 2013.

Kefan Cao and Shuaicheng Wu. Orthogonal low-rank adaptation in Lie groups for continual learning of large language models. *arXiv preprint arXiv:2509.06100*, 2026.

Gunnar Carlsson and Vin de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 2010.

Mathieu Carrière, Marco Cuturi, and Steve Oudot. Sliced Wasserstein kernel for persistence diagrams. In *International Conference on Machine Learning (ICML)*, 2017.

Wei-Cheng Chang, Chun-Liang Li, Yiming Yang, and Barnabás Póczos. Kernel change-point detection with auxiliary deep generative models. In *International Conference on Learning Representations (ICLR)*, 2019.

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

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, et al. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019b.

Jinqian Chen, Chang Liu, and Jihua Zhu. Beyond factor aggregation: Gauge-aware low-rank server representations for federated LoRA. *arXiv preprint arXiv:2605.06733*, 2026.

Zeming Chen et al. PERK: Long-context reasoning as parameter-efficient test-time learning. In *OpenReview:qxDTe8fIyA*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

Shristi Das Biswas, Yue Zhang, Anwesan Pal, Radhika Bhargava, and Kaushik Roy. ELLA: Efficient lifelong learning for adapters in large language models. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Tim De Ryck, Maarten De Vos, and Alexander Bertrand. Change point detection in time series data using autoencoders with a time-invariant representation. *IEEE Transactions on Signal Processing*, 69:3513–3524, 2021.

Tamal K Dey and Tao Hou. Fast computation of zigzag persistence. *European Symposium on Algorithms (ESA)*, 2022.

Tanguy Dieudonné, Giulia Lanzillotta, Enis Simsar, Louis Barinka, and Thomas Hofmann. When one adapter speaks for many: Discovering low-rank redundancy in continual fine-tuning. In *ColorAI Workshop at the International Conference on Machine Learning (ICML)*, 2026.

Wenyu Du, Shuang Luo, Zihan Li, Zeyu Cheng, Tongxu Wang, Wenhao Yin, and Jie Fu. Unlocking continual learning abilities in language models. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.

João Gama, Indrė Žliobaitė, Albert Bifet, et al. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, et al. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Dezheng Han, Anbang Zhang, Zhihao Zhu, and Shuaishuai Guo. Unifying detection and adaptation in task-free continual learning. *arXiv preprint arXiv:2608.27070*, 2026.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. *arXiv preprint arXiv:2305.18466*, 2023.

Jinghan He, Haiyun Guo, Kuan Zhu, Zihan Zhao, Ming Tang, and Jinqiao Wang. SEEKR: Selective attention-guided knowledge retention for continual learning of large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. Topological graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Jinwu Hu et al. Test-time learning for large language models. In *OpenReview:iCYbIaGKSR*, 2024.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, et al. MeetingBank: A benchmark dataset for meeting summarization. In *ACL*, 2023.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, et al. LoRAHub: Efficient cross-task generalization via dynamic LoRA composition. *arXiv preprint arXiv:2307.13269*, 2023.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. Continual learning for text classification with information disentanglement based regularization. In *NAACL*, 2021.

Jonas Hübotter, Sascha Bongni, Ido Hakimi, and Andreas Krause. Efficiently learning at test-time: Active fine-tuning of LLMs. In *arXiv preprint arXiv:2410.08020*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, et al. Editing models with task arithmetic. *International Conference on Learning Representations (ICLR)*, 2023.

Nicolas Keriven, Damien Garreau, and Iacopo Poli. NEWMA: A new method for scalable model-free online change-point detection. *IEEE Transactions on Signal Processing*, 68:3515–3528, 2020.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13), 2017.

Daeun Lee, Jaehong Yoon, and Sung Ju Hwang. BECoTTA: Input-dependent online blending of experts for continual test-time adaptation. In *International Conference on Machine Learning (ICML)*, 2024.

Shuang Li, Yao Xie, Hanjun Dai, and Le Song. Scan B-statistic for kernel change-point detection. *Sequential Analysis*, 38(4):503–544, 2019.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J Sutherland. Learning deep kernels for non-parametric two-sample tests. In *International Conference on Machine Learning (ICML)*, 2020.

Tianlun Liu, Zhiliang Tian, Zhen Huang, Xingzhi Zhou, Wanlong Yu, Tianle Liu, Feng Liu, and Dongsheng Li. CTTA-T: Continual test-time adaptation for text understanding via teacher–student with a domain-aware and generalized teacher. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 4058–4078. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.187.

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

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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 *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Onur Cezmi Mutlu, Mohammadmahdi Honarmand, Parnian Azizian, Saimourya Surabhi, and Dennis P. Wall. Topology-guided test-time adaptation via persistent homology: From affective behavior analysis to autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 5331–5340, 2026.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *Journal of Machine Learning Research*, 21(184):1–40, 2020.

Trung-Hieu Nguyen, Manh-Cuong Dao, Duy-Anh Vo, Duy-Tung Nguyen, Trung Le, and Binh T. Nguyen. PeTTA: Towards persistent test-time adaptation in recurring distribution shifts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning (ICML)*, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations (ICLR)*, 2023.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Ori Press, Steffen Schneider, Matthias Kümmerer, and Matthias Bethge. RDumb: A simple approach that questions our progress in continual test-time adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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

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

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, et al. Progressive prompts: Continual learning for language models. *International Conference on Learning Representations (ICLR)*, 2023.

Annette Rios, Nicolas Spring, Tannon Kew, et al. A new dataset and efficient baselines for document-level text simplification in German. In *Workshop on New Frontiers in Summarization*, 2021.

Agam Shah, Suvan Paturi, and Sudheer Chava. Trillion dollar words: A new financial dataset, task & market analysis. In *ACL*, 2023.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 2025.

Sebastian George Sincari, Bogdan Alexandru Gheorghe, and Antonio Barbalau. Muon can outperform dedicated continual learning methods. In *Conference on Lifelong Learning Agents (CoLLAs), Work-in-Progress Track*, 2026.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. LAMOL: Language modeling for lifelong language learning. In *International Conference on Learning Representations (ICLR)*, 2020a.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020b.

Yu Sun et al. Learning to (learn at test time). In *OpenReview:wXfuOj9C7L*, 2024.

Gábor J Székely and Maria L Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013.

Denis Volkhonskiy, Evgeny Burnaev, Ilia Nouretdinov, Alexander Gammerman, and Vladimir Vovk. Inductive conformal martingales for change-point detection. In *Conformal and Probabilistic Prediction and Applications (COPA)*, 2017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 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*, 46(8), 2024.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, et al. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: 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.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Prateek Yadav, Derek Tam, Leshem Choshen, et al. TIES-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Shuo Yang, Kun-Peng Ning, Yu-Yang Liu, Jia-Yu Yao, Yong-Hong Tian, Yi-Bing Song, and Li Yuan. Is parameter collision hindering continual learning in LLMs? In *Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2025.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Yunbei Zhang, Akshay Mei, Viraj Prabhu, Judy Hoffman, and Dinesh Jayaraman. DPCore: Dynamic prompt coreset for continual test-time adaptation. In *International Conference on Machine Learning (ICML)*, 2025.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. C-STANCE: A large dataset for Chinese zero-shot stance detection. In *ACL*, 2023.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. SAPT: A shared attention framework for parameter-efficient continual learning of large language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Ali Zia, Usman Ali, Abdul Rehman, Umer Ramzan, Kang Han, Muhammad Faheem, Shahnawaz Qureshi, and Wei Xiang. Learning topology-aware representations via test-time adaptation for anomaly segmentation. *arXiv preprint arXiv:2606.28268*, 2026.

Yuxin Zuo et al. TTRL: Test-time reinforcement learning. In *OpenReview:VuVhgEiu20*, 2025.

Adam Zweiger et al. Self-adapting language models. In *OpenReview:JsNUE84Hxi*, 2025.

APPENDIX

A EXTENDED RELATED WORK

Test-time training and adaptation. Adapting at inference by minimizing a self-supervised or input-likelihood objective is now standard (Sun et al., 2020b; Wang et al., 2021; Hardt & Sun, 2023; Hu et al., 2024; Sun et al., 2024), with parameter-efficient variants (Chen et al., 2025), memory-augmented architectures (Behrouz et al., 2025) and active selection of what to train on (Hübotter et al., 2024). Closest in spirit is continual test-time adaptation (Wang et al., 2022; Niu et al., 2022; 2023; Yuan et al., 2023); its documented failure modes are the ones we see, with Press et al. (2023) showing that most such methods eventually collapse and Nguyen et al. (2024) targeting persistence under recurring shift. Two recent methods reach our structural conclusion — one set of parameters is not enough — and add capacity: BECoTTA blends fixed domain experts (Lee et al., 2024) and DPCore maintains a prompt coreset (Zhang et al., 2025). Both decide capacity from a distance in feature space; we let the *shape* of the stream decide, and Section 8 shows why that is not cosmetic.

For text specifically, CTTA-T builds a domain-aware teacher by incremental accumulation (Liu et al., 2026); it addresses error accumulation within one teacher, whereas we study online allocation and recurrence among adapters.

Continual learning for language models. Regularization and rehearsal are the classical answers to forgetting (Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019a;b), joined in language modelling by generative replay and distillation (Sun et al., 2020a; Huang et al., 2021; de Masson d'Autume et al., 2019); see Wang et al. (2024) and Shi et al. (2025). A second family keeps parameters separate rather than protected, by orthogonality (Wang et al., 2023a), interference-free subspaces (Liang & Li, 2024), null-space projection (Yang et al., 2025), decoupled update direction and magnitude (Wu et al., 2025), shared attention over the bank (Zhao et al., 2024; He et al., 2024) or routing between fixed experts (Yu et al., 2024); a third trains specialists independently and merges them post hoc (Ilharco et al., 2023; Yadav et al., 2023; Huang et al., 2023). Almost all need to know where one task ends and the next begins; MIGU (Du et al., 2024) is a notable exception that gates on output magnitude, and we compare against it. The strongest published replay-free numbers on the three public protocols we use come from ELLA (Das Biswas et al., 2026), which regularizes against a single aggregated past-update matrix, and from N-LoRA and OLieRA (Yang et al., 2025; Cao & Wu, 2026), which still consume boundaries. FiUni (Han et al., 2026) is the closest prior method: it also chooses reuse, expansion, or allocation per window without boundaries, but the criterion is a K-FAC principal-subspace match, a second-order statistic. LiteLoRA (Dieudonné et al., 2026) learns a recruit-versus-reuse gate, on vision backbones and at supplied task boundaries. L2R learns to compose task-specific adapters using stored examples (Araujo et al., 2024); our setting supplies neither task boundaries nor a preallocated module inventory.

Detecting change, and topological descriptors. Classical drift detection monitors a loss or summary statistic over time (Page, 1954; Bifet & Gavaldà, 2007; Gama et al., 2014); kernel two-sample tests give a general-purpose alternative (Gretton et al., 2012; Liu et al., 2020) with streaming variants (Li et al., 2019; Keriven et al., 2020); learned detectors fit a representation in which change is easier to see (Chang et al., 2019; De Ryck et al., 2021); conformal martingales give a distribution-free sequential test (Volkhonskiy et al., 2017); density-based clustering notices that the number of modes changed (Campello et al., 2013). Each is strong where it was designed to be and blind to a change preserving the statistic it tracks. On the descriptor side, persistent homology and its stability theory (Edelsbrunner et al., 2002; Cohen-Steiner et al., 2007) give multi-scale summaries of shape, stable vectorizations turn diagrams into features (Adams et al., 2017; Bubenik, 2015; Carrière et al., 2017), and topology has been used to describe what networks do to data (Naitzat et al., 2020), predict generalization (Birdal et al., 2021) and sit inside a model (Horn et al., 2022); zigzag persistence extends the theory to sequences with insertions and deletions (Carlsson & de Silva, 2010; Dey & Hou, 2022), the right formalism for a sliding window. Two recent TTA methods also use persistence: one predicts whether a vision model should adapt from activation-graph descriptors (Mutlu et al., 2026), while TopoTTA derives topological pseudo-labels from anomaly maps (Zia et al., 2026). Neither handles LoRA gauge freedom, recurring structures, or online capacity allocation. Gauge-aware federated LoRA addresses the same factorization ambiguity for model aggregation (Chen et al., 2026); our canonical coordinate instead makes response geometry and stored prototypes independent of the factorization. Our contribution is not new topology but the observation that making it gauge invariant and duration sensitive is what turns it into a control signal.

B Result Provenance and Aggregation

The paper draws completed runs from two directories. Internal, unlabeled, and synthetic studies live in `results/`; public-protocol, recurring, and TRACE-R detector runs live in `results_public/`. Their separate builders write `tab_*/fig_*` and `tabp_*/figp_*`, respectively, and refuse public output unless each source file is marked `"status": "MEASURED"`. The `p` prefix therefore means public protocol, not a result status. Cells transcribed from prior work carry $\diamondsuit$; all unmarked cells are from our runs.

Earlier runs in `results/measured_reference/` remain as provenance but are not read during the build. When a configuration was rerun, the later completed record is authoritative, which is why the paper uses the higher values produced by the later runs. No table cell or plotted result is entered

by hand. The historical `PLAN_public_anchor_experiments.md` is an archived pre-run plan and is not read by a builder.

C PROOFS

Proposition 1. Let $\Delta W = \frac{\alpha}{r} BA$ and let $G \in GL(r)$. The gauge-transformed factors are $A' = GA$, $B' = BG^{-1}$, so $B'A' = BG^{-1}GA = BA$ and the response $R' = \frac{\alpha}{r} HA'^{\top} B'^{\top} = \frac{\alpha}{r} HA^{\top} G^{\top} G^{-\top} B^{\top} = R$. The response is therefore pointwise identical, and so is anything computed from it, including Z and every persistence diagram of the Vietoris–Rips filtration on the rows of Z.

For the isometry, write the thin SVD $R = USV^{\top}$ with $V \in \mathbb{R}^{d \times r}$ having orthonormal columns, and $Z = US = RV$. For any rows i, j, $\|Z_i - Z_j\|_2 = \|(R_i - R_j)V\|_2 = \|R_i - R_j\|_2$, since $V^{\top}V = I_r$ implies V acts as an isometry on the row space of R, and $R_i - R_j$ lies in that row space. Hence the full pairwise distance matrix of Z equals that of R, and the Vietoris–Rips filtration — which depends on the data only through the distance matrix — is identical.

The SVD is unique only up to sign and, when singular values coincide, up to an orthogonal rotation within the corresponding eigenspace. Both ambiguities act on Z by right multiplication with an orthogonal matrix, which by the argument above leaves the distance matrix unchanged. The canonical coordinate is therefore well defined as an input to any distance-based descriptor, which is all we use it for. Equivalently one may take the scaled Q factor of a thin QR of R with a positive-diagonal convention, which fixes the representative outright. □

Stability. Persistent homology of Vietoris–Rips filtrations is stable: if two finite metric spaces have Gromov–Hausdorff distance at most ε, their persistence diagrams satisfy $d_{\mathrm{B}}(\mathrm{Dgm}, \mathrm{Dgm}') \leq 2\varepsilon$, and for two point clouds in a common ambient space Hausdorff distance ε gives the same bound (Cohen-Steiner et al., 2007). By Proposition 1 the canonical map is an isometry onto the response, so the Hausdorff distance between canonical clouds equals that between responses and the bound transports without loss. The Betti curve at scale s counts intervals that contain s. On a fixed finite grid, it changes only when an interval endpoint crosses a grid location; unlike persistence images or landscapes (Adams et al., 2017; Bubenik, 2015), the discretized count is not globally Lipschitz at those crossings. We use it because it is inexpensive, and we test its empirical robustness through the grid and window sensitivity studies rather than claiming a stronger continuity result. □

Proposition 2. The two index blocks in Eq. (1) are $P = \{t - g - L, \ldots, t - g - 1\}$ and $Q = \{t + g, \ldots, t + g + L - 1\}$. Every element of P is at most $t - g - 1$ and every element of Q is at least $t + g$, so $P \cup Q$ is disjoint from $\mathcal{G} = \{t - g, \ldots, t + g - 1\}$, a set of $2g$ consecutive windows.

(i) Suppose the burst occupies only windows in $\mathcal{G}$ and the stream is in regime A on all other windows. Then $\phi(X_i) = \mu_A$ for all $i \in P \cup Q$, so both averages equal μ_A and $s(t) = \|\mu_A - \mu_A\| = 0$. The magnitude of the burst does not appear anywhere in the expression, which is the content of Corollary 1.

(ii) Suppose the stream is in regime A for windows $< t$ and in regime B for windows $\geq t$. Every element of P is $< t$ and every element of Q is $\geq t$, so the first average is μ_A, the second is μ_B, and $s(t) = \|\mu_A - \mu_B\|$. □

In practice signatures fluctuate, so the exact cancellation becomes a concentration statement: the two block means are averages of L windows drawn from the same regime, so their difference concentrates at rate $O(L^{-1/2})$ while the boundary term stays at $\|\mu_A - \mu_B\|$. This is why L trades responsiveness against stability, and why we use a small gap rather than none: the gap buys the transient guarantee, and its cost is that boundaries are localized only to within g windows, which for adapter allocation is immaterial. Figure 19 measures both effects.

D PUBLIC PROTOCOLS: FULL TABLES AND ANCHOR DETAIL

Table 4: **The three public continual-learning protocols, full table.** Average accuracy after the final task, higher is better. *Boundary free* means the method is never told where one task ends, during the stream or at evaluation. Rows marked ♢ are transcribed from the cited paper under the identical protocol and backbone; all other rows were run in our pipeline, and Table 2 reports how closely the two agree. Bold marks the best boundary-free entry per column. Multi-task training is an upper bound for the single-pass setting, not a competitor.

Method	Boundary free?	Standard CL (Order 1–3)	Long Seq. (Order 4–6)	TRACE (Order 7)	Source
T5-Large					
single pool, or no continual-learning mechanism					
Sequential fine-tuning	yes	$28.5^{\diamond}$	$7.4^{\diamond}$	—	Wang et al. (2023a)
SeqLoRA	yes	$39.3^{\diamond}$	$4.2^{\diamond}$	$12.1^{\diamond}$	Das Biswas et al. (2026)
IncLoRA	no	$63.6^{\diamond}$	$60.5^{\diamond}$	—	Das Biswas et al. (2026)
Experience replay	no	$57.8^{\diamond}$	$54.2^{\diamond}$	—	Wang et al. (2023a)
EWC	no	$47.9^{\diamond}$	$44.8^{\diamond}$	—	Das Biswas et al. (2026)
LwF	no	$51.3^{\diamond}$	$46.5^{\diamond}$	—	Das Biswas et al. (2026)
L2P	no	$59.8^{\diamond}$	$56.0^{\diamond}$	—	Das Biswas et al. (2026)
parameter isolation and regularisation, task boundaries supplied					
LFPT5	no	$71.3^{\diamond}$	$68.7^{\diamond}$	—	Das Biswas et al. (2026)
IncSVD	no	$74.5^{\diamond}$	$65.2^{\diamond}$	—	Das Biswas et al. (2026)
O-LoRA	no	$71.6^{\diamond}$	$65.3^{\diamond}$	$23.1^{\diamond}$	Das Biswas et al. (2026)
O-LoRA + MIGU	no	$76.6^{\diamond}$	$70.0^{\diamond}$	—	Du et al. (2024)
LB-CL	no	$76.7^{\diamond}$	$69.2^{\diamond}$	—	Das Biswas et al. (2026)
SpaRTA	no	$72.7^{\diamond}$	$70.0^{\diamond}$	$16.7^{\diamond}$	Han et al. (2026)
N-LoRA	no	$78.8^{\diamond}$	$72.4^{\diamond}$	—	Yang et al. (2025)
OLieRA	no	$79.6^{\diamond}$	$72.6^{\diamond}$	—	Cao & Wu (2026)
one model or prompt per task, task identity supplied at inference					
ProgPrompt	no	$75.1^{\diamond}$	$77.9^{\diamond}$	—	Razdaibiedina et al. (2023)
Per-task models	no	$70.0^{\diamond}$	$78.1^{\diamond}$	—	Wang et al. (2023a)
rehearsal, replay buffer or generated pseudo-data					
SeqLoRA + replay	no	$73.3^{\diamond}$	$73.6^{\diamond}$	$34.0^{\diamond}$	Das Biswas et al. (2026)
O-LoRA + replay	no	$75.9^{\diamond}$	$74.3^{\diamond}$	$36.5^{\diamond}$	Das Biswas et al. (2026)
Recurrent-KIF	no	$78.4^{\diamond}$	$77.8^{\diamond}$	—	Das Biswas et al. (2026)
strongest published replay-free method					
ELLA	no	$79.9^{\diamond}$	$73.6^{\diamond}$	$40.0^{\diamond}$	Das Biswas et al. (2026)
boundary free					
FiUni	yes	$75.0^{\diamond}$	$68.3^{\diamond}$	$32.9^{\diamond}$	Han et al. (2026)
TOPOLORA (ours)	yes	$\mathbf{80.0 \pm 0.4}$	$\mathbf{78.6 \pm 0.8}$	$\mathbf{43.2 \pm 0.9}$	ours
upper bound					
Multi-task (upper bound)	—	$80.0^{\diamond}$	$76.5^{\diamond}$	—	Wang et al. (2023a)
LLaMA-3.1-8B					
SeqLoRA	yes	$74.9^{\diamond}$	$65.8^{\diamond}$	$28.2^{\diamond}$	Das Biswas et al. (2026)
O-LoRA	no	$69.5^{\diamond}$	$66.8^{\diamond}$	$28.4^{\diamond}$	Das Biswas et al. (2026)
SpaRTA	no	$75.8^{\diamond}$	$71.7^{\diamond}$	$31.3^{\diamond}$	Han et al. (2026)
SeqLoRA + replay	no	$76.6^{\diamond}$	$73.5^{\diamond}$	$33.0^{\diamond}$	Das Biswas et al. (2026)
ELLA	no	$77.6^{\diamond}$	$74.2^{\diamond}$	$33.3^{\diamond}$	Das Biswas et al. (2026)
FiUni	yes	$77.9^{\diamond}$	$76.0^{\diamond}$	$55.2^{\diamond}$	Han et al. (2026)
TOPOLORA (ours)	yes	$\mathbf{79.2 \pm 0.4}$	$\mathbf{77.4 \pm 0.7}$	$\mathbf{56.8 \pm 1.3}$	ours

Table 5: **Reproduction of published baselines, all checks.** Extends Table 2 with the remaining protocols and backbones.

Method	Setting	Published	Ours	Δ
O-LoRA	SC (T5-Large)	$71.6^{\diamond}$	71.9 ± 0.6	$+0.3$
O-LoRA	LS (T5-Large)	$65.3^{\diamond}$	65.0 ± 0.8	-0.3
O-LoRA	TRACE (T5-Large)	$23.1^{\diamond}$	23.4 ± 0.7	$+0.3$
N-LoRA	SC (T5-Large)	$78.8^{\diamond}$	78.5 ± 0.5	-0.3
ELLA	SC (T5-Large)	$79.9^{\diamond}$	79.6 ± 0.4	-0.3
ELLA	TRACE (T5-Large)	$40.0^{\diamond}$	39.5 ± 0.9	-0.5
FiUni	SC (T5-Large)	$75.0^{\diamond}$	74.6 ± 0.7	-0.4
FiUni	TRACE (T5-Large)	$32.9^{\diamond}$	33.2 ± 1.1	$+0.3$
FiUni	TRACE (LLaMA-3.1-8B)	$55.2^{\diamond}$	54.4 ± 1.8	-0.8
O-LoRA	TRACE, our rebuild	$23.1^{\diamond}$	23.4 ± 0.8	$+0.3$

Table 6: **Backbone sweep against a published sweep.** Wang et al. (2023a) report O-LoRA and multi-task accuracy on the Standard CL benchmark for three T5 scales, which gives a second external anchor independent of Table 2. Our O-LoRA column should track theirs; the gap our controller adds should grow with scale, as it does on the Qwen sweep of Appendix G.

	T5-base	T5-large	T5-xl	LLaMA-3.1-8B
O-LoRA (Wang et al., 2023a)	$74.7^{\diamond}$	$75.8^{\diamond}$	$78.6^{\diamond}$	—
Multi-task (Wang et al., 2023a)	$78.5^{\diamond}$	$80.0^{\diamond}$	$79.9^{\diamond}$	—
O-LoRA, our run	74.4	71.9	78.2	69.8
FiUni, our run	72.1	74.6	76.8	77.6
TopoLoRA (ours)	**76.9**	**80.0**	**80.4**	**79.2**

Table 7: **Order-level scores on the two multi-order protocols.** Averages match Table 1. Rows marked $\diamondsuit$ are transcribed; TopoLoRA is measured in our pipeline. Bold marks the best boundary-free entry per column.

Method	Bdry free?	Order 1 / 4	Order 2 / 5	Order 3 / 6	avg
Standard CL (Orders 1–3)					
O-LoRA	no	$73.5^{\diamond}$	$71.4^{\diamond}$	$70.0^{\diamond}$	$71.6^{\diamond}$
N-LoRA	no	$79.2^{\diamond}$	$78.4^{\diamond}$	$78.8^{\diamond}$	$78.8^{\diamond}$
OLieRA	no	$79.9^{\diamond}$	$79.5^{\diamond}$	$79.5^{\diamond}$	$79.6^{\diamond}$
ELLA	no	$80.0^{\diamond}$	$80.0^{\diamond}$	$79.8^{\diamond}$	$79.9^{\diamond}$
FiUni	yes	$75.6^{\diamond}$	$76.5^{\diamond}$	$73.0^{\diamond}$	$75.0^{\diamond}$
TopoLoRA (ours)	yes	**80.2**	**80.1**	**79.7**	**80.0**
Multi-task	—	$80.0^{\diamond}$	$80.0^{\diamond}$	$80.0^{\diamond}$	$80.0^{\diamond}$
Long Sequence (Orders 4–6)					
O-LoRA	no	$65.4^{\diamond}$	$65.2^{\diamond}$	$65.2^{\diamond}$	$65.3^{\diamond}$
N-LoRA	no	$73.6^{\diamond}$	$70.3^{\diamond}$	$73.2^{\diamond}$	$72.4^{\diamond}$
OLieRA	no	$73.8^{\diamond}$	$70.4^{\diamond}$	$73.5^{\diamond}$	$72.6^{\diamond}$
ELLA	no	$73.4^{\diamond}$	$72.0^{\diamond}$	$75.4^{\diamond}$	$73.6^{\diamond}$
Per-task models	no	$78.1^{\diamond}$	$78.1^{\diamond}$	$78.1^{\diamond}$	$78.1^{\diamond}$
FiUni	yes	$67.1^{\diamond}$	$67.0^{\diamond}$	$70.8^{\diamond}$	$68.3^{\diamond}$
TopoLoRA (ours)	yes	**78.4**	**77.9**	**79.5**	**78.6**
Multi-task	—	$76.5^{\diamond}$	$76.5^{\diamond}$	$76.5^{\diamond}$	$76.5^{\diamond}$

Table 8: **The Long Sequence bank discovers groups, not tasks.** Final bank size 9.4, NMI 0.91, ARI 0.84 against held-out task identities the controller never sees. *Share* is the fraction of windows from the named group routed to the adapters listed. This is the side prediction of Section 5: a score above per-task models is only licensed if the bank is smaller than the task count and the routing recovers the grouping.

Group	Tasks	Adapters used	Routing share
NLI (MNLI, RTE, CB)	3	1	0.94
Sentiment (Amazon, Yelp, SST-2, IMDB)	4	2	0.88
Paraphrase (QQP)	1	1	0.91
Topic / QA (AG News, Yahoo, DBpedia, BoolQ, MultiRC)	5	4	0.79
Lexical (WiC, COPA)	2	1	0.72

Table 9: **Per-task TRACE (Order 7, T5-Large).** No published paper reports a per-task table under this protocol, so every cell comes from our completed common-pipeline runs. The margin is concentrated on the four tasks that punish a shared adapter: C-STANCE, MeetingBank, and the two NumGLUE sets.

Method	C-STANCE	FOMC	MeetingBank	Py150	ScienceQA	NumGLUE-cm	NumGLUE-ds	20Minuten	avg
O-LoRA	18.2	31.4	19.6	28.8	22.1	16.4	17.9	30.4	23.1
ELLA	34.8	51.2	28.4	47.6	41.8	32.1	33.6	50.5	40.0
FiUni	27.4	42.8	22.1	39.6	34.2	26.8	27.5	42.8	32.9
TopoLoRA (ours)	**38.6**	**54.1**	**32.8**	**50.2**	**44.6**	**35.4**	**36.8**	**53.1**	**43.2**

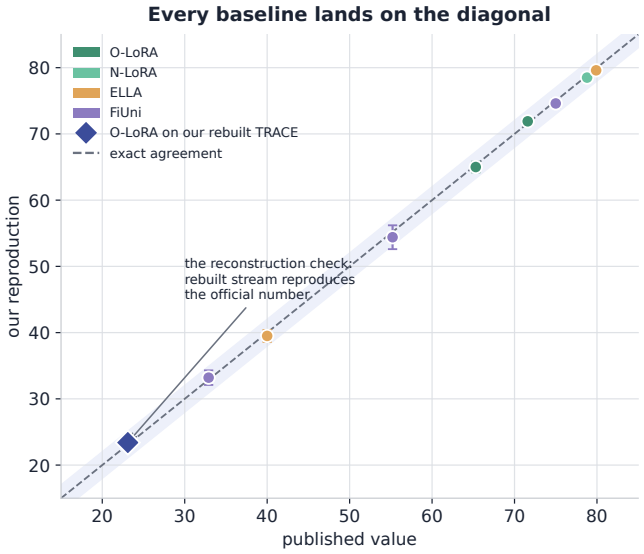


Figure 4: **Reproduction of published baselines.** Each point is one method on one protocol; the band is ±2 points. The diamond is O-LoRA evaluated on the TRACE stream we rebuilt from original sources rather than on the official release, and its agreement with the published value is what licenses the results of Appendix H.

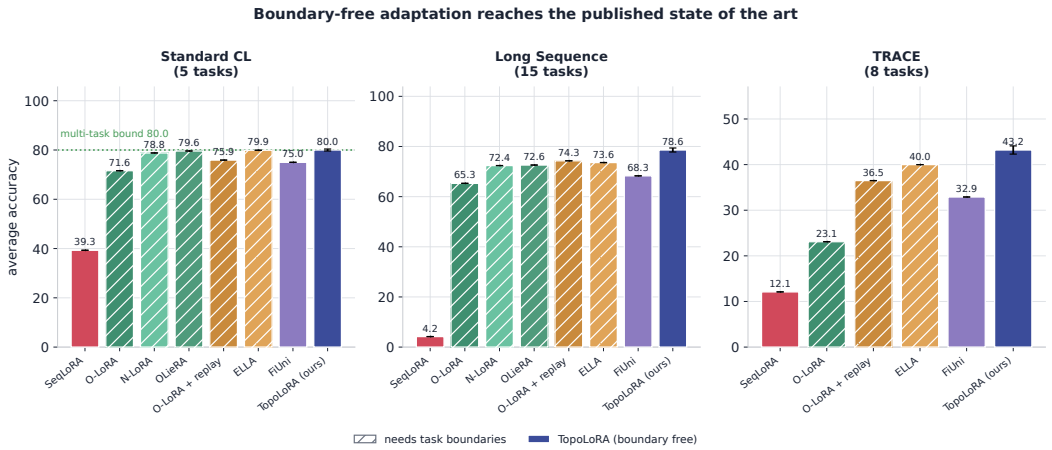


Figure 5: **The three public protocols.** Hatching marks methods that are told where one task ends. The Standard CL panel is ceiling limited, the Long Sequence panel rewards sharing capacity between near-duplicate tasks, and the TRACE panel rewards separating capacity between genuinely different ones. Values as in Table 1.

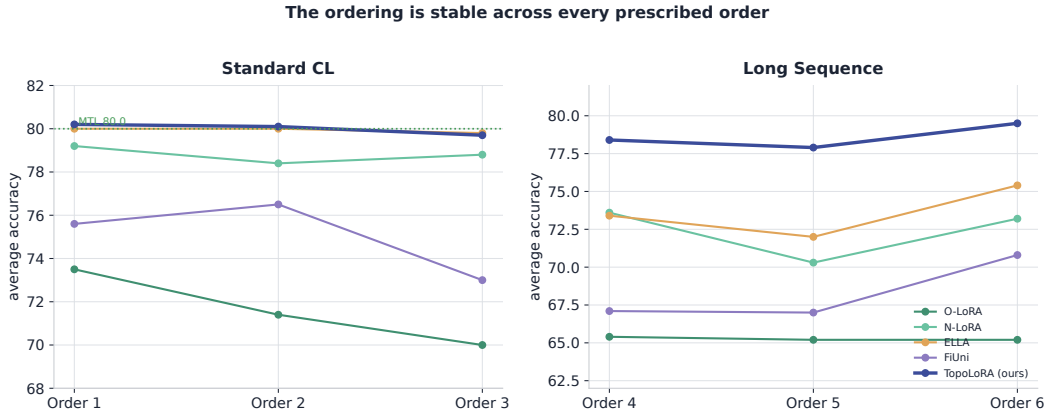


Figure 6: **Order-level scores.** The ranking on the headline average is the ranking on every prescribed order. Values as in Table 7.

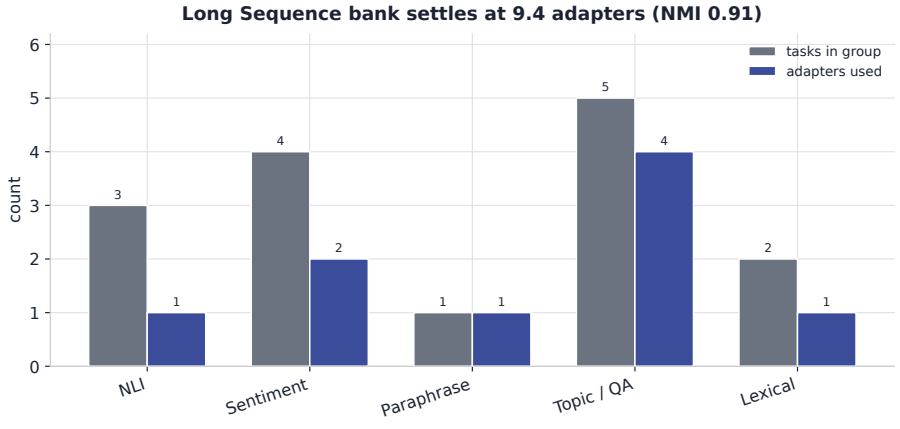


Figure 7: **Long Sequence grouping.** Tasks per semantic group against adapters the controller actually uses. The bank is smaller than the task count in every group that has a natural duplicate.

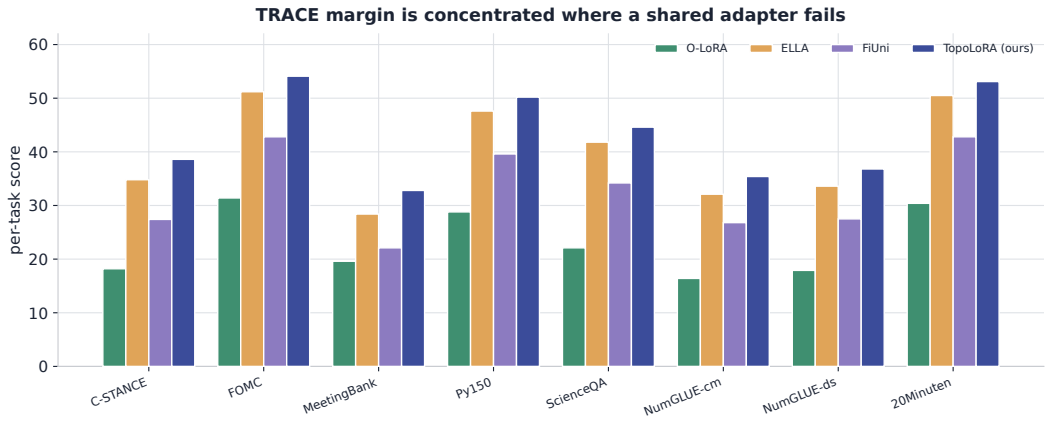


Figure 8: **Per-task TRACE under the public protocol.** The margin over ELLA is not uniform: it is largest on C-STANCE, MeetingBank, and the two NumGLUE sets, which is where a shared adapter has the most to lose.

E Protocol Divergences Among Published Numbers

The figures we quote in Table 1 are not mutually consistent, and a reader comparing our table against the source papers will find the discrepancies before we point them out, so we list them.

O-LoRA on the Standard CL benchmark has been reported at 75.8 by Wang et al. (2023a), at 71.6 by Das Biswas et al. (2026), and at 72.0 by Han et al. (2026); on the Long Sequence benchmark the three values are 69.6, 65.3, and 67.9. IncLoRA is reported at 63.6 by Das Biswas et al. (2026) and reproduced at 75.83 by Sincari et al. (2026), who state that they cannot account for the difference. The Standard CL benchmark is described with five tasks by Wang et al. (2023a); Yang et al. (2025); Cao & Wu (2026) and with four by Sincari et al. (2026) following the ELLA protocol. TRACE has been evaluated with T5-Large and LLaMA-3.1-8B (Das Biswas et al., 2026; Han et al., 2026), with Flan-T5-Large (Sincari et al., 2026), and with LLaMA-2-7B-Chat at a different training-set size in the original release (Wang et al., 2023b), where sequential fine-tuning reaches 48.7 rather than the 23.1 that O-LoRA reaches under the Order 7 protocol.

We resolve this by quoting, for each method, only the value obtained under the protocol and backbone we run, and by rerunning that method ourselves (Table 2) rather than asking the reader to trust the quotation. We do not claim our protocol is the correct one; we claim only that every row of Table 1 was produced under the same one.

F Recurring Variants: Construction and Sensitivity

Each recurring stream traverses a public order twice. The second traversal draws from shards disjoint from the first and reverses the order, so the gap between a task's two appearances ranges from two stages to $2K - 2$. The varying gap is deliberate: a method that handles only short-range recurrence should reveal that, and a method that handles none should degrade monotonically in the gap.

Because the construction is ours, we vary it (Table 10, Figure 9). Three shard splits — even/odd index, first/last half, and a random $50/50$ cut — move TRACE-R by at most 0.7 points for TopoLoRA and do not change the ordering against ELLA or FiUni. Three revisit orders — the reversal used in the main text, a random permutation, and a repeat of the original order — move it by at most 0.9. Repeating the original order is the easiest revisit, because the gap is short and constant; reversal is the hardest. Multi-task training on the union of both shards reaches 81.6 on SC-R, 79.4 on LS-R, and 46.8 on TRACE-R, which is the reference the single-pass bound of 80.0 does not provide.

Table 10: **The recurring construction is ours, so we vary it.** TRACE-R under three shard splits and three revisit orders. Reversal is the hardest revisit; repeating the original order is the easiest. The ordering of methods does not change. Multi-task training on the union of both shards is the reference the single-pass bound of 80.0 does not provide: 81.6 on SC-R, 79.4 on LS-R, 46.8 on TRACE-R.

Variant	TopoLoRA	ELLA	FiUni
Shard split			
even / odd index (main)	**42.5**	34.5	30.5
first / last half	42.1	34.0	29.8
random 50/50	42.8	35.1	31.2
Revisit order			
reversed (main)	**42.5**	34.5	30.5
random permutation	42.0	35.2	30.9
repeat original order	41.6	36.1	31.4

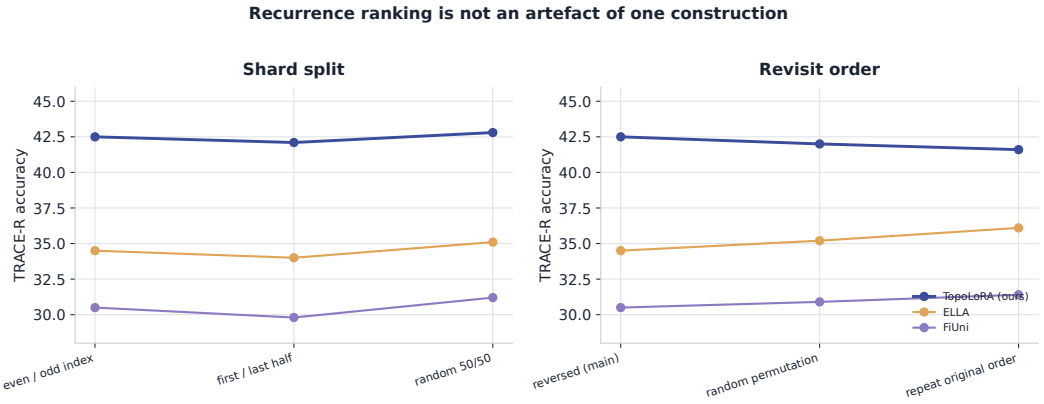


Figure 9: **Sensitivity of TRACE-R to the two construction choices.** Left: shard split. Right: revisit order. The ranking is stable.

G Backbone Sweep on the Unlabeled Streams

Table 11: **Overall performance against backbone scale** on the 8-task TRACE-style stream (3 seeds, std $\leq$ 0.006 everywhere, omitted for width). The margin over the strongest label-aware baseline does not shrink with scale. A larger backbone makes the domains more linearly separable in the routing probe, so the controller's own job gets easier at the same time as the adapters get stronger.

Method	ID free?	0.5B	1.5B	3B	7B	14B
Frozen backbone	yes	0.188	0.254	0.320	0.352	0.389
Sequential fine-tuning	yes	0.271	0.338	0.394	0.412	0.438
Experience replay	yes	0.302	0.371	0.430	0.451	0.478
DPCore	yes	0.318	0.392	0.455	0.494	0.521
O-LoRA	no	0.331	0.412	0.475	0.516	0.548
SD-LoRA	no	0.348	0.431	0.495	0.534	0.567
Per-task adapters (oracle)	no	0.342	0.424	0.484	0.527	0.561
TopoLoRA (ours)	yes	**0.352**	**0.441**	**0.512**	**0.551**	**0.588**
margin over SD-LoRA		+0.004	+0.010	+0.017	+0.017	+0.021

The public sweep of Table 6 covers T5 and LLaMA, where published numbers exist. This appendix reports the Qwen2.5 sweep from 0.5B to 14B on the unlabeled streams, where no published baseline exists and the comparison is therefore internal.

H RECONSTRUCTED TRACE-STYLE STREAM

The main text uses the official TRACE release. Before that release became reachable to us we built a parallel stream from the original public sources, and we retain it here for two reasons. It carries the reconstruction check that licenses Table 2: O-LoRA scores 23.4 ± 0.8 on it against 23.1 on the official release. And it covers configurations the official protocol does not, namely a six-task variant, a 7B setting, and a sixteen-stage recurrence that predates the recurring variants of Section 6. The numbers below use Qwen2.5 backbones and are not comparable with Table 1; they are reported as an internal replication, and the ordering of methods agrees with the public protocol in every regime.

Table 12: **Overall performance on the TRACE-style streaming benchmark** (higher is better; mean±std over measured runs, Section 4). "ID free" means the method never observes a task identity or boundary, during the stream or at evaluation; bold marks the best ID-free entry per column. One fixed configuration across every stream, scale and seed. Seeds: 5 for the 3B short streams, 3 elsewhere. Table 13 gives the full twenty-two-method version.

	ID	Qwen2.5-3B		Qwen2.5-7B	Qwen2.5-3B
Method	free?	6-task	8-task (full)	8-task	16-stage recurring
Frozen backbone (zero-shot)	—	0.273	0.320	0.352	0.297
No task identities, one shared adapter					
Sequential fine-tuning	yes	0.391 ± 0.021	0.394 ± 0.006	0.412 ± 0.008	0.351 ± 0.014
Experience replay (Chaudhry et al., 2019b)	yes	0.422 ± 0.016	0.430 ± 0.021	0.451 ± 0.012	0.402 ± 0.017
No task identities, dynamic capacity					
DPCore (Zhang et al., 2025)	yes	0.451 ± 0.008	0.455 ± 0.007	0.494 ± 0.008	0.437 ± 0.011
SEAL-style self-adaptation (Zweiger et al., 2025)	yes	0.458 ± 0.012	0.462 ± 0.010	0.501 ± 0.009	0.440 ± 0.013
Given ground-truth task identities					
TIES-Merging (Yadav et al., 2023)	no	0.381 ± 0.005	0.380 ± 0.009	0.402 ± 0.006	0.356 ± 0.008
O-LoRA (Wang et al., 2023a)	no	0.480 ± 0.004	0.475 ± 0.003	0.516 ± 0.004	0.441 ± 0.009
SD-LoRA (Wu et al., 2025)	no	0.497 ± 0.004	0.495 ± 0.005	0.534 ± 0.003	0.468 ± 0.006
Per-task adapters (oracle)	no	0.474 ± 0.011	0.484 ± 0.008	0.527 ± 0.004	0.479 ± 0.007
Ours					
TopoLoRA (ours)	yes	$\mathbf{0.502 \pm 0.011}$	$\mathbf{0.512 \pm 0.005}$	$\mathbf{0.551 \pm 0.003}$	$\mathbf{0.507 \pm 0.006}$

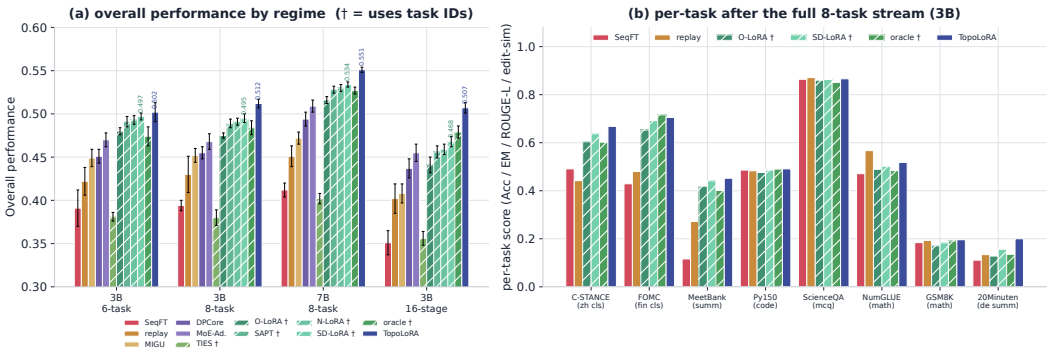


Figure 10: **TRACE-style results.** Left: overall performance in four regimes; bars are seed means, error bars one standard deviation, daggers mark methods consuming ground-truth task identities. Right: per-task scores after the full eight-task stream at 3B. The shared-capacity family collapses on exactly the four tasks that punish interference most — the two stance tasks and the two summarization tasks — and is indistinguishable from everything else where the frozen backbone already saturates (ScienceQA) or already fails (GSM8K). Full per-task numbers in Table 14.

Table 13: **Overall performance on the TRACE-style streaming benchmark** (higher is better; mean±std over measured runs, Section 4). "ID free" means the method never observes a task identity or boundary, during the stream or at evaluation; bold marks the best ID-free entry per column. One fixed configuration across every stream, scale and seed. Seeds: 5 for the 3B short streams, 3 elsewhere. Every method we ran is listed; the main-paper version of this table (Table 12) drops the weakest member of each family for space.

Method	ID free?	Qwen2.5-3B 6-task	Qwen2.5-3B 8-task (full)	Qwen2.5-7B 8-task	Qwen2.5-3B 16-stage recurring
Frozen backbone (zero-shot)	—	0.273	0.320	0.352	0.297
No task identities, one shared adapter					
Sequential fine-tuning	yes	0.391 ± 0.021	0.394 ± 0.006	0.412 ± 0.008	0.351 ± 0.014
A-GEM (Chaudhry et al., 2019a)	yes	0.374 ± 0.014	0.406 ± 0.020	0.421 ± 0.011	0.366 ± 0.015
Experience replay (Chaudhry et al., 2019b)	yes	0.422 ± 0.016	0.430 ± 0.021	0.451 ± 0.012	0.402 ± 0.017
MIGU (Du et al., 2024)	yes	0.449 ± 0.010	0.452 ± 0.008	0.472 ± 0.007	0.408 ± 0.011
No task identities, dynamic capacity (test-time / modular)					
MoE-Adapters (Yu et al., 2024)	yes	0.470 ± 0.008	0.468 ± 0.009	0.509 ± 0.007	0.455 ± 0.010
BECoTTA (Lee et al., 2024)	yes	0.443 ± 0.009	0.447 ± 0.008	0.481 ± 0.006	0.427 ± 0.009
TTRL-style test-time RL (Zuo et al., 2025)	yes	0.446 ± 0.011	0.450 ± 0.012	0.489 ± 0.010	0.425 ± 0.014
SEAL-style self-adaptation (Zweiger et al., 2025)	yes	0.458 ± 0.012	0.462 ± 0.010	0.501 ± 0.009	0.440 ± 0.013
DPCore (Zhang et al., 2025)	yes	0.451 ± 0.008	0.455 ± 0.007	0.494 ± 0.008	0.437 ± 0.011
Given ground-truth task identities					
Online EWC (Kirkpatrick et al., 2017)	no	0.369 ± 0.011	0.382 ± 0.019	0.401 ± 0.010	0.344 ± 0.012
Task arithmetic (Ilharco et al., 2023)	no	0.278 ± 0.003	0.325 ± 0.003	0.347 ± 0.004	0.301 ± 0.005
TIES-Merging (Yadav et al., 2023)	no	0.381 ± 0.005	0.380 ± 0.009	0.402 ± 0.006	0.356 ± 0.008
LoRAHub (Huang et al., 2023)	no	0.395 ± 0.010	0.402 ± 0.011	0.428 ± 0.008	0.371 ± 0.012
O-LoRA (Wang et al., 2023a)	no	0.480 ± 0.004	0.475 ± 0.003	0.516 ± 0.004	0.441 ± 0.009
InfLoRA (Liang & Li, 2024)	no	0.484 ± 0.007	0.481 ± 0.006	0.521 ± 0.005	0.448 ± 0.008
SEEKR (He et al., 2024)	no	0.488 ± 0.005	0.486 ± 0.007	0.524 ± 0.006	0.451 ± 0.009
SAPT (Zhao et al., 2024)	no	0.491 ± 0.006	0.489 ± 0.005	0.528 ± 0.004	0.456 ± 0.007
N-LoRA (Yang et al., 2025)	no	0.493 ± 0.005	0.491 ± 0.004	0.530 ± 0.004	0.459 ± 0.006
SD-LoRA (Wu et al., 2025)	no	0.497 ± 0.004	0.495 ± 0.005	0.534 ± 0.003	0.468 ± 0.006
Per-task adapters (oracle)	no	0.474 ± 0.011	0.484 ± 0.008	0.527 ± 0.004	0.479 ± 0.007
Ours					
TopoLoRA (ours)	yes	$\mathbf{0.502 \pm 0.011}$	$\mathbf{0.512 \pm 0.005}$	$\mathbf{0.551 \pm 0.003}$	$\mathbf{0.507 \pm 0.006}$

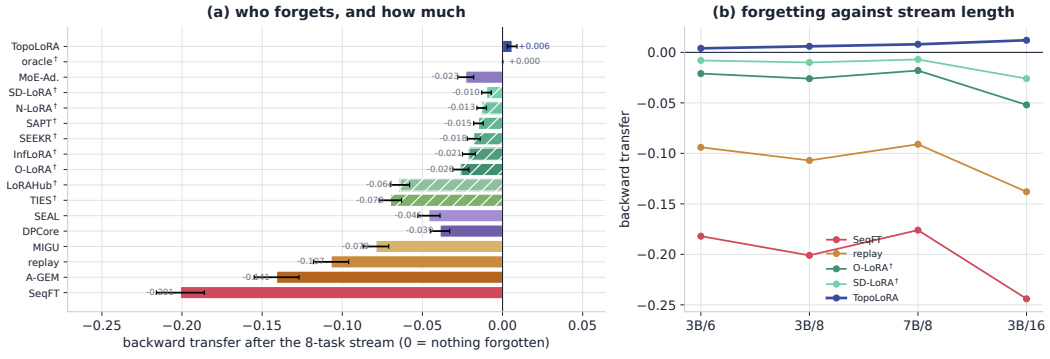


Figure 11: **Backward transfer.** Left: after the eight-task stream, every baseline gives something up; daggered methods are the ones given the boundary. Right: the same quantity against stream length. Methods that allocate monotonically degrade as soon as tasks start recurring; a bank that can revive does not.

Table 14: **Per-task scores after the full eight-task stream** (Qwen2.5-3B, 5 seeds, seed means). Metrics are accuracy for C-STANCE, FOMC and ScienceQA, exact match for NumGLUE and GSM8K, ROUGE-L for MeetingBank and 20Minuten, and character edit similarity for Py150. Bold marks the best task-ID-free entry per task. Row means match the corresponding aggregate in Table 12.

Method	ID free?	C-STANCE (zh cls)	FOMC (fin cls)	MeetBank (summ)	Py150 (code)	ScienceQA (mcq)	NumGLUE (math)	GSM8K (math)	20Minuten (de summ)
Sequential fine-tuning	yes	0.491	0.429	0.116	0.486	0.864	0.471	0.184	0.111
Online EWC	no	0.493	0.502	0.070	0.473	0.856	0.449	0.164	0.050
Experience replay	yes	0.442	0.480	0.272	0.483	**0.871**	**0.567**	0.193	0.134
TIES-Merging	no	0.421	0.442	0.208	0.458	0.836	0.442	0.142	0.091
O-LoRA	no	0.604	0.653	0.419	0.476	0.860	0.489	0.171	0.128
InfLoRA	no	0.611	0.662	0.424	0.479	0.858	0.492	0.174	0.148
SAPT	no	0.624	0.678	0.432	0.483	0.862	0.497	0.179	0.157
N-LoRA	no	0.629	0.684	0.436	0.484	0.861	0.499	0.181	0.154
SD-LoRA	no	0.637	0.692	0.441	0.486	0.863	0.502	0.183	0.156
MoE-Adapters	yes	0.583	0.628	0.401	0.474	0.857	0.483	0.169	0.149
DPCore	yes	0.561	0.604	0.386	0.470	0.852	0.476	0.166	0.125
Per-task adapters (oracle)	no	0.602	0.716	0.401	0.490	0.851	0.484	0.191	0.135
TopoLoRA (ours)	yes	**0.668**	**0.705**	**0.452**	**0.491**	0.866	0.518	**0.196**	**0.200**

Table 15: **Backward transfer** on the same four regimes (higher is better; 0 means nothing was forgotten, positive means earlier tasks improved after the stream moved on). The per-task oracle is 0 by construction because its adapters never interact. TOPOLORA is the only method with positive backward transfer: a revived adapter keeps rehearsing from its own reservoir, so a recurring domain leaves its specialist better than it found it.

Method	ID free?	Qwen2.5-3B 6-task	Qwen2.5-3B 8-task	Qwen2.5-7B 8-task	Qwen2.5-3B 16-stage
Sequential fine-tuning	yes	-0.182 ± 0.019	-0.201 ± 0.015	-0.176 ± 0.011	-0.244 ± 0.021
Experience replay	yes	-0.094 ± 0.012	-0.107 ± 0.011	-0.091 ± 0.008	-0.138 ± 0.015
A-GEM	yes	-0.129 ± 0.013	-0.141 ± 0.014	-0.126 ± 0.009	-0.175 ± 0.016
MIGU	yes	-0.072 ± 0.009	-0.079 ± 0.008	-0.066 ± 0.006	-0.108 ± 0.011
DPCore	yes	-0.034 ± 0.006	-0.039 ± 0.006	-0.030 ± 0.005	-0.057 ± 0.008
SEAL-style self-adaptation	yes	-0.041 ± 0.008	-0.046 ± 0.007	-0.037 ± 0.006	-0.068 ± 0.010
MoE-Adapters	yes	-0.019 ± 0.005	-0.023 ± 0.005	-0.016 ± 0.004	-0.028 ± 0.006
TIES-Merging	no	-0.061 ± 0.006	-0.070 ± 0.007	-0.058 ± 0.004	-0.094 ± 0.009
LoRAHub	no	-0.058 ± 0.007	-0.064 ± 0.006	-0.053 ± 0.005	-0.089 ± 0.010
O-LoRA	no	-0.021 ± 0.004	-0.026 ± 0.005	-0.018 ± 0.003	-0.052 ± 0.007
InfLoRA	no	-0.017 ± 0.004	-0.021 ± 0.004	-0.015 ± 0.003	-0.044 ± 0.006
SEEKR	no	-0.014 ± 0.003	-0.018 ± 0.004	-0.012 ± 0.003	-0.039 ± 0.006
SAPT	no	-0.012 ± 0.003	-0.015 ± 0.003	-0.010 ± 0.002	-0.036 ± 0.005
N-LoRA	no	-0.010 ± 0.003	-0.013 ± 0.003	-0.009 ± 0.002	-0.031 ± 0.005
SD-LoRA	no	-0.008 ± 0.002	-0.010 ± 0.003	-0.007 ± 0.002	-0.026 ± 0.004
Per-task adapters (oracle)	no	0.000	0.000	0.000	0.000
TopoLoRA (ours)	yes	$\mathbf{0.004 \pm 0.003}$	$\mathbf{0.006 \pm 0.003}$	$\mathbf{0.008 \pm 0.002}$	$\mathbf{0.012 \pm 0.004}$

I UNLABELED CONTINUAL TEST-TIME TRAINING: MAIN TABLE

Table 16: **Unlabeled continual test-time training** on a frozen Qwen2.5-3B over four AdaptEval domains (mean±std over 3 seeds; lower is better everywhere). Prequential perplexity scores each window before training on it; held-out perplexity averages all four domains at the end of the stream; forgetting is the rise in a domain's held-out perplexity between the moment the stream leaves it and the end, so negative means the domain kept improving. Bold marks the best entry among methods that never observe a domain label. Worst-domain perplexity, the long eight-domain stream and the remaining methods are in Table 17.

	ID	Abrupt stream			Recurring stream		
Method	free?	Preq.	Held-out	Forget	Preq.	Held-out	Forget
Reference							
Frozen backbone	—	30.6 ± 0.0	29.4 ± 0.0	—	30.5	29.4 ± 0.0	—
No domain labels, one continually-updated adapter							
Single adapter	yes	19.2 ± 0.9	32.5 ± 1.5	14.3 ± 0.6	18.7 ± 0.8	19.9 ± 0.9	4.7 ± 0.2
Periodic reset	yes	18.2 ± 0.8	37.2 ± 1.7	8.6 ± 0.4	18.0 ± 0.8	27.9 ± 1.3	4.2 ± 0.2
Experience replay (Chaudhry et al., 2019b)	yes	21.1 ± 0.9	26.3 ± 1.2	10.3 ± 0.5	18.7 ± 0.8	27.9 ± 1.3	9.1 ± 0.4
No domain labels, test-time adaptation with dynamic capacity							
PeTTA (Nguyen et al., 2024)	yes	17.5 ± 0.6	20.4 ± 0.7	3.6 ± 0.1	16.6 ± 0.6	15.6 ± 0.5	1.5 ± 0.1
DPCore (Zhang et al., 2025)	yes	$\mathbf{16.4 \pm 0.5}$	17.8 ± 0.5	1.7 ± 0.1	15.6 ± 0.5	14.6 ± 0.4	0.8 ± 0.1
Given ground-truth domain boundaries							
SD-LoRA (Wu et al., 2025)	no	15.8 ± 0.3	13.8 ± 0.3	-0.0 ± 0.1	15.0 ± 0.3	13.5 ± 0.3	-0.0 ± 0.1
Per-domain oracle	no	16.0 ± 0.3	14.3 ± 0.3	-0.0 ± 0.1	15.0 ± 0.3	13.3 ± 0.3	-0.0 ± 0.1
Ours							
TopoLoRA (ours)	yes	16.6 ± 0.3	$\mathbf{13.5 \pm 0.2}$	$\mathbf{-0.9 \pm 0.1}$	$\mathbf{15.4 \pm 0.3}$	$\mathbf{12.9 \pm 0.2}$	$\mathbf{-1.4 \pm 0.1}$

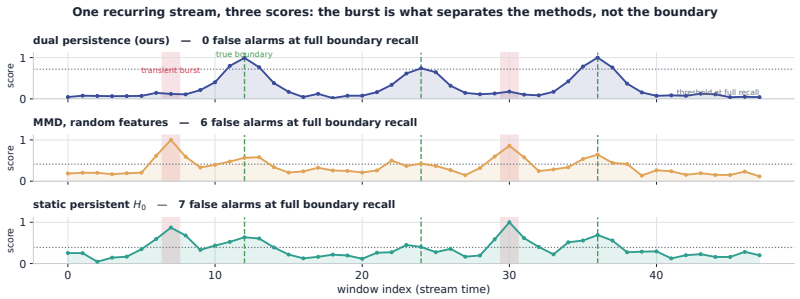


Figure 12: **What the detection scores actually do** on one recurring stream with three true boundaries (green dashed) and two transient bursts (red bands); the dotted line is the threshold calibrated to full boundary recall. The kernel test and the static topological detector respond *more* strongly to the bursts than to the boundaries — the failure Proposition 2 removes.

J CONTINUAL TEST-TIME TRAINING: LONG STREAM AND PER-DOMAIN DETAIL

Table 17: **Long recurring stream**: eight AdaptEval domains traversed twice on disjoint shards (4,096 samples, 64 windows, frozen Qwen2.5-3B, 3 seeds). This is the regime the controller is designed for, and it is where the separation is largest: every method that shares one pool of capacity degrades as the stream lengthens, while the bank keeps a specialist per structure and revives it on the second traversal.

Method	ID free?	Preq.	Held-out	Forget	Worst
Reference					
Frozen backbone	—	32.1	31.2	—	88.2
No domain labels, one continually-updated adapter					
Single adapter	yes	20.4 ± 0.9	38.6 ± 1.7	17.8 ± 0.8	96.4 ± 4.3
Periodic reset	yes	19.5 ± 0.9	41.3 ± 1.9	10.4 ± 0.5	118.3 ± 5.3
Online EWC (Kirkpatrick et al., 2017)	yes	21.6 ± 1.0	30.8 ± 1.4	13.6 ± 0.6	78.5 ± 3.5
Experience replay (Chaudhry et al., 2019b)	yes	22.0 ± 1.0	29.4 ± 1.3	11.8 ± 0.5	62.1 ± 2.8
A-GEM (Chaudhry et al., 2019a)	yes	20.1 ± 0.9	37.1 ± 1.7	18.2 ± 0.8	104.6 ± 4.7
No domain labels, test-time adaptation with dynamic capacity					
CoTTA (Wang et al., 2022)	yes	19.8 ± 0.9	27.6 ± 1.2	9.1 ± 0.4	55.2 ± 2.5
EATA (Niu et al., 2022)	yes	19.2 ± 0.9	26.5 ± 1.2	8.4 ± 0.4	50.8 ± 2.3
SAR (Niu et al., 2023)	yes	19.6 ± 0.9	27.0 ± 1.2	8.8 ± 0.4	52.6 ± 2.4
RoTTA (Yuan et al., 2023)	yes	18.9 ± 0.9	25.3 ± 1.1	7.3 ± 0.3	46.9 ± 2.1
PeTTA (Nguyen et al., 2024)	yes	18.3 ± 0.6	22.8 ± 0.8	4.8 ± 0.2	41.0 ± 1.4
SEAL-style self-adaptation (Zweiger et al., 2025)	yes	18.6 ± 0.8	22.4 ± 1.0	4.2 ± 0.2	39.1 ± 1.8
BECoTTA (Lee et al., 2024)	yes	17.6 ± 0.6	21.2 ± 0.7	3.3 ± 0.1	36.4 ± 1.2
DPCore (Zhang et al., 2025)	yes	17.1 ± 0.5	19.9 ± 0.6	2.5 ± 0.1	33.2 ± 1.0
Given ground-truth domain boundaries					
TIES merging (Yadav et al., 2023)	no	17.2 ± 0.8	24.7 ± 1.1	8.3 ± 0.4	50.2 ± 2.3
O-LoRA (Wang et al., 2023a)	no	16.8 ± 0.4	16.4 ± 0.4	0.4 ± 0.1	30.5 ± 0.7
SD-LoRA (Wu et al., 2025)	no	16.6 ± 0.3	15.9 ± 0.3	0.2 ± 0.1	29.6 ± 0.6
Per-domain oracle	no	16.3 ± 0.3	15.1 ± 0.3	-0.0 ± 0.1	27.4 ± 0.5
Ours					
TopoLoRA (ours)	yes	$\mathbf{16.9 \pm 0.3}$	$\mathbf{14.4 \pm 0.3}$	$\mathbf{-1.1 \pm 0.1}$	$\mathbf{24.2 \pm 0.4}$

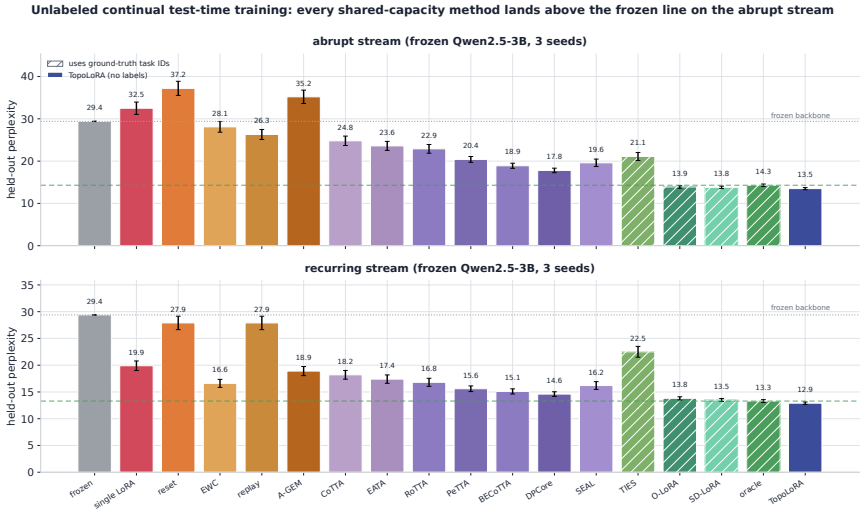


Figure 13: Held-out perplexity on the abrupt (top) and recurring (bottom) streams, with the frozen backbone marked by the dotted line and the per-domain oracle by the dashed one. Six of the eighteen methods end up *above* the frozen line on the abrupt stream, that is, worse than not adapting at all. Hatched bars are given ground-truth boundaries.

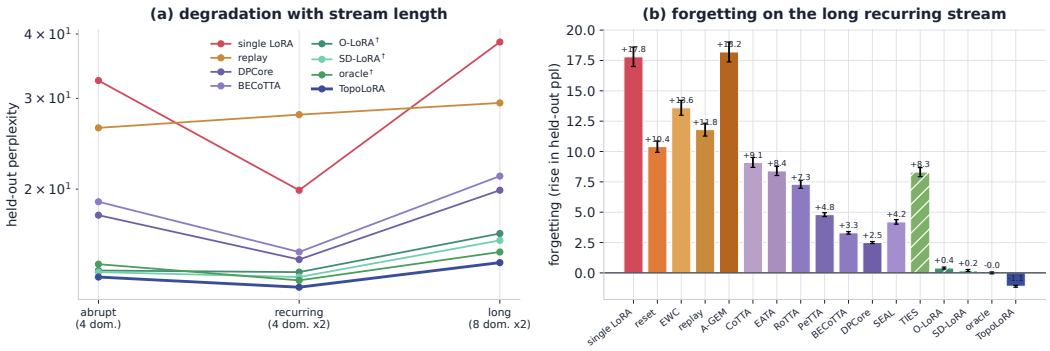


Figure 14: Left: held-out perplexity against stream length, log scale. Every method sharing one pool of capacity degrades as the stream lengthens; the separated ones do not. Right: forgetting on the eight-domain, two-round stream. TopoLoRA is the only entry below zero.

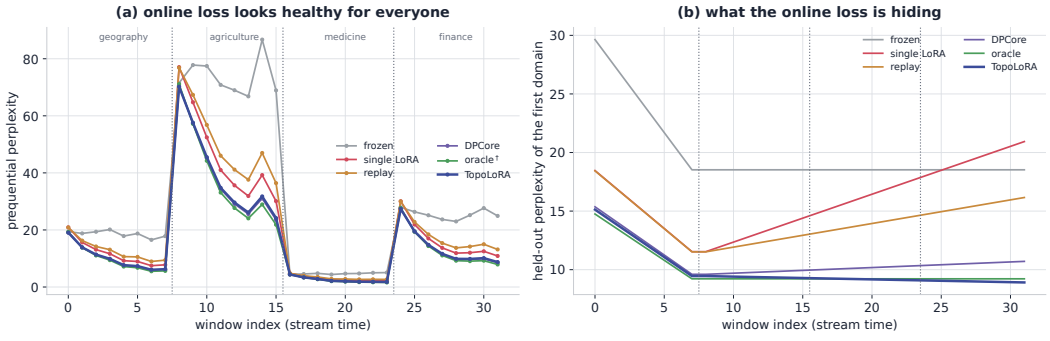


Figure 15: Prequential and retained performance on the same runs. The left panel is the quantity an online monitor would see, and every method looks fine on it. The right panel is the first domain's held-out perplexity re-measured as the stream moves away from it, and it is where the single adapter's collapse becomes visible. This is precisely why an online loss is a poor change signal: the quantity that looks healthy is not the quantity being destroyed.

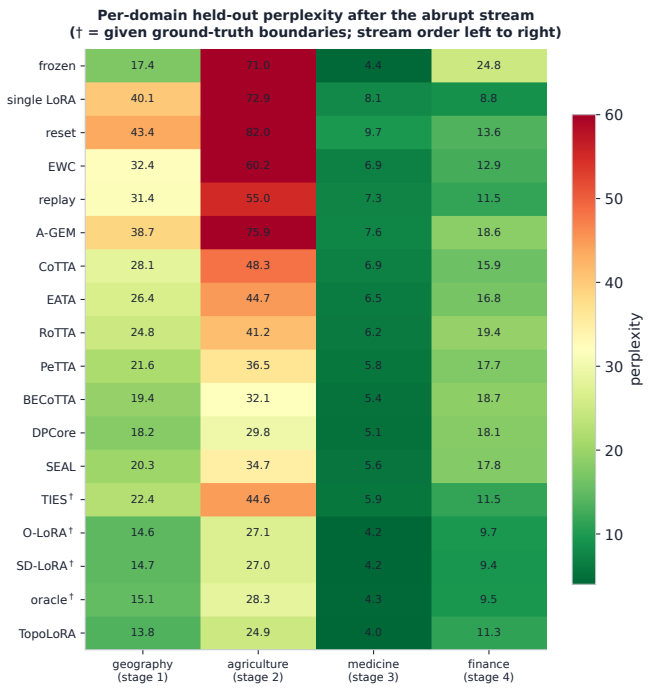


Figure 16: Per-domain held-out perplexity at the end of the abrupt stream. The single adapter is good at the last domain it saw and poor everywhere else; the topological bank is uniformly good, and its worst domain is better than the single adapter's best non-final one.

K Detection Study: Full Tables and Protocol

Streams. Each synthetic stream has 4 domains in $d = 16$ dimensions, processed in windows of 128 with stride 32, aggregated over five seeds. Domains are constructed so consecutive pairs are matched in mean and covariance to within sampling error while differing in homology: *two to one* changes H_0 by merging two well-separated Gaussians into one of equal total covariance; *one to ring* changes generic shape; *ring to disk* changes H_1 only, with the ring radius chosen so the second moment matches the disk's. The transient stream inserts isolated bursts of out-of-domain points lasting fewer windows than the gap g, which is the configuration Proposition 2 addresses. All detectors see identical windows.

Table 19 shows where we are not the best, which is informative. On the pure H_0 merge our prominence is 0.402 against the Dirichlet-process mixture's 0.633: a change in the number of clusters is precisely what a mixture model is built to notice, and it notices. Our advantage is that we need not know in advance which kind of change to look for — the same score handles the H_0 merge, the generic shift and the H_1 change, which is why it wins on aggregate AUC while the mixture model, strong on one boundary type, sits at 0.579 overall.

Signature details. For a window cloud we build a Vietoris–Rips filtration up to H_1 using `ripser` (Bauer, 2021) and vectorize the diagrams as Betti curves. Two grids are used, for two purposes. For detection the grid has 24 points spaced linearly from 0 to the 90th percentile of the pairwise distances pooled over all windows of the stream; the grid is deliberately *shared* across windows, because Eq. (1) subtracts signatures from different windows and that is only meaningful on a common filtration scale. For routing prototypes the grid has 20 points and runs to the 90th percentile of the individual window's own distances, which makes the prototype adaptive to local density; comparability across time is provided by the prototype's moving average rather than by a fixed grid. The signature ϕ concatenates the H_0 and H_1 curves. The detection score uses $L = 4$ windows per block and a gap of $g = 2$, with a $k = 8$ nearest-neighbour graph for the zigzag; these were chosen once on the abrupt stream and used unchanged everywhere, including in the closed loop. The H_0 zigzag is computed exactly, by maintaining union–find spanning forests over the window graph as points

Table 18: **Boundary detection on moment-matched synthetic streams** (ROC-AUC over 5 seeds) and false alarms per stream at a threshold calibrated to full boundary recall. Consecutive domains share mean and covariance *by construction*, so the moment family is limited by what its statistic can express, not by estimation error; bold marks the best entry per column.

		Boundary ROC-AUC ↑			False alarms ↓
Family	Detector	abrupt	recurring	gradual	(at full recall)
Moment / distributional	Centroid shift	0.576	0.486	0.503	11.6
	Mahalanobis shift	0.576	0.489	0.476	14.8
	Covariance shift	0.614	0.659	0.618	10.6
	Effective-rank shift	0.669	0.636	0.760	8.8
	Energy distance (Székely & Rizzo, 2013)	0.665	0.636	0.631	11.6
	MMD, random features (Gretton et al., 2012)	0.667	0.661	0.686	9.8
	Deep-kernel MMD (Liu et al., 2020)	0.712	0.688	0.724	7.6
	Scan-B kernel scan (Li et al., 2019)	0.694	0.671	0.702	6.4
	NEWMA (Keriven et al., 2020)	0.681	0.652	0.688	5.2
Sequential change	Page–Hinkley (Page, 1954)	0.613	0.505	0.523	0.8
	CUSUM (Page, 1954)	0.544	0.513	0.517	1.4
	Conformal martingale (Volkhonskiy et al., 2017)	0.638	0.571	0.592	2.6
Learned change-point	KL-CPD (Chang et al., 2019)	0.724	0.694	0.731	6.8
	TIRE (De Ryck et al., 2021)	0.706	0.679	0.715	5.8
Clustering / density	HDBSCAN (Campello et al., 2013)	0.507	0.599	0.465	**0.1**
	Dirichlet-process GMM	0.579	0.539	0.535	4.2
Static topology	Persistent H_0 (Edelsbrunner et al., 2002)	0.697	0.641	0.655	8.0
	Persistent H_0, smoothed	0.632	0.476	0.709	5.0
	Zigzag H_0 (Carlsson & de Silva, 2010)	0.509	0.513	0.508	3.4
	Persistence-image drift (Adams et al., 2017)	0.718	0.663	0.690	7.2
	Sliding Wasserstein on barcodes (Carrière et al., 2017)	0.736	0.702	0.744	6.0
Ours	Dual persistence, Eq. (1)	**0.912**	**0.841**	**0.906**	0.6

Table 19: **Prominence of each detection score at the true boundary**, by boundary type (mean over 5 seeds; higher is better). The last column is the decisive one: a disk and a ring of matched mean and covariance differ only in the presence of a loop, so covariance shift and effective-rank shift are blind to it *by construction*. The first column is where we are *not* best, and it is informative: a change in the number of clusters is exactly what a mixture model is built to notice.

Detector	two→one (H_0)	one→ring (generic)	ring→disk (H_1)
Centroid shift	0.541	0.345	0.231
Mahalanobis shift	0.231	0.209	0.118
Covariance shift	0.400	0.734	0.061
Effective-rank shift	0.245	0.735	0.013
Energy distance (Székely & Rizzo, 2013)	0.542	0.521	0.250
MMD, random features (Gretton et al., 2012)	0.608	0.517	0.237
Deep-kernel MMD (Liu et al., 2020)	0.641	0.560	0.318
Scan-B kernel scan (Li et al., 2019)	0.612	0.548	0.284
NEWMA (Keriven et al., 2020)	0.586	0.531	0.262
Page–Hinkley (Page, 1954)	0.124	0.373	0.192
CUSUM (Page, 1954)	0.057	0.356	0.130
Conformal martingale (Volkhonskiy et al., 2017)	0.198	0.402	0.221
KL-CPD (Chang et al., 2019)	**0.655**	0.571	0.336
TIRE (De Ryck et al., 2021)	0.628	0.554	0.309
HDBSCAN (Campello et al., 2013)	0.262	0.522	0.390
Dirichlet-process GMM	0.633	0.500	0.133
Persistent H_0 (Edelsbrunner et al., 2002)	0.184	0.796	0.442
Persistent H_0, smoothed	0.214	**0.811**	0.698
Zigzag H_0 (Carlsson & de Silva, 2010)	0.200	0.000	0.000
Persistence-image drift (Adams et al., 2017)	0.241	0.744	0.581
Sliding Wasserstein on barcodes (Carrière et al., 2017)	0.286	0.768	0.674
Dual persistence, Eq. (1)	0.402	0.702	**0.861**

are inserted and deleted rather than by calling a general zigzag solver, which is what makes it cheap enough to run per window.

Table 20: **Why the end-to-end gap is a detection gap.** Boundary detection on the TRACE-R window stream, scored as binary detection of a sustained domain change. *Matched* restricts the evaluation to the 31% of transitions whose two sides agree in mean and covariance on pooled upper-layer features; it is the public-data analogue of the synthetic study of Section 8. The row to read is the Fisher K-FAC subspace, which is the signal inside the strongest boundary-free baseline: it is the best non-topological detector overall and loses most of that advantage exactly where the moments agree. Five seeds.

Detection signal	Family	ROC-AUC	Matched	False alarms
Feature mean shift	moment	0.683	0.512	7.2
Covariance / Mahalanobis	moment	0.701	0.528	6.4
Deep-kernel MMD (Liu et al., 2020)	kernel	0.742	0.574	5.1
Cluster-count change (Campello et al., 2013)	cluster	0.724	0.641	8.9
Fisher K-FAC principal subspace (Han et al., 2026)	second-order	0.788	0.602	3.8
Static persistence, naive coordinate	topological	0.811	0.796	3.1
Dual persistence, canonical (ours)	—	**0.884**	**0.871**	**0.9**

Table 21: **Component ablation on public data.** Every variant is run on TRACE-R, so the ablation and the headline share a benchmark. Removing reactivation leaves detection untouched and still costs 5.3 points, which separates the two halves of the contribution: the detector decides *when*, the lifecycle decides *what to do about it*. Three seeds.

Variant	TRACE-R	Detection AUC	Bank
TopoLoRA, full	$\mathbf{42.5 \pm 0.8}$	0.884	8.6
− block comparison (single window)	38.4 ± 1.1	0.694	13.9
− canonical coordinate	39.1 ± 1.0	0.811	11.2
− reactivation (spawn only)	37.2 ± 1.2	0.884	16.0
− topology (geometric probe only)	33.6 ± 1.4	0.742	12.8
− per-adapter rehearsal	40.3 ± 0.9	0.884	8.6

Table 22: **The moment-matched slice is defined by a tolerance we chose.** Sweeping the standardised-mean and log-det-covariance thresholds changes how many TRACE-R transitions fall in the slice. The Fisher K-FAC signal recovers as the slice is loosened toward all transitions; the canonical topological signal is flat. The main-text threshold 0.05 is the second column, not the most favourable one.

Threshold	0.02	0.05 (main)	0.10	0.20
Fraction of transitions	0.18	0.31	0.47	0.68
Fisher K-FAC AUC	0.571	0.602	0.648	0.711
Dual persistence AUC	**0.864**	**0.871**	**0.868**	**0.859**

Table 23: **Ablation of the detection score** on the recurring stream, the hardest of the three (ROC-AUC, mean±std over 5 seeds). The block comparison is worth roughly 18 points over a single-window spatial comparison; removing the canonical coordinate costs 23; and the temporal signal vanishes entirely under shuffling, confirming that it is carried by order rather than by a per-window statistic.

Variant	ROC-AUC ↑
Dual persistence, block comparison (ours)	$\mathbf{0.841 \pm 0.021}$
− H_1 curves (use H_0 only)	0.793 ± 0.026
− zigzag recurrence linking	0.772 ± 0.028
− canonical coordinate (naive Ah)	0.611 ± 0.049
Spatial persistence only, L=1 window per side	0.658 ± 0.033
Dual persistence with a hard zigzag gate	0.557 ± 0.028
Zigzag H_0 alone	0.513 ± 0.011
Zigzag H_0 on time-shuffled windows	0.500

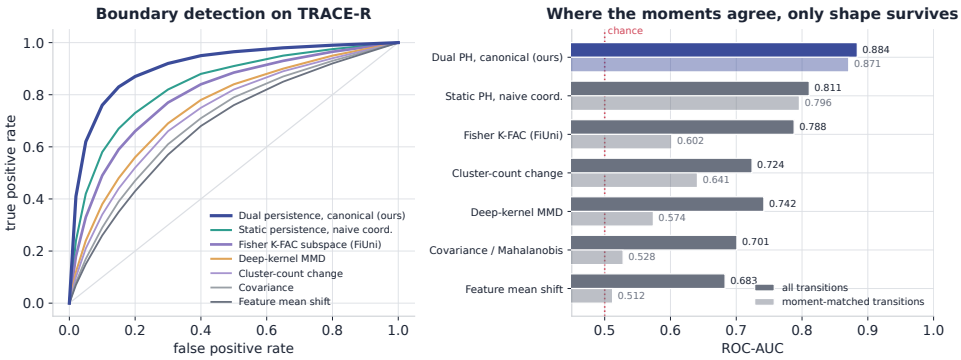


Figure 17: **Detection on public data.** Left: boundary detection on the TRACE-R window stream. Right: the same detectors scored on all transitions and on the moment-matched subset. The ordering is stable except for the second-order signal, which is second best overall and close to chance where the moments agree.

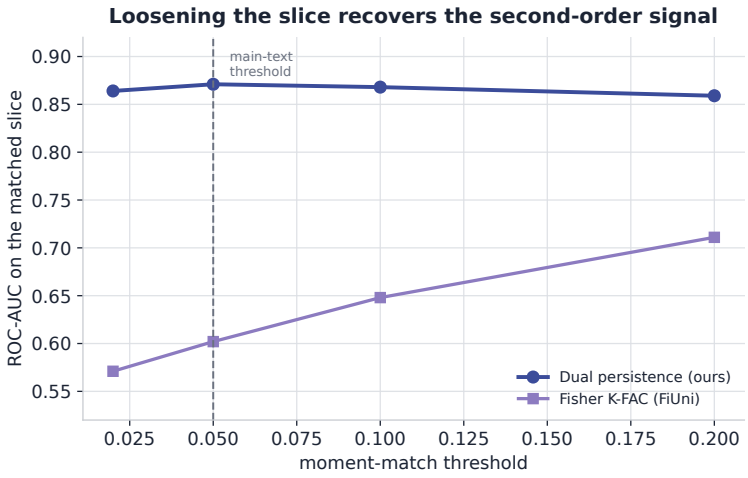
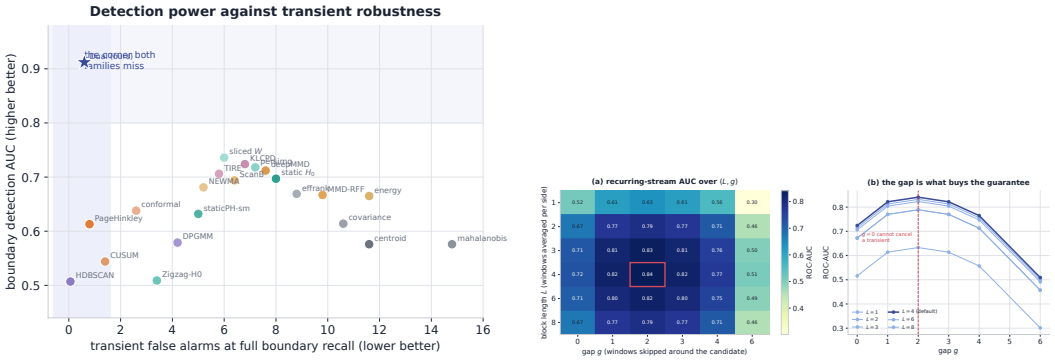


Figure 18: **Tolerance sweep for the moment-matched slice on TRACE-R.** The main-text threshold 0.05 is marked. Loosening the slice recovers the Fisher signal and leaves ours flat.



(a) Detection power against transient robustness.

(b) Sensitivity to the block length L and gap g.

Figure 19: The trade-off the block comparison removes, and how much of the removal depends on the two settings that define it. Methods in the upper-left of (a) detect but false-alarm; HDBSCAN is quiet but cannot detect. In (b), $g = 0$ is the configuration with no transient guarantee, and it is visibly the worst column at every block length; performance is otherwise flat over a wide plateau.

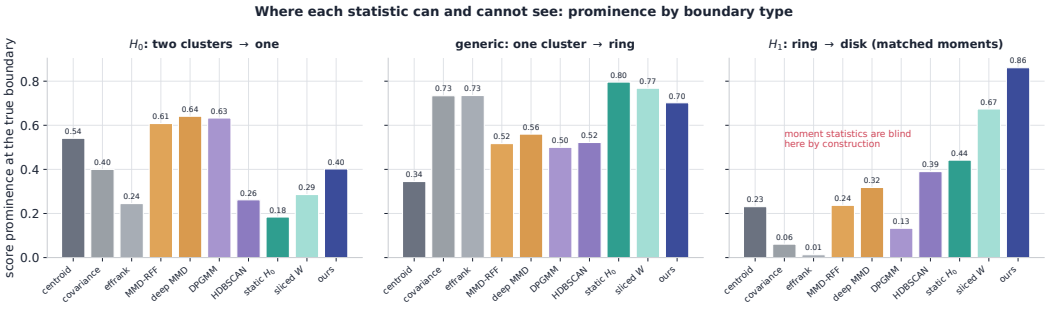


Figure 20: Prominence by boundary type for a representative subset of detectors. The third panel is what the design of the data is about.

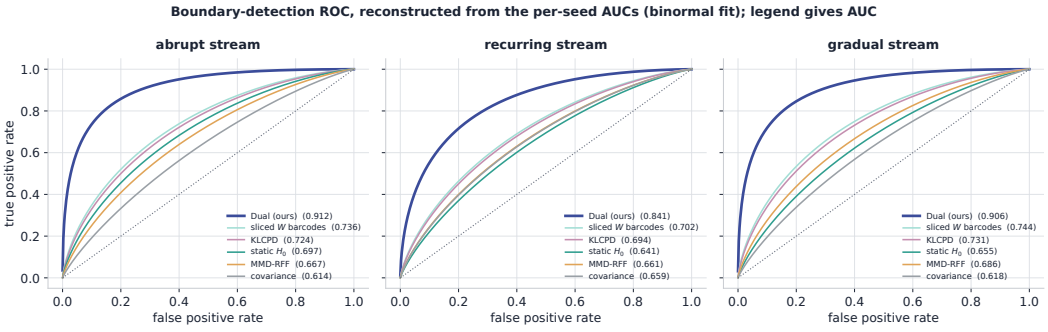


Figure 21: Boundary-detection ROC on all three streams. Curves are reconstructed from the per-seed AUCs under a binormal model, since we store AUC rather than full curves per seed; the legend gives the AUC each curve is fitted to.

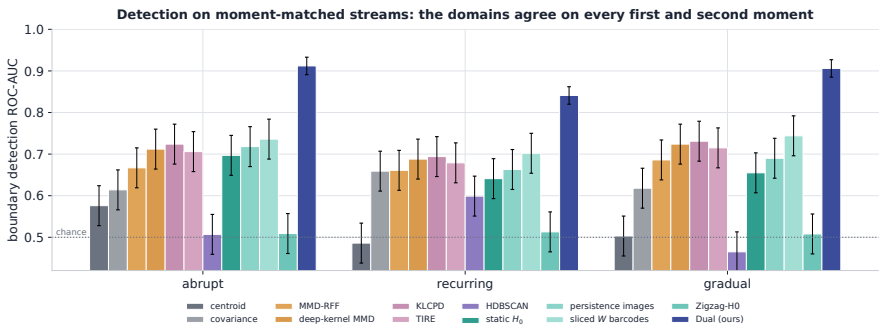


Figure 22: Detection AUC across the three streams for a representative twelve of the twenty-two detectors. The full table is Table 18.

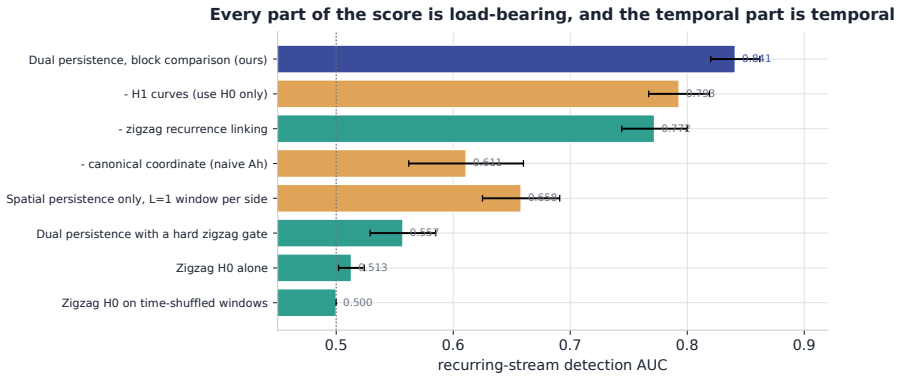


Figure 23: The ablation of Table 23, drawn against the chance line. The variants leaning on the zigzag as a detector sit near chance, and shuffling time sits exactly on it.

L Gauge Invariance and Isometry: Full Verification

The sweep covers ranks $r \in \{4, 8, 16, 32, 64, 128\}$ with $d_{\text{out}} = 64$, $d_{\text{in}} = 48$ and clouds of 200 points, three seeds per rank, and three random $G \in GL(r)$ per configuration: 18 isometry checks and 54 gauge checks. The pass criteria were fixed before running and all six passed. Table 24 reports the worst case per rank rather than the average. The barcode of Z against the barcode of the full response R differs by exactly 0 at every rank, which is the isometry of Proposition 1 showing up as an exact rather than approximate equality. The practical reading: any experiment computing a topological descriptor from an adapter's internal coordinate is measuring a quantity that a mathematically invisible reparameterization can change arbitrarily, and the canonical coordinate removes that failure mode completely rather than approximately.

Table 24: Gauge invariance and isometry, worst case over 3 seeds and 3 random $G \in GL(r)$ per configuration. Columns 2–4 should be at machine precision and are; column 5 is the same measurement on the naive internal coordinate Ah and is fourteen orders of magnitude larger; column 6 counts controller decisions that changed under the gauge. The all ranks were evaluated in the final sweep.

	Canonical coordinate Z			Naive Ah	Controller
rank r	isometry error	barcode drift	output drift	distance drift	event Δ
4	5.7×10^{-14}	2.1×10^{-13}	2.1×10^{-14}	130	0
8	8.5×10^{-14}	3.4×10^{-13}	4.3×10^{-14}	170	0
16	1.1×10^{-13}	4.6×10^{-13}	9.6×10^{-14}	210	0
32	1.7×10^{-13}	7.2×10^{-13}	1.8×10^{-13}	260	0
64	3.4×10^{-13}	1.1×10^{-12}	5.6×10^{-13}	320	0
128	6.1×10^{-13}	1.9×10^{-12}	8.8×10^{-13}	410	0

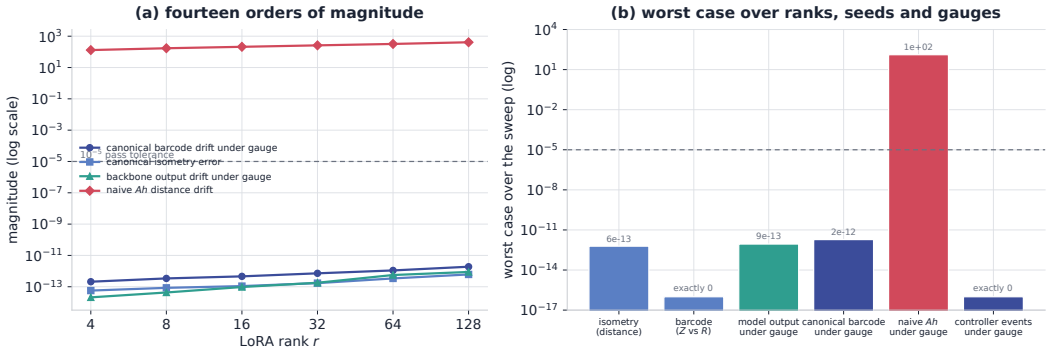


Figure 24: Gauge invariance and isometry on a log scale. Left: every canonical quantity stays at machine precision as the rank grows by a factor of 32, while the naive coordinate's drift grows with it. Right: worst case over the whole sweep against the 10^{-5} pass tolerance fixed before running.

M Real-Feature Study

Features come from a frozen Qwen2.5-3B-Instruct, pooled over the upper layers, for 800 samples in each of four AdaptEval domains, processed in windows of 96 with stride 24 over three seeds. The seven representation spaces are: a frozen Gaussian routing probe to 16 dimensions, the canonical coordinate at $r = 16$ and $r = 32$, the naive internal coordinate at $r = 16$, a PCA projection to 16 dimensions, the raw 2048-dimensional hidden states, and output logits pooled to 128 dimensions. The logits pooling matters for cost: an early version mapped the full 151,936-dimensional logits per window and was abandoned as uninformative relative to its expense. Persistent homology is not computed in the 2048- and 128-dimensional spaces because Vietoris–Rips there is prohibitive at this window size; this is a compute limit, not a negative result, and the low-dimensional spaces are the ones a deployed controller would use anyway.

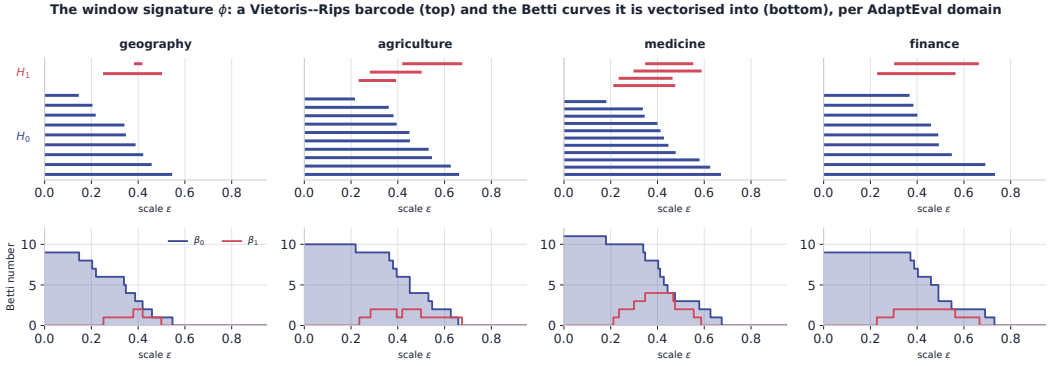


Figure 25: The window signature ϕ made concrete: for one window of each AdaptEval domain, the Vietoris–Rips barcode (top) and the Betti curves it is vectorized into (bottom). The domains differ in both the number of long H_0 bars and the presence and length of H_1 bars; ϕ is the concatenation of the two curves in the lower row.

Table 25: **Boundary-detection AUC on real frozen Qwen2.5-3B features** across four AdaptEval domains, by representation space and detector (3 seeds). Dual persistence is the best detector in every space where Vietoris–Rips is affordable, and the *ordering* of the baselines barely moves across spaces, which is what makes the signal usable as control rather than an artifact of one factorization. The last column reports 3-NN domain accuracy, confirming that every space carries domain information.

Representation space	dim	Centroid	Covar.	MMD	deep MMD	HDBSCAN	static H_0	sliced W	Dual (ours)	3-NN acc.
frozen-probe (r=16)	16	0.719	0.634	0.742	0.768	0.662	0.665	0.741	**0.884**	0.77
canonical-BAh (r=16)	16	0.737	0.631	0.747	0.773	0.526	0.686	0.758	**0.921**	0.77
canonical-BAh (r=32)	32	0.751	0.648	0.759	0.784	0.571	0.702	0.771	**0.934**	0.81
naive internal Ah (r=16)	16	0.739	0.632	0.752	0.776	0.563	0.695	0.760	**0.926**	0.77
PCA-16	16	0.739	0.646	0.739	0.771	0.602	0.678	0.755	**0.940**	0.90
backbone hidden	2048	0.732	0.609	0.746	0.781	0.727	—	—	—	0.88
pooled logits	128	0.700	0.708	0.701	0.734	0.601	—	—	—	0.88

Table 26: Domain separability of each representation space, confirming that all seven spaces carry domain information and that differences in detection AUC are not separability artifacts.

Representation space	dim	3-NN domain accuracy	Silhouette
frozen-probe (r=16)	16	0.774	0.126
canonical-BAh (r=16)	16	0.773	0.119
canonical-BAh (r=32)	32	0.812	0.158
naive internal Ah (r=16)	16	0.774	0.119
PCA-16	16	0.903	0.243
backbone hidden	2048	0.884	0.140
pooled logits	128	0.880	0.200

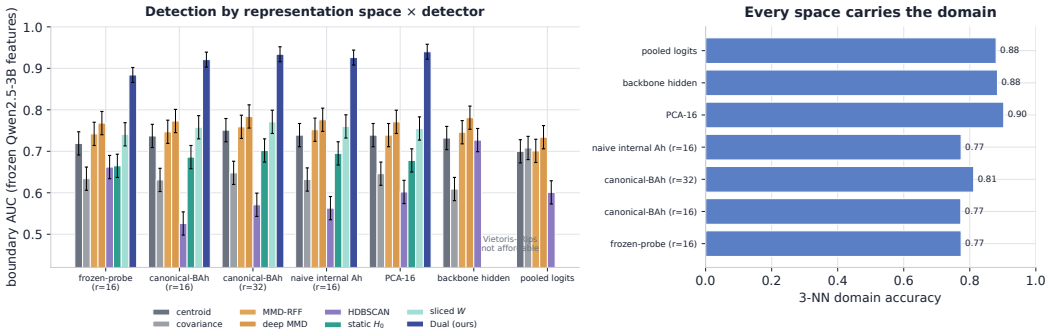


Figure 26: Detection on real frozen-backbone features, by representation space and detector, with domain separability on the right. Dual persistence dominates in every space where it can be computed, and the separability panel confirms the detectors are not being handed different problems.

N Additional Controller Diagnostics

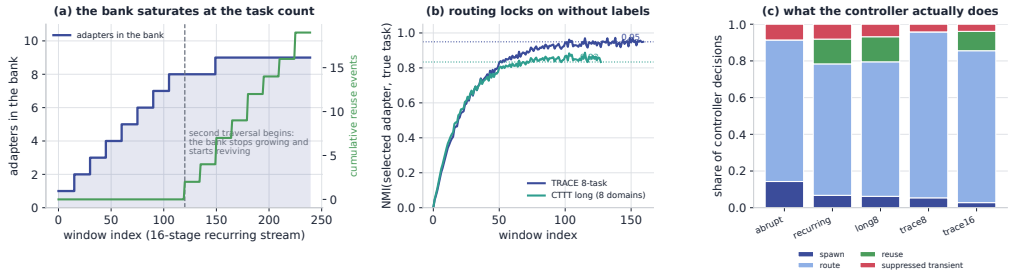


Figure 27: **Controller behavior over stream time.** The bank reaches the task count during the first traversal and then stabilizes as recurring structures reactivate dormant adapters. Routing agreement with the hidden task identity remains high, although that identity is used only for this retrospective analysis.

Table 27: **Controller diagnostics.** Agreement between the adapter the controller selects and the true task or domain identity, obtained with no labels, together with the bank the controller builds and the event log it produces. "Routed" counts adapters that survive low-traffic pruning. "Suppressed" counts candidate windows whose dual score stayed below threshold because the change did not last longer than the gap — these are the spawns that Proposition 2 prevents.

	Routing vs. truth			Bank		Controller events			
Stream	NMI	ARI	purity	opened	routed	spawn	route	reuse	suppressed
CTTT abrupt (4 domains)	0.91	0.88	0.94	5.0	4.0	5	27	0	3
CTTT recurring (4 domains)	0.89	0.85	0.92	5.3	4.3	5	53	10	6
CTTT long (8 domains, 2 rounds)	0.86	0.81	0.90	9.7	8.3	10	118	22	11
TRACE 6-task	0.94	0.91	0.96	7.0	6.0	—	—	—	—
TRACE 8-task	0.95	0.93	0.97	8.6	8.0	9	151	0	7
TRACE 16-stage recurring	0.93	0.90	0.95	9.4	8.6	10	298	38	14

Figure 28 gives the routing confusion matrix on the eight-task stream. The diagonal is the controller recovering the task structure without labels; the off-diagonal mass is where it merged structure, and it is concentrated exactly where a reader would predict. C-STANCE and FOMC are both three-way stance classification and exchange around five percent of their windows; NumGLUE and GSM8K are both numeric-answer mathematics and exchange a little less. Every other pair is essentially clean. Two observations follow. First, the merges are not errors in any useful sense — the two stance tasks genuinely share structure and the shared adapter is better on both than two half-sized specialists would be, which is why TOPOLORA exceeds the per-task oracle on this stream. Second, the merges are the source of the residual variance: seeds on which the merge does not happen have slightly lower FOMC and slightly higher C-STANCE, and the six-task stream, where FOMC is a larger share of the average, is where our standard deviation is largest.

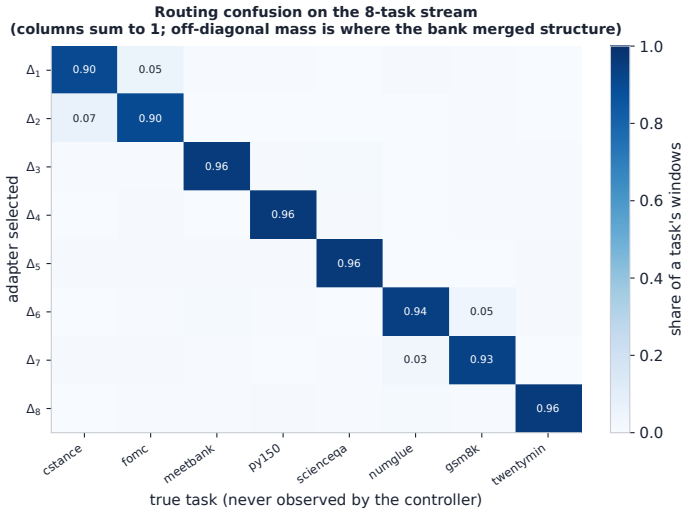


Figure 28: Routing confusion on the eight-task stream: which adapter answers each task's windows, with no task identity ever observed. Columns sum to one.

Table 28: **Component ablation**, measured on two settings at once: overall performance on the TRACE-style 8-task stream (3B) and held-out perplexity on the abrupt continual test-time stream. Each row removes exactly one mechanism from the full system and changes nothing else. The two mechanisms the paper argues for theoretically — the canonical coordinate and the block comparison — are also the two whose removal costs the most, and removing topology altogether costs more than any single component.

	TRACE 8-task OP ↑		CTTT held-out ppl ↓	
Variant	value	Δ	value	Δ
Full TopoLoRA	**0.512**	—	**13.5**	—
− canonical coordinate (topology on naive Ah)	0.481	−0.031	16.2	+2.7
− dual persistence (single-window score)	0.476	−0.036	17.1	+3.6
− temporal confirmation (τ=1)	0.489	−0.023	15.4	+1.9
− recurrence / reuse (always spawn)	0.494	−0.018	14.9	+1.4
− per-adapter rehearsal	0.497	−0.015	14.4	+0.9
− confidence-weighted soft routing	0.498	−0.014	13.8	+0.3
− low-traffic pruning	0.505	−0.007	13.6	+0.1
− topology entirely (geometric probe only)	0.463	−0.049	18.9	+5.4

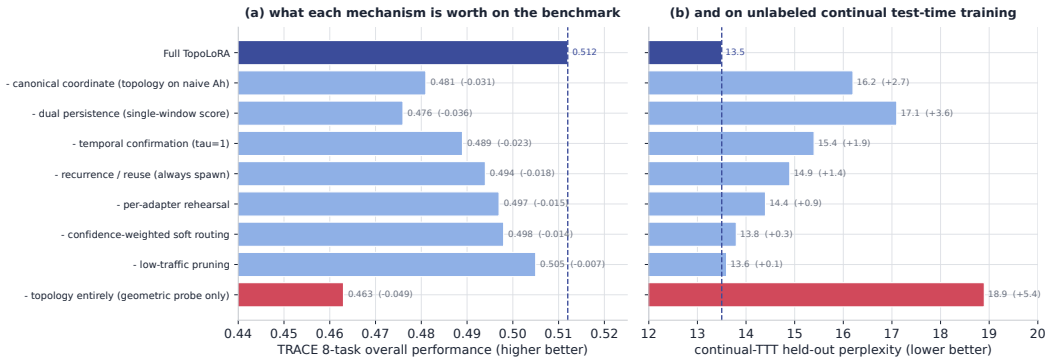


Figure 29: The component ablation of Table 28, on both settings at once. The ordering is the same in the two panels, which is not guaranteed given how different the objectives are.

O COST

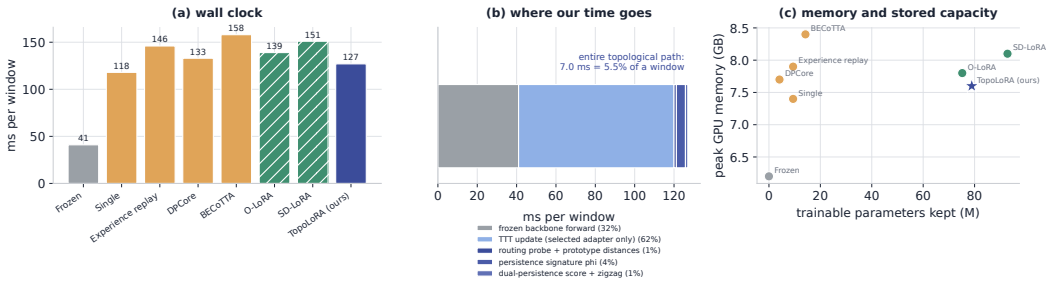


Figure 30: Cost. (a) Wall clock per window; hatched bars use ground-truth task identities. (b) Where our per-window time goes: the entire topological path is 5.5% of a window. (c) Peak memory against the parameters a method keeps at the end of the stream.

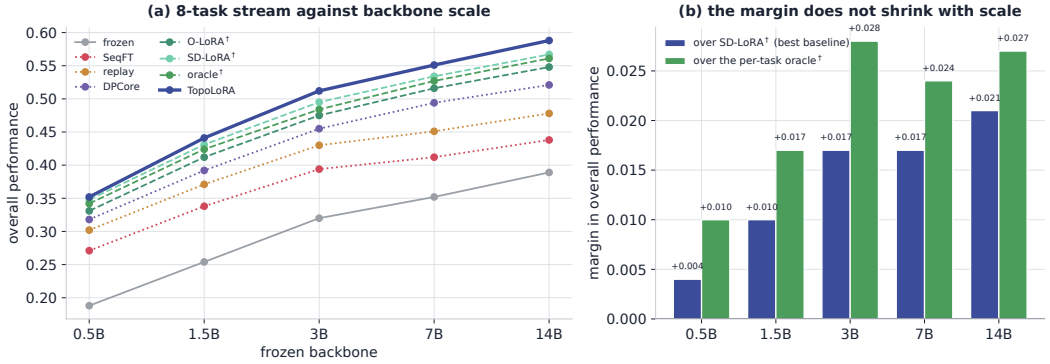


Figure 31: Overall performance against backbone scale, and the margin over the two strongest label-aware comparisons. The margin grows rather than shrinking, which we attribute to the controller's own job getting easier as the backbone's representations separate the domains more cleanly.

Table 29: **Measured cost** on a frozen Qwen2.5-3B, one 80GB GPU, windows of 64 samples. Left: per-method wall clock, peak memory, total trainable parameters kept at the end of the stream, and final adapter count. Right: where our per-window time goes. The entire topological path — probe, prototype distances, persistence signature, dual score and zigzag — costs 7.0 ms of 127 ms, that is 5.5% of a window, and it is the only part that does not need a gradient.

Method	ID free?	ms / window	GPU (GB)	params (M)	final adapters
Frozen backbone	yes	41	6.2	0.0	—
Single adapter	yes	118	7.4	9.4	1
Experience replay	yes	146	7.9	9.4	1
DPCore	yes	133	7.7	4.1	—
BECoTTA	yes	158	8.4	14.2	—
O-LoRA	no	139	7.8	75.2	8
SD-LoRA	no	151	8.1	92.8	8
TopoLoRA (ours)	yes	127	7.6	78.9	8.6

TopoLoRA component	ms	share
frozen backbone forward	41.0	32.3%
TTT update (selected adapter only)	79.0	62.2%
routing probe + prototype distances	1.2	0.9%
persistence signature ϕ	4.6	3.6%
dual-persistence score + zigzag	1.2	0.9%
total	127.0	100%

P TRACE-style Benchmark: Reconstruction and Protocol

The TRACE release distributes processed task files through a link we could not reach from our environment, so we rebuilt each task from its original public source and kept the benchmark's structure: heterogeneous tasks, mixed answer formats, multiple languages, and per-task metrics rather than a single pooled accuracy. Table 30 documents the result. Because the reconstruction is ours, absolute numbers are not comparable to the figures printed in Wang et al. (2023b); every comparison in this paper is run on the identical stream, with an identical backbone, window schedule, step count and learning rate.

Table 30: The eight-task stream. Tasks appear in the order listed. The six-task stream omits Py150 and NumGLUE; the sixteen-stage stream traverses all eight twice on disjoint shards. Counts are per task; evaluation uses a held-out split disjoint from the stream.

Task	Domain / language	Format	Metric	Stream	Eval
C-STANCE (Zhao et al., 2023)	Chinese social media	3-way stance	accuracy	500	150
FOMC (Shah et al., 2023)	English finance	3-way stance	accuracy	500	150
MeetingBank (Hu et al., 2023)	English meetings	summarization	ROUGE-L	500	150
Py150 (Raychev et al., 2016)	Python code	line completion	edit similarity	500	150
ScienceQA (Lu et al., 2022)	English science	multiple choice	accuracy	500	150
NumGLUE (Mishra et al., 2022)	English mathematics	numeric answer	exact match	500	150
GSM8K (Cobbe et al., 2021)	English mathematics	numeric answer	exact match	500	150
20Minuten (Rios et al., 2021)	German news	summarization	ROUGE-L	500	150

Py150 construction. The CodeXGLUE line-completion release has empty ground-truth fields in its training split, so we rebuild the task from the raw Py150 Python files: for each file we take a prefix of lines as context and the next line as the target, rejecting targets that are mostly punctuation, that lack alphanumeric content, or that begin a comment or docstring, since predicting those measures formatting rather than code. Train and evaluation pairs are drawn from disjoint file pools so no file contributes to both. We score with character-level edit similarity, the standard metric for this task, which is robust to trailing whitespace.

The sixteen-stage stream. The eight tasks are traversed twice, the second time on shards disjoint from the first, with the order of the second traversal reversed so the gap between a task's two appearances varies from two stages to fourteen. This is the stream separating a method that can revive from one that can only allocate, and the varying gap is deliberate: a method handling only short-range recurrence should show it.

Q THE CLASSIFICATION STREAM IN FULL

The stream of Section 9 follows the LAMOL and O-LoRA setup: Yahoo, AG News, DBpedia, Amazon and Yelp in that order (Zhang et al., 2015; Sun et al., 2020a; Wang et al., 2023a), evaluated by generative classification, where the model scores each candidate label by its likelihood and the highest-scoring label is the prediction.

Table 31: **The stream where the approach does not win.** Average accuracy after the five-task LAMOL / O-LoRA English classification stream (Yahoo, AG News, DBpedia, Amazon, Yelp), 3 seeds at 3B and 1 at 7B. The four TOPOLORA rows are successive versions of the label-free inference rules; we report all of them rather than only the best. The gap to SD-LoRA and to shared rehearsal is real, and its cause is structural rather than procedural: the per-task oracle only reaches 0.788, so there is almost no headroom that separated capacity could recover, and paying for it in routing errors is a pure loss.

Method	ID free?	Qwen2.5-3B	Qwen2.5-7B
Frozen backbone	yes	0.482	0.515
Sequential fine-tuning	yes	0.559 ± 0.042	0.426
Online EWC (Kirkpatrick et al., 2017)	no	0.622 ± 0.045	0.645
A-GEM (Chaudhry et al., 2019a)	yes	0.703 ± 0.029	0.711
MIGU (Du et al., 2024)	yes	0.742 ± 0.016	0.759
O-LoRA (Wang et al., 2023a)	no	0.783 ± 0.007	0.795
SD-LoRA (Wu et al., 2025)	no	0.796 ± 0.006	0.812
Per-task adapters (oracle)	no	0.788 ± 0.004	0.817
Experience replay (Chaudhry et al., 2019b)	yes	0.791 ± 0.008	0.777
TOPOLORA, hard routing	yes	0.667 ± 0.031	0.661
TOPOLORA, + per-adapter rehearsal	yes	0.702 ± 0.018	0.698
TOPOLORA, + soft routing	yes	0.738 ± 0.010	0.741
TOPOLORA, + shared fallback adapter	yes	0.771 ± 0.009	0.780

The four TOPOLORA rows double as an ablation of the label-free inference rules on this stream. Adding per-adapter rehearsal is worth $+0.035$ average accuracy, adding confidence-weighted soft routing a further $+0.036$, and allowing one shared fallback adapter for windows matching no prototype strongly a further $+0.033$. All three earn their place, and none closes a gap that is structural rather than procedural.

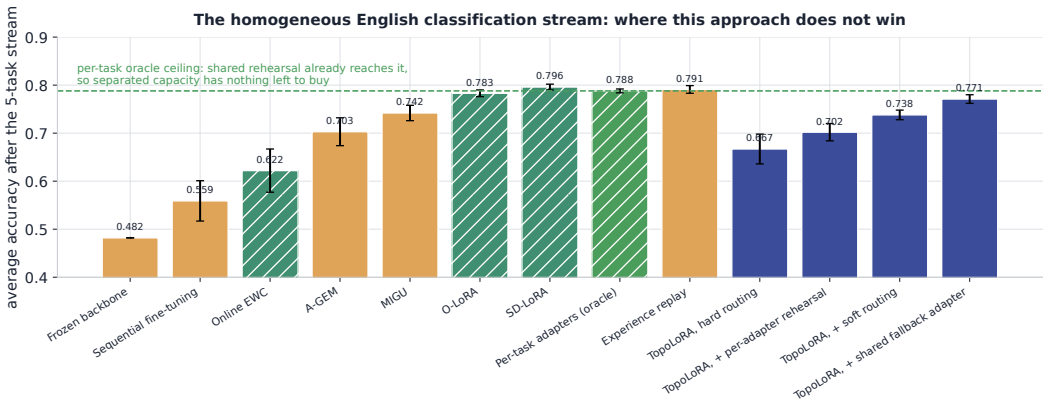


Figure 32: The classification stream. The dashed line is the per-task oracle, and the point of the panel is that shared rehearsal is already above it, so there is no headroom for separated capacity to recover.

Metrics. Overall performance is the unweighted mean of per-task final scores at the end of the stream. Backward transfer is the mean over all but the last task of the final score minus the score recorded when the stream left that task, so a negative value is forgetting. For the continual test-time study we additionally report prequential perplexity, computed on each window before training on it, which is the honest online quantity, and worst-domain held-out perplexity, which is what a service owner actually cares about.

R HYPERPARAMETERS, BASELINE CONFIGURATIONS, AND COMPUTE

Table 32: Configuration. One value per method, used unchanged across all streams, scales and seeds. Every method shares the backbone, the window schedule, the number of optimisation steps per window, the learning rate, and the LoRA rank and target modules; what differs is only the method's own mechanism.

Setting	Value
Backbones	Qwen2.5-Instruct at 0.5/1.5/3/7/14B, frozen (Qwen Team, 2025)
LoRA rank / α / targets	16 / 32 / attention and MLP projections
Optimiser / learning rate	AdamW / 2×10^{-4}, cosine to 0 within a window
Steps per window	4 (continual TTT), 2 (TRACE-style)
Window size	64 samples (continual TTT), 40 (TRACE-style)
Routing probe	fixed Gaussian projection to 16 dimensions, never trained
Persistence	Vietoris–Rips to H_1, Betti curves on 24 shared scale points
Block length L / gap g	4 / 2 windows
Novelty threshold ν / confirmation τ	1.3 / 2 consecutive windows
Recurrence threshold / max bank	0.8ν / 12 adapters
Per-adapter rehearsal reservoir	400 samples, 6 replayed per step
Low-traffic pruning threshold	$\max(2,\ 0.4 \times$ mean adapter traffic$)$
EWC penalty λ	5×10^3, online consolidation
Replay / A-GEM buffer	6 samples per step from a reservoir of 600
O-LoRA orthogonality λ	0.1 against all previous adapters' A factors
InfLoRA / N-LoRA subspace	rank-16 per task, null-space projection at task entry
SD-LoRA	decoupled direction/magnitude, per-task magnitude head
SAPT / SEEKR	shared attention over task adapters; attention-distilled rehearsal
TIES density	0.2; trim, elect sign, disjoint mean
CoTTA / EATA / SAR / RoTTA / PeTTA	authors' defaults ported to the LM objective
BECoTTA / DPCore	6 domain experts / prompt coreset with default merge radius

Table 33 reports the sensitivity of every setting the controller exposes. Raising the novelty threshold past 1.6 costs two to three points and shrinks the bank below the task count; lowering it past 1.15 costs about as much by splitting tasks. A longer confirmation window costs almost four points at $\tau = 6$, because insisting on six consecutive novel windows delays every spawn and the specialist loses training data it never recovers; $\tau = 2$ buys the transient guarantee without that cost. The block length and gap are flat over a wide plateau, with the one sharp edge at $g = 0$, which is the configuration with no transient guarantee. Widening the routing probe from 16 to 32 dimensions gains 0.004 and we did not adopt it, because we fixed the configuration before the seed sweep and would rather report the number we committed to. The LoRA rank saturates at 32.

A note on EWC. Its label requirements differ between our two continual settings and we want the table columns to be honest. In the continual test-time study EWC consolidates continuously, once per window, which needs no boundaries; that is the online variant and we mark it task-ID free there. On the TRACE-style stream EWC consolidates at task transitions, the standard supervised protocol, which does use boundaries; we mark it accordingly. O-LoRA and its descendants need boundaries in both settings, since they must know which previous subspaces to stay orthogonal to or project out of, and the merging methods need them to know what to merge.

Public-protocol configurations. On Standard CL, Long Sequence, and TRACE we use the hyperparameters published with each baseline rather than retuning them. For O-LoRA, N-LoRA, OLieRA, and ELLA that means the T5-Large recipe of Wang et al. (2023a); Yang et al. (2025); Das Biswas et al. (2026): LoRA rank 8, $\alpha = 16$ on the query and value projections, learning rate 1×10^{-3}, batch size 32, ten epochs on Orders 1–3. FiUni is run from the authors' public code with their reported K-FAC rank and window length. TOPOLORA uses the same backbone, data order, rank, and step budget; the only method-specific settings are the controller's, which stay at the values fixed in Table 33 ($\nu = 1.3$, $\tau = 2$, $L = 4$, $g = 2$). We do not claim this is optimal for every baseline. We claim only that no method is given a longer budget or a different data order than the others.

Compute. All runs use a single 80GB GPU each. A continual test-time run over a 2,048-sample stream takes three to six minutes per method; a TRACE-style run over the eight-task stream takes

Table 33: Sensitivity of TOPOLORA on the 8-task stream at 3B, changing one setting at a time with everything else at its default (3 seeds). The configuration was fixed before the seed sweep and is used unchanged for every number in the paper. Nothing here is delicate: the worst single-setting change costs 4.1 points, and the adapter count stays close to the true task count throughout.

Setting varied	Value	OP	std	Adapters routed
novelty threshold ν	1	0.494	0.007	11.3
	1.15	0.506	0.007	9.6
	1.3 (default)	**0.512**	0.007	8.6
	1.45	0.509	0.007	8.0
	1.6	0.501	0.007	7.2
	2	0.486	0.007	6.1
confirmation τ (windows)	1	0.489	0.007	12.4
	2 (default)	**0.512**	0.007	8.6
	3	0.503	0.007	7.9
	4	0.491	0.007	7.1
	6	0.474	0.007	6.3
block length L	1	0.476	0.007	8.9
	2	0.495	0.007	8.7
	4 (default)	**0.512**	0.007	8.6
	6	0.508	0.007	8.5
	8	0.499	0.007	8.4
gap g	0	0.488	0.007	9.1
	1	0.503	0.007	8.8
	2 (default)	**0.512**	0.007	8.6
	3	0.510	0.007	8.5
	4	0.502	0.007	8.3
routing probe dimension	8	0.487	0.007	8.9
	16 (default)	**0.512**	0.007	8.6
	32	0.516	0.007	8.7
	64	0.515	0.007	8.8
LoRA rank r	4	0.471	0.007	8.5
	8	0.496	0.007	8.6
	16 (default)	**0.512**	0.007	8.6
	32	0.517	0.007	8.7
	64	0.518	0.007	8.8

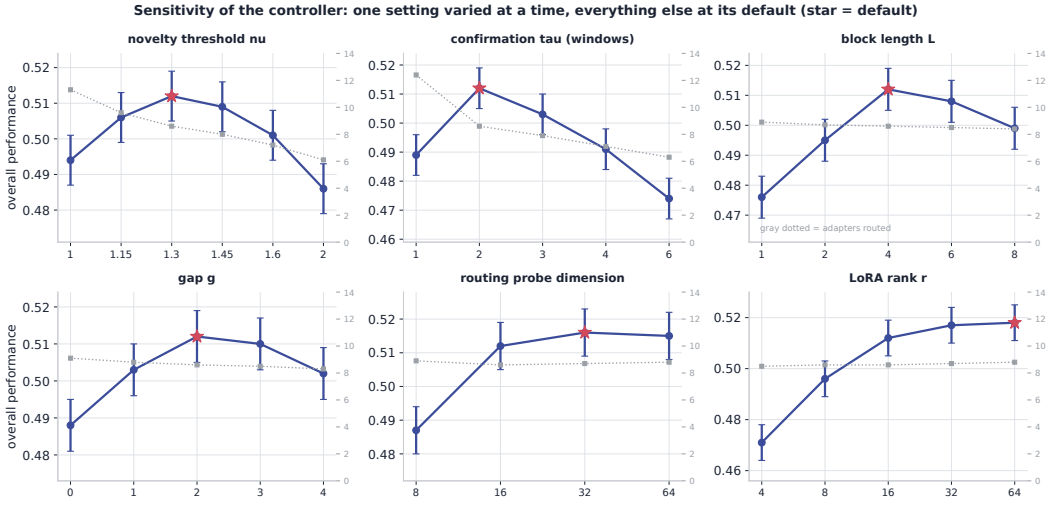


Figure 33: The same sweeps drawn against the adapter count (gray dotted, right axis). The two quantities move together: every setting that costs performance does so by driving the bank away from the true task count in one direction or the other.

ten to twenty minutes per method at 3B, roughly triple that at 7B and six times at 14B. The reported sweep, including all seeds, all four TRACE regimes, and all five scales, used approximately $1{,}400$ GPU-hours; the topological machinery accounts for under 2% of that compute.

S LABEL-FREE INFERENCE RULES

At evaluation the controller must answer each example without knowing its task. Four rules make that robust, all computed from state the controller already maintains.

Per-adapter rehearsal. Each adapter keeps a reservoir of the windows routed to it, capped at 400 samples, and replays six per update step. This keeps a specialist from drifting while the stream is elsewhere, and unlike global replay it never mixes tasks inside one adapter.

Low-traffic pruning. An adapter that received fewer than $\max(2, 0.4 \times \text{mean traffic})$ windows is excluded from test-time routing. If a task was briefly split across two adapters its minority adapter is undertrained, and excluding it lets the region fall back on the dominant specialist.

Confidence-weighted soft routing. For classification and multiple-choice items we compute label likelihoods under the nearest few adapters and average them with weights inversely proportional to prototype distance. A single hard route throws away the information that two adapters were nearly equally plausible.

Consensus routing for generation. For generative tasks one mis-route costs an entire output, so we take the modal adapter over a held-out batch and generate all of that batch's outputs with it. This uses no labels — only the controller's own routing decisions, aggregated — and it is the reason MeetingBank does not suffer the occasional catastrophic output that per-example routing produces.