WHICH HOST VERSIONS SHOULD A REMOVABLE BRANCH LEARN FROM?

Anonymous authors
Paper under double-blind review

ABSTRACT

A branch trained beside a continuing host should be removable without changing the host's training. Optimization-Recoverable Augmentation (ORA) enforces this through complete-state isolation, with conditional guarantees and Qwen/BERT recovery tests. Which host versions should the branch observe before deployment on the final host? In one-block Qwen2.5-0.5B, reader-progress observation (P) exceeds terminal-only training by $4.67/13.87\,pp$ on GSM8K/SQuAD with only the version sequence changed. The advantage survives terminal learning-rate and checkpoint selection: $+5.25/+6.06\,pp$ on a separate estimation half of the historical holdout. Two prospective comparisons with latest-version observation give $40/40$ positive paired differences. The benefit is conditional on the branch's data. Independent branch orders attenuate the gain over latest-version observation. In a preregistered equal-size data control, P exceeds terminal-only by $4.08/13.51\,pp$ when host and branch share a training stream, but trails by $1.86/1.16\,pp$ when host and branch train on group-disjoint data. A retained-seed extension separates order and data contrasts: independent ordering reduces the shared-data advantage, while SQuAD retains a positive gap. The tested full-model recipe also favors terminal-only training. These results establish a repeatable shared-stream observation effect, rather than a general advantage of earlier host versions, and show why observation policies must be evaluated jointly with branch data and the deployment host.

1 INTRODUCTION

Consider evaluating an optional adapter, early exit, or draft head while its host model continues training. Such attachments extend a shared backbone with a selectable route (Schuster et al., 2022; Elhoushi et al., 2024; Houlsby et al., 2019). When such a trial is abandoned, the host should retain exactly the progress it would have made without it. Removing the branch's route does not achieve this: joint training may already have written to shared parameters, optimizer state, randomness, or data position, and rolling back discards later host progress unless that work is replayed. We call the resulting departure from the matched standalone process a *recoverability failure*.

ORA (Optimization-Recoverable Augmentation) removes this failure by isolation: only host updates write host state, and the branch owns its optimizer, randomness, and data position. Under stated sufficient conditions the host's complete training state then equals the standalone run at every step, so abandoning or restoring a branch preserves the continuation (Section 2). Freezing the host, as DeeBERT and Side-Tuning do (Xin et al., 2020; Zhang et al., 2020), gives the strongest branch-enabled system in our main setting. It also leaves the branch-disabled route at its initial quality. Host continuation raises that route's GSM8K/SQuAD score from $3.21/37.42$ to $37.03/55.50$ (Table 65). We study observation when retaining this independent host progress is required, even if the branch is abandoned.

Exact recovery fixes the host's trajectory $s_0, \ldots, s_T$ but not what the branch learns from. A branch deployed on the final host s_T can be trained on s_T alone after the host finishes (terminal-only), on whichever version is newest when it updates (latest-version, H), or on the checkpoint that matches its own training progress (reader-progress, P). All three leave the host untouched; they differ only in the host versions paired with the branch's batches. Which gives the better branch on s_T?

A preregistered intervention answers this in one-block Qwen2.5-0.5B. On ten retained host trajectories per task, P exceeds terminal-only training by $4.67/13.87\,pp$ on GSM8K/SQuAD, with all 20 paired

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

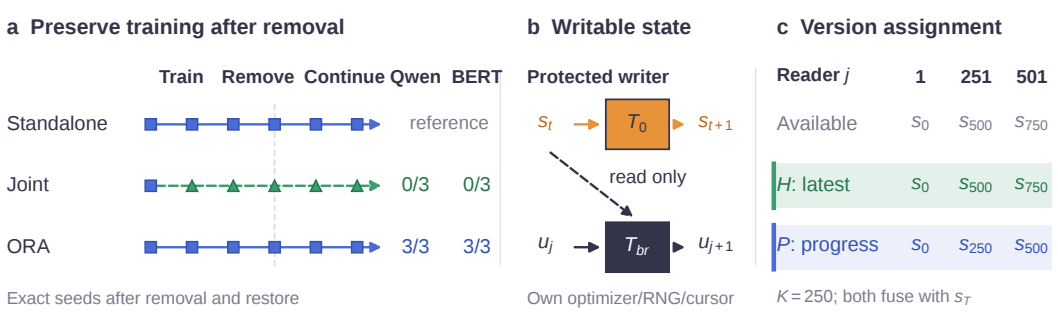

Figure 1: **Removing an attachment does not undo its training influence; ORA preserves the host and leaves observation as a free choice.** (a) Removal and restoration: counts of exact continuations out of three seeds per stack (Tables 17 and 18); shapes compare matched states. (b) Only the host process (writer) changes s; the branch process (reader) updates u with its own optimizer, randomness, and data cursor. (c) H reads the newest published version; P assigns $s_{K\lfloor(j-1)/K\rfloor}$ to branch update j, waiting if it is not yet published. Both deploy on s_T. The schematic illustrates assignments; measured branches lag the host (Figure 2).

differences positive (Table 1). Initialization, recipe, batches, order, random stream, budget, and deployment host are fixed; only the observed versions change. A subsequent four-recipe, five-checkpoint terminal search retains a $+5.25/+6.06\,pp$ advantage for full-budget P on a separate estimation half of the historical holdout (Section 4.2). In two prospective studies it beats latest-version observation in $40/40$ paired comparisons. In the live study, the branch is already usable at host release; descriptive host-time ratios are $1.01/1.02\times$ standalone (Section 4.3).

The gain depends on which examples accompany which versions. Independent branch orders remove most of the $P - H$ gap. A preregistered data-split control goes further: with the same half-size host trajectory and branch budget, P exceeds terminal-only by $4.08/13.51\,pp$ when the branch follows the host's stream, but trails by $1.86/1.16\,pp$ when it learns from group-disjoint data (Section 4.5). Earlier versions therefore help under shared-stream conditions, not uniformly. A retained-seed extension adds shared data with independent order: both order and data contrasts contribute, while SQuAD retains a positive gap after reordering (Appendix S). These controls do not identify a unique mechanism. The advantage also reverses in the tested full-model recipe, and a jointly trained branch transferred to the clean host remains competitive on GSM8K.

Our contributions are:

- **A checkable recovery contract for continuing hosts.** Complete-state isolation with auditable sufficient conditions, predicted first-divergence interventions, and Qwen/BERT abandonment and restoration tests that reproduce the standalone continuation.
- **Observation as a design variable.** A version-only intervention shows that the host versions a branch learns from change its quality on the same final host; two prospective studies repeat the quality gain over latest-version observation. Live execution supplies descriptive timing and release-time quality measurements.
- **Data dependence and limits of the observation effect.** Batch-pairing, equal-size shared/disjoint-data, and fixed-branch deployment-host controls delimit the effect; a full-model test and the transferred-joint alternative bound it.

2 Problem and Recovery Contract

The process to preserve. We study optional learning alongside a designated host that must continue training. Its *matched standalone reference* fixes initialization, protected objective and data order, optimizer and scheduler recipe, arithmetic, randomness, and control while removing optional work. A construction is *optimization-recoverable* through horizon T if its complete protected transition state equals this reference at every protected boundary. Equality includes inputs to the next update,

so abandoning a branch or restoring a checkpoint preserves the subsequent continuation. This is a strong counterfactual target; task-metric equality alone cannot establish it.

Removing a route does not restore that process. Forward removability, $F(x;\theta,\phi,g{=}0) = F_0(x;\theta)$ for protected parameters θ, branch parameters ϕ, and switch g, is not enough. For $F = \theta x + g\phi x$, losses $\mathcal{L}_0 = \frac{1}{2}(\theta-1)^2$ and $\mathcal{L}_{\mathrm{br}} = \frac{1}{2}(\theta+\phi-2)^2$, and $\theta_0 = \phi_0 = 0$, one SGD step of size η gives $\theta_1^0 = \eta$ under $\mathcal{L}_0$ but $\theta_1^{\mathrm{joint}} = 3\eta$ under the summed losses, and setting $g = 0$ leaves the latter in place. Matched BERT early-exit and Qwen LoRA experiments exhibit this failure (Appendix T).

Isolation protects the next transition. The complete protected state in our tested stack is

$$s_t = (\theta_t, b_t, o_t, q_t, a_t, r_t, d_t, c_t), \tag{1}$$

covering parameters, persistent buffers, optimizer, scheduler, precision/scaler, protected randomness, data position, and control. Completeness is relative to a fixed execution stack; some fields may be vacuous. With branch-owned state u_t, write

$$s_{t+1}^0 = T_0(s_t^0; x_t), \qquad (s_{t+1}, u_{t+1}) = T_{\mathrm{opt}}(s_t, u_t; x_t, z_t), \tag{2}$$

Here x_t/z_t contain protected/optional inputs. ORA confines optional writes to u_t, enforcing

$$\pi_s T_{\mathrm{opt}}(s_t, u_t; x_t, z_t) = T_0(s_t; x_t) \quad \text{at every matched step.} \tag{3}$$

Detached gradients alone do not enforce this condition: clipping, optimizer state, RNG, or data position may still couple the processes. Our audit opens each channel separately and checks the predicted first divergence (Appendix T).

Proposition 1 (Sufficient conditions for recoverability)**.** *Suppose standalone and an isolated construction $\mathcal{I}$ begin with identical complete protected state. At every protected step, both select the same protected batch and apply the same protected objective, arithmetic policy, randomness, and control flow. Suppose every optional transition in $\mathcal{I}$ is the identity on protected state and serialization restores the next transition exactly. Then $s_t^{\mathcal{I}} = s_t^0$ for every finite $t \le T$.*

The proof is a direct induction (Appendix A); its use is that each premise is an auditable obligation of the implementation (Table 28). With the same obligations, stopping or replacing a branch preserves later matched boundaries without replaying the protected prefix (Proposition 2). The proposition does not repair earlier joint-training writes and says nothing about branch quality. State completeness and converse results are discussed in Appendices A and T; the long-horizon evidence uses eager FP32, and BF16 and synthetic DDP checks have narrower scope.

3 Observation Design on a Preserved Trajectory

What recovery leaves open. ORA gives the host one writer and the branch its own optimizer, randomness, and data position; read-only copies of host versions may live on other processes or devices. At release T, the branch-disabled and branch-enabled routes share the host parameters θ_T, and future host training retains s_T (Table 5). An *observation policy* specifies which host version is paired with each branch batch, and on which host the branch is deployed. For branch update j,

$$u_{j+1} = T_{\mathrm{br}}\big(u_j; \mathrm{readonly}(s_{\sigma_j}), z_j\big), \tag{4}$$

where σ_j is the observed host step. Recovery fixes the host trajectory; it leaves this version–batch sequence free.

Terminal-only, frozen-base, and latest-version observation. A terminal-only branch sets $\sigma_j = T$ for every update and starts after the host finishes. A frozen-base branch sets $\sigma_j = 0$. Latest-version observation (H) reads the newest published version at each update; because the host can pass several versions between branch updates, or finish while many remain, relative process speed changes H's learning inputs. All three isolate the host; their quality must be compared on a specified deployment host.

Reader-progress observation. P assigns versions by the branch's own update count:

$$\sigma_j = K\lfloor (j-1)/K \rfloor, \qquad K = 250. \tag{5}$$

Updates 1–250 read s_0 and 251–500 read s_{250}, even if newer versions exist. In concurrent execution the branch waits when its assigned version is not yet published; the host never waits. The rule makes the version sequence independent of relative process speed (Figure 1c). It does not bound lag or select a cadence, and its quality requires a separate comparison.

Version–batch pairing. In the main studies the branch reads the host's own training stream, and update j takes the batch the host takes at step j. P then pairs every batch with a checkpoint from before the host reaches it in the current epoch, whereas H, whose host runs ahead, pairs most batches with versions that already have (Figure 2). Serial $\mathrm{ORA}_{\mathrm{pre}}$/$\mathrm{ORA}_{\mathrm{post}}$ make the same contrast exactly, reading the host immediately before/after it updates on the batch; $\mathrm{ORA}_{\mathrm{ckpt}}$ reads periodic checkpoints serially and realizes P's schedule up to a publication-boundary offset (Appendix T.6). Independent branch orders, host-disjoint branch data, and alternative deployment hosts are separate interventions on this pairing (Sections 4.4 to 4.6). The full-model test executes P serially to realize the same schedule without retaining all snapshots (Appendix O).

4 Experiments

On a fixed host trajectory we ask whether the observed versions change branch quality on the same final host (Section 4.2), whether the gain repeats and at what cost (Section 4.3), what it depends on (Sections 4.4 to 4.6), and where it stops (Section 4.7).

4.1 Setting, inference, and host checks

Setting. Quality studies use Qwen2.5-0.5B-Instruct (Qwen Team, 2025) on GSM8K (Cobbe et al., 2021) and SQuAD (Rajpurkar et al., 2016) with group-disjoint fit/tune/holdout partitions. In the one-block setting the host's trainable state is the final decoder block and normalization, and the branch is rank-8 LoRA (Hu et al., 2022) on the preceding query/value projections; disabling LoRA gives the fallback route. Prompts use a plain `Question:`/`Answer:` format without a chat template, so the untrained model scores low ($3.21/37.42$) and all policies share one greedy evaluator (Appendix E). *Two-stage* training implements terminal-only observation. *Clean plus joint* (C) trains a second, joint host beside the clean one. The full-model test trains all non-LoRA parameters under a fixed full-model recipe.

Inference and host checks. Confirmatory studies are preregistered with locked rules (Table 10); tune data are disjoint from evaluation. P itself was specified after a post-outcome exploration on six retained trajectories (Appendix K.7); every study reported below uses new seeds or a locked replay, and all reuse the existing partitions (Appendix L). Superiority inference uses paired 95% intervals and exact one-sided sign-flip tests with Holm correction across tasks; the post-E5 order/data extension uses two-sided tests with Holm across four contrasts; noninferiority to C uses a $1\,pp$ margin. Each study checks the host against its standalone reference: each prospective study passes 20 protected-state equality gates, and complete recorded state matches at $40{,}110$ Qwen0.5B and $10{,}095$ Qwen1.5B boundaries (Appendices C and T). These checks establish the fixed-host premise, not branch quality.

4.2 Observed versions change branch quality on the same final host

A version-only intervention. P's version sequence improves quality over training every branch update on the deployment host (Table 1). The preregistered follow-up reuses ten live host trajectories per task and retains P's branch initialization, recipe, batches, order, random stream, update budget, and deployment host; only the observed versions change, as the terminal-only arm always reads s_T. P leads by $4.67/13.87\,pp$ on GSM8K/SQuAD, with all 20 pairs positive (Holm $p = 0.002$): at fixed batches and order, observing the trajectory accounts for the gap. Against the terminal-only recipe's separately selected learning rate and decay schedule, P still leads by $4.31/7.75\,pp$ (Appendix P).

Table 1: **Matched comparisons on the same clean final host.** P minus comparator (pp), paired 95% intervals, ten retained live trajectories/task. Terminal-only arms share P's initialization and batch order. Bold: same-recipe version-only contrast; tuned: selected learning rate and decay. Both terminal-only contrasts pass on both tasks (Holm $p = 0.002$). Transfer passes only on SQuAD ($p = 0.002$; GSM8K $p = 0.937$).

Comparator	GSM8K accuracy	SQuAD F1
Terminal-only, same recipe	$\mathbf{+4.67}$ $\mathbf{[+3.50, +5.85]}$	$\mathbf{+13.87}$ $\mathbf{[+12.99, +14.74]}$
Terminal-only, tuned recipe	$+4.31$ $[+2.91, +5.71]$	$+7.75$ $[+7.24, +8.26]$
C branch on clean s_T	-0.92 $[-2.21, +0.36]$	$+7.96$ $[+4.85, +11.07]$

The gain survives terminal recipe and checkpoint selection. A locked follow-up gives terminal-only four learning-rate recipes and five stopping checkpoints per seed, selected on half A of the historical holdout and evaluated on group-disjoint half B (Appendix Q). Full-budget P remains ahead by $+5.25\,pp$ on GSM8K ($[+3.94, +6.57]$) and $+6.06\,pp$ on SQuAD ($[+5.37, +6.75]$); all 20 pairs are positive and both tasks pass the prespecified rule (Holm $p = 0.001953$). Selecting P from five checkpoints gives secondary differences of $+5.31/+6.79\,pp$. Terminal selection improves on its original tuned endpoint by $0.21/2.04\,pp$, although on GSM8K it falls below the fixed multiplier-16 endpoint on B. The search therefore addresses optimization and stopping sensitivity without establishing a globally optimal terminal comparator. A/B separation protects this selection rule; it is not fresh-data confirmation.

Which alternative applies depends on the requirement. Table 65 places every policy on one scale. If host continuation is optional, the frozen-base branch deployed on s_0 has the highest enabled-route quality among the tested policies; moving it to the continued host loses $12.4/16.2\,pp$. C keeps a clean host but trains its branch on a second, joint host. Transferred to clean s_T, that branch loses $8.2\,pp$ on SQuAD and $1.5\,pp$ on GSM8K, so P leads on SQuAD, while on GSM8K P's advantage is not established and the case against C rests on its extra host and state provenance alone. Among policies trained against the clean continuing host, the four trajectory observers lie within $1.3/1.8\,pp$ of one another and above the terminal-only and latest-version policies in that full-holdout comparison.

4.3 The gain repeats with small measured host overhead

Two prospective studies against latest-version observation. A branch update costs more than a host update, so the branch process needs about $2.2\times$ the host's training time (Table 39). The live H branch therefore falls behind and observes s_T for 52–64%/53–63% of its updates (Figure 2); P never does during training. This concentration on s_T is consistent with the terminal-only comparison above; its contribution to the $P - H$ gap is not separately identified. In both prospective studies all 40 paired differences favor P, and both tasks pass in each study (Holm $p = 0.00195$; Figure 3a; Table 42). The live study gives each process its own GPU; the shared study places both on one device. Each completes 80 workflows, 20 host equality gates, and 100 evaluations before analysis, with shared initialization, branch order, budgets, retention, and deployment host (Appendix L).

Table 2: **Reader-progress observation improves quality in both confirmations.** Terminal accuracy/F1 (0–100), 10 fresh paired seeds per row. H: latest version; P: reader progress; C: clean plus joint. Differences are in pp; paired intervals and noninferiority boundaries appear in Figure 3.

Study	Task	H	P	C	$P - H$	$P - C$
Live	GSM8K	39.12	43.11	45.53	$+3.99$	-2.42
Live	SQuAD	67.30	77.88	78.14	$+10.57$	-0.26
Shared	GSM8K	38.81	43.21	45.52	$+4.40$	-2.30
Shared	SQuAD	67.84	77.29	78.04	$+9.45$	-0.75

Cost and availability. P leaves the host at $1.01/1.02\times$ standalone training time, whereas serial ORA_{post} places branch updates on the host's critical path at $3.18\times$ (Tables 39 and 45). The cost falls on the branch process: relative to C (also two processes), live P uses $1.247/1.214\times$ the summed process occupancy, which counts waiting, and its median branch lag after host completion

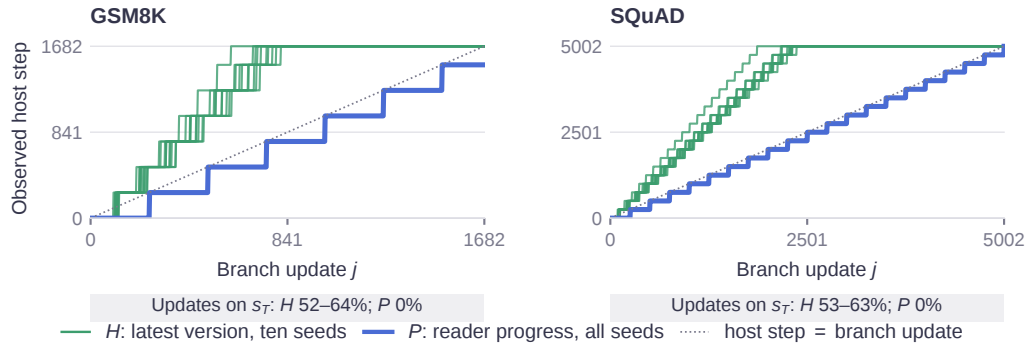

Figure 2: **Realized observation sequences differ under the two policies.** Host step read at each branch update in the live confirmation, ten seeds per task. Thin green lines: H, one per seed; blue: P, identical across seeds. Below the dotted diagonal, the branch reads a version from before the host reaches that update's batch. Quality comparisons follow in Figure 3.

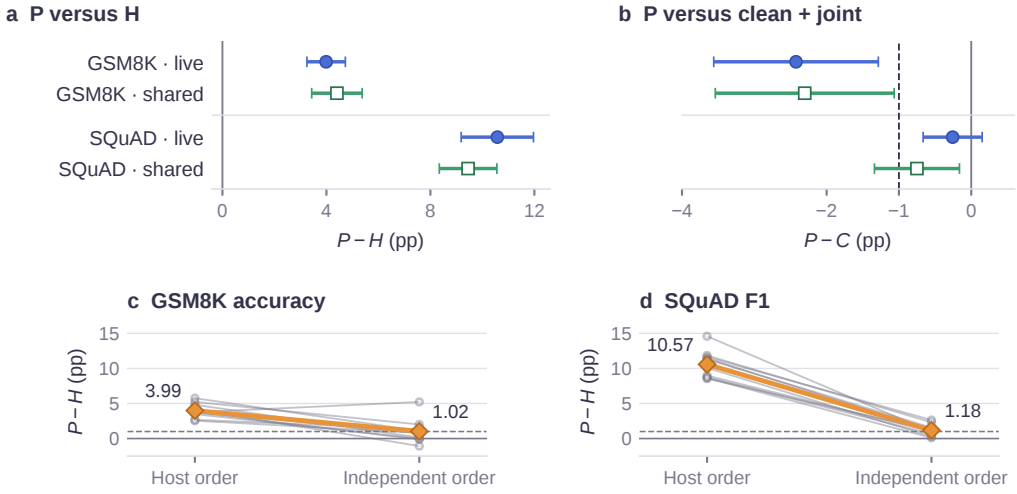

Figure 3: **Prospective observation gains and their batch-order dependence.** (a,b) Paired means and 95% t intervals, 10 fresh seeds/task; aligned rows. C: clean plus joint; dashed line: $-1\,pp$ margin. Only live SQuAD passes noninferiority (Table 42). (c,d) Preregistered replay of the same ten live seeds with one independent branch order per seed (thin lines), means (diamonds); dashed line: $1\,pp$ mean threshold; the complete residual-superiority rule is in Section 4.4. GSM8K: accuracy; SQuAD: F1; differences in pp.

is $1.252/1.211$ times the host's full training duration. The branch is nevertheless usable at release: when the host finishes, P's partially trained branch already scores $43.45/77.67$, against $43.11/77.88$ fully trained, whereas terminal-only training has not begun. P retains every published version (0.48 GB on GSM8K). These execution measurements are descriptive (Appendix L.1).

Clean plus joint bounds the quality claim. Neither study passes joint noninferiority to C: only live SQuAD passes the per-task $1\,pp$ rule; GSM8K trails by $2.42/2.30\,pp$ (Figure 3b).

4.4 The gain depends on version–batch pairing

Independent branch orders. The $P - H$ gap depends on batch order. On all ten live seeds per task we replay both branches offline from retained versions, keeping H's recorded and P's prescribed version sequences, initialization, random streams, recipe, budgets, and final s_T, and change only the branch order to one prespecified permutation per seed; original-order replays on

one seed per task first reproduce the live branches bitwise. Independent orders reduce the gap: the interaction $(P - H) - (P_{\text{own}} - H_{\text{own}})$ is $+2.97/9.39\,pp$ ($9/10$ and $10/10$ pairs shrink, Holm $p = 0.0029/0.0020$). Residual superiority requires a mean above $1\,pp$, a paired 95% interval above zero, and Holm $p < 0.05$. P retains a $+1.18\,pp$ $[+0.61, +1.76]$ advantage on SQuAD ($10/10$ positive), whereas the residual of $+1.02\,pp$ on GSM8K is not established, as its interval $[-0.21, +2.24]$ crosses zero (Figure 3c,d; Appendix N). Shuffling changes order, batch composition, and pairing together, so it establishes batch dependence without identifying a mediator.

Before or after the host fits the batch. Serial observation makes the pairing explicit. On the same batch, reading the host just before its update (ORA_{pre}) beats reading it just after (ORA_{post}) by $1.45/1.69\,pp$ (20 seeds per task; Table 30) and by $0.94/1.82\,pp$ on the ten seeds of the decorrelation control. With independent branch batches the ordering disappears: both pre–post intervals meet the $\pm 1\,pp$ equivalence rule and both members lose quality (Table 32); the locked interaction is $-1.85\,pp$ on SQuAD (Holm $p = 0.0039$) and unresolved on GSM8K ($-1.15\,pp$, $p = 0.0586$; Table 33).

4.5 The advantage reverses on host-disjoint branch data

A possible explanation is that a branch fitted on examples already fit by its host sees residuals that understate held-out error, as in stacked generalization (Wolpert, 1992); intermediate models can also be better teachers than the final model (Wang et al., 2022). We test whether the benefit survives removing host–branch data overlap. A preregistered eight-seed study splits the training pool into group-disjoint, equally sized halves A/B. The host trains on A; the branch trains either on B in an independent order or on A in host order. Within each condition, P and terminal-only share everything except observed versions. Across conditions, host endpoints match bitwise, and branch sample counts and budgets match (Appendix R).

When the branch trains on data disjoint from the host's, P's advantage over terminal-only observation is not established on GSM8K ($-1.86\,pp$ $[-3.13, -0.58]$) or SQuAD ($-1.16\,pp$ $[-1.54, -0.78]$; Table 3): the one-block gain is established only for branches trained on the host's own training stream. Sharing the host's training data changes the $P-$terminal difference by $+5.94/+14.67\,pp$ (intervals $[+4.37, +7.51]/[+13.38, +15.95]$; descriptive).

Table 3: **Host-disjoint branch data.** P minus terminal-only (pp), paired 95% intervals, eight new seeds per task. The host trains on one half of the training pool; the branch trains on the group-disjoint other half (disjoint) or on the host half in host order (shared). Both arms share initialization, batches, order, random stream, recipe, budget, and final host. Shared rows are descriptive.

Branch data	GSM8K accuracy	SQuAD F1
Host-disjoint	-1.86 $[-3.13, -0.58]$	-1.16 $[-1.54, -0.78]$
Shared with host	$+4.08$ $[+2.99, +5.18]$	$+13.51$ $[+12.31, +14.71]$
Shared minus disjoint	$+5.94$ $[+4.37, +7.51]$	$+14.67$ $[+13.38, +15.95]$

The shared-data control retains a $+4.08/+13.51\,pp$ advantage at the same reduced budget. A locked post-E5 extension adds shared data in independent branch order on the same seeds (Appendix S). The difference becomes $+0.64\,pp$ on GSM8K ($[-1.02, +2.29]$) and $+8.48\,pp$ on SQuAD ($[+6.18, +10.77]$). Reordering reduces the gap by $3.45/5.03\,pp$; replacing shared with disjoint data under independent order reduces it by $2.49/9.63\,pp$ (all four two-sided contrasts: Holm $p = 0.03125$). Order alone therefore does not explain the SQuAD advantage. These conditional contrasts do not identify a unique mechanism; a zero-containing interval does not establish equivalence.

4.6 The deployment host changes the comparison

A branch's quality also depends on the host it is deployed on. Holding frozen-base and periodic (K=250) branch weights fixed while crossing deployment hosts s_0/s_T (Figure 7; Table 34), matching the frozen-base branch to its training host s_0 raises quality by $12.44/16.18\,pp$; the policy–host interaction is $52.64/73.88\,pp$, and all six locked contrasts pass (Holm $p = 0.0117$; Table 35). Changing the deployment host reverses the ranking of the two policies.

4.7 WHERE THE ADVANTAGE STOPS

Full-model training. A prospective test asks whether P's one-block advantage extends to a fixed full-model recipe. It does not: P trails terminal-only by $7.93/4.74\,pp$, with all 16 paired differences negative (Table 4). Eight new seeds per task keep Qwen0.5B and the partitions; within each pair everything but the observed versions matches, with host/branch rates $2 \times 10^{-5}/2 \times 10^{-4}$. An earlier full-model study agrees: the one-block advantage of ORA_{post} over two-stage did not replicate under full-model protection ($-5.96/-4.15\,pp$; Table 31). Recipe changes across settings prevent attributing the reversal to the protected fraction alone (Appendix O).

Table 4: **Matched observation comparison in full-model training.** P minus terminal-only, in pp with paired 95% intervals; eight new seeds per task. Both arms share initialization, batches, order, random stream, recipe, budget, and final host. All 16 differences favor terminal-only.

Comparison	GSM8K accuracy	SQuAD F1
P minus terminal-only	$-7.93\ [-9.72, -6.15]$	$-4.74\ [-5.69, -3.79]$

Supporting comparisons. In the one-block setting, ORA_{pre}, ORA_{post}, and periodic observation each beat selected two-stage training by 3.20–4.65 pp on GSM8K and 5.09–6.77 pp on SQuAD (Table 31), and ORA_{post} checkpoints gain $2.01/10.74\,pp$ on the standard GSM8K test/SQuAD validation splits ($n = 20$; Table 27); GSM8K scores are lower on the official test split than on our holdout, which is drawn from the training split (31.83 versus 43.04 for the same checkpoints). Exact recovery can also cost quality: on BERT QNLI, patience-2 accuracy is 75.64 with exact isolation versus 83.58 with coupled updates (Table 21).

5 RELATED WORK

Selectable computation and added modules. Early exits and adaptive computation choose routes through shared features (Lee et al., 2015; Graves, 2016; Bolukbasi et al., 2017; Huang et al., 2018; Schuster et al., 2022; Teerapittayanon et al., 2016; Zhou et al., 2020; Elhoushi et al., 2024); slimmable and weight-sharing networks choose subnetworks (Yu et al., 2019; Yu & Huang, 2019; Cai et al., 2020; Yu et al., 2020a; Pham et al., 2018); adapters, side networks, and expert branching add state to inherited models (Houlsby et al., 2019; Zhang et al., 2020; Rebuffi et al., 2017; Pfeiffer et al., 2021; Rusu et al., 2016; Sung et al., 2022; Li et al., 2022; Fedus et al., 2022; Komatsuzaki et al., 2023). These methods specify forward routes, not the training process that remains when a module is removed. DeeBERT and Side-Tuning keep removal clean by freezing the backbone (Xin et al., 2020; Zhang et al., 2020); we keep it clean while the host continues training.

Coupled learning, isolation, and counterfactuals. Gradient methods reduce interference between coupled updates (Shipard et al., 2022; Chen et al., 2018; Yu et al., 2020b; Sener & Koltun, 2018) but still write to shared state; ORA instead requires the host to equal standalone throughout training (Appendix D). Noninterference and disjoint-state reasoning provide the isolation foundation (Goguen & Meseguer, 1982; O'Hearn, 2007), and EasyScale tracks implicit state for bitwise elastic training (Li et al., 2023); we apply this foundation to optional learning and test predicted first divergence. Unlearning certifies or reconstructs removal counterfactuals after training (Guo et al., 2020; Abdullah X, 2025; Bourtoule et al., 2021), whereas ORA preserves the protected prefix as it is computed. Following execution-audit precedents (Thudi et al., 2022; Jia et al., 2021; Zhuang et al., 2022; He & Thinking Machines Lab, 2025), we separate state equality from authentication of historical data use.

Versioned learning inputs. Stale Synchronous Parallel bounds read staleness during shared-parameter updates (Ho et al., 2013); PipeDream preserves versions across forward/backward computation (Narayanan et al., 2019); MoCo stabilizes representations through a momentum encoder (He et al., 2020). Adapter portability across base-model versions is handled after the host has changed: Trans-LoRA regenerates adapters from synthetic data, PortLLM adds an old patch to the evolved base, and Trans-PEFT trains adapters to survive base updates (Wang et al., 2024; Khan et al., 2025; Gu et al., 2025). Our question arises while the host is still changing: which of its versions should accompany a removable branch's batches? We compare this choice at a common final host and test how batch pairing and training setting bound its effect (Appendix D.2).

Learning from another learner's outputs. Stacked generalization uses out-of-sample predictions to train a second learner (Wolpert, 1992), and checkpoint distillation uses intermediate teacher models (Wang et al., 2022). These precedents motivate checking whether a terminal host's fitted training examples give the branch a different learning problem from held-out examples. Our data-separation intervention tests that dependency for a branch deployed on a common final host; it does not establish that progress-indexed observation implements cross-fitting or isolate residual bias as the cause.

6 Discussion and Conclusion

Choosing an observation policy. The requirement decides the comparison. If the host need not continue, freeze it and train the branch on s_0; in our setting this gives the strongest branch (Table 65). If the host must keep training and the branch must ship on the released host, the tested one-block shared-stream evidence favors trajectory observation over terminal-only and latest-version observation: progress-indexed observation, concurrently as P or serially as $\mathrm{ORA}_{\mathrm{pre}}$ and $\mathrm{ORA}_{\mathrm{ckpt}}$, gave the best branches on s_T, with measured host completion times of $1.01/1.02\times$ standalone for live P. If a second, joint host is affordable, transferring its branch is a competitive alternative on GSM8K. On host-disjoint branch data and in full-model training we found the opposite ranking, so the policy must be evaluated with the intended data and training recipe.

What the controls establish together. At a common final host, the version-only comparison isolates the observed version sequence from initialization, recipe, batch order, randomness, and budget (Table 1). The expanded terminal search retains the shared-stream advantage after recipe and stopping-point selection (Section 4.2). The batch interventions show that the advantage depends on which batches accompany which versions (Section 4.4); the equal-size data study reverses the advantage, and its shared-independent extension separates order and data contrasts while retaining a SQuAD gap (Section 4.5); and the fixed-branch control separates learning from the deployment host (Figure 7). Together, they identify the version sequence, its pairing with batches, and the deployment host as consequential choices under a preserved trajectory, without isolating a unique mechanism.

Limitations. Quality evidence is concentrated on Qwen2.5-0.5B, one protected block, and two tasks. Freezing gives higher branch-enabled quality in the main setting ($50.13/80.59$ versus $43.11/77.88$ for P), while continuing improves the branch-disabled host ($3.21/37.42$ to $37.03/55.50$). This establishes a measured trade-off between enabled and fallback routes; demand for independent host continuation in deployment remains unmeasured. Terminal-only training is stronger in the tested full-model recipe; because the host learning rate and configuration changed with the protected fraction, the cause is unresolved. The expanded terminal grid reaches multiplier 32, selected by one GSM8K seed; further tuning and warm-started terminal variants remain untested. Selection and estimation halves reuse historical holdout data. P was specified after a post-outcome exploration and confirmed with fresh seeds on the same partitions. Independent-batch controls retain a SQuAD gap, but order changes include batch composition. The post-E5 extension reuses seeds and data and is not a full overlap-by-order factorial. Neither P/H study establishes noninferiority to clean plus joint. Timing is descriptive and measured on shared hardware; recovery tests on other stacks and scales do not establish quality transfer or an automatic policy selector.

Conclusion. Under the stated isolation conditions, abandoning a branch preserves the host's matched training state and continuation. The quality of the branch depends on a separate choice: which host versions accompany its training examples. Earlier versions improve final-host branch quality in the tested one-block shared-stream setting, but the ranking reverses with host-disjoint branch data and in the tested full-model recipe. Observation policy, branch data, and deployment host must therefore be evaluated together; exact recovery alone selects none of them.

AI USE STATEMENT

AI assistance supported literature queries and related-work comparisons; feedback on experimental design and interpretation; implementation of training, evaluation, analysis, and verification scripts; figure preparation; and manuscript drafting and revision. Authors checked retained sources, citations, and scientific claims, reviewed AI-assisted work, and take responsibility for the final content. Reported measurements come from executed experiments and recorded artifacts; AI-generated assessments are not evidence. The correction separating the decay pilot from its independent extension and the post-outcome role of the observation-sequence intervention are documented in Appendices E.1, K.7 and L.1.

ETHICS STATEMENT

We use publicly released models and QNLI, SST-2, GSM8K, and SQuAD data, with no human-subject study or new data collection. The provenance study introduces canary secrets and evaluation examples into the optional branch's stream to measure retained information and protected-state isolation (Appendix E.6). These are controlled auditing tests; exactness does not authenticate historical data use or establish a general privacy guarantee. The exact-fallback evidence-reuse study (Appendix E.18) does not license skipping deployment validation; its preregistered operational gate failed.

REPRODUCIBILITY STATEMENT

Recipes, data splits, environment records, protocol locks, and execution amendments are in Appendices E and E.1; proofs, state inventories, and statistical rules are in Appendices A to C and I. Separate tune data support selection. The observation-sequence study that first specified P is a descriptive post-outcome offline intervention on six retained trajectories (Appendix K.7). The quality claims rest on locked studies with new seeds or locked replays on the existing partitions: two prospective P/H confirmations (Appendices L and L.1), the matched terminal-only and transfer follow-up on the ten live trajectories per task (Appendix P), the independent-order replay (Appendix N), the full-model test (Appendix O), and the host-disjoint study (Appendix R); the earlier three-seed batch control is in Appendix M. The anonymized supplement contains code, analyses, and the recorded inputs for manuscript verification. Its CPU entry point checks shipped files, reconstructs statistics, and verifies source bindings; omitted model/dataset tensors and the two prospective confirmations' prediction dumps are enumerated with digests, and the portable check recomputes their quality statistics from bound score receipts without regenerating predictions. Receipt verification inspects recorded equality checks, while rerunning tensor comparisons requires those artifacts and the pinned environment.

REFERENCES

Abdullah X. Unlearning at scale: State-exact trace-preserving deletion in billion-parameter language models, 2025. arXiv:2508.12220v2, updated 12 August 2026.

Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *IEEE Symposium on Security and Privacy*, 2021.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. LayerSkip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 12622–12642, 2024. doi: 10.18653/v1/2024.acl-long.681.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Joseph A. Goguen and José Meseguer. Security policies and security models. In *IEEE Symposium on Security and Privacy*, pp. 11–20, 1982. doi: 10.1109/SP.1982.10014.

Alex Graves. Adaptive computation time for recurrent neural networks, 2016.

Naibin Gu, Peng Fu, Xiyu Liu, Ke Ma, Zheng Lin, and Weiping Wang. Adapt once, thrive with updates: Transferable parameter-efficient fine-tuning on evolving base models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14765–14783, 2025.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Horace He and Thinking Machines Lab. Defeating nondeterminism in LLM inference. Thinking Machines Lab: Connectionism, `https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/`, 2025.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Qirong Ho, James Cipar, Henggang Cui, Jin Kyu Kim, Seunghak Lee, Phillip B. Gibbons, Garth A. Gibson, Gregory R. Ganger, and Eric P. Xing. More effective distributed ML via a stale synchronous parallel parameter server. In *Advances in Neural Information Processing Systems*, 2013.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification. In *International Conference on Learning Representations*, 2018.

Hengrui Jia, Mohammad Yaghini, Christopher A. Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning: Definitions and practice. In *IEEE Symposium on Security and Privacy*, pp. 1039–1056, 2021. doi: 10.1109/SP40001.2021.00106.

Rana Muhammad Shahroz Khan, Pingzhi Li, Sukwon Yun, Zhenyu Wang, Shahriar Nirjon, Chau-Wai Wong, and Tianlong Chen. PortLLM: Personalizing evolving large language models with training-free and portable model patches. In *International Conference on Learning Representations*, 2025.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *International Conference on Learning Representations*, 2023.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models, 2022.

Mingzhen Li, Wencong Xiao, Hailong Yang, Biao Sun, Hanyu Zhao, Shiru Ren, Zhongzhi Luan, Xianyan Jia, Yi Liu, Yong Li, Wei Lin, and Depei Qian. EasyScale: Elastic training with consistent accuracy and improved utilization on GPUs. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2023. doi: 10.1145/3581784.3607054.

Deepak Narayanan, Aaron Harlap, Amar Phanishayee, Vivek Seshadri, Nikhil R. Devanur, Gregory R. Ganger, Phillip B. Gibbons, and Matei Zaharia. PipeDream: Generalized pipeline parallelism for DNN training. In *Proceedings of the 27th ACM Symposium on Operating Systems Principles*, pp. 1–15, 2019. doi: 10.1145/3341301.3359646.

Peter W. O'Hearn. Resources, concurrency, and local reasoning. *Theoretical Computer Science*, 375 (1–3):271–307, 2007. doi: 10.1016/j.tcs.2006.12.035.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, 2021.

Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, 2017.

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

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, 2022.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, 2018.

Jordan Shipard, Arnold Wiliem, and Clinton Fookes. Does interference exist when training a once-for-all network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. LST: Ladder side-tuning for parameter and memory efficient transfer learning. In *Advances in Neural Information Processing Systems*, 2022.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *International Conference on Pattern Recognition*, 2016. doi: 10.1109/ICPR.2016.7900006.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *USENIX Security Symposium*, pp. 4007–4022, 2022.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Chaofei Wang, Qisen Yang, Rui Huang, Shiji Song, and Gao Huang. Efficient knowledge distillation from model checkpoints. In *Advances in Neural Information Processing Systems*, 2022.

Runqian Wang, Soumya Ghosh, David Cox, Diego Antognini, Aude Oliva, Rogerio Feris, and Leonid Karlinsky. Trans-LoRA: Towards data-free transferable parameter efficient finetuning. In *Advances in Neural Information Processing Systems*, 2024.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2246–2251, 2020. doi: 10.18653/v1/2020.acl-main.204.

Jiahui Yu and Thomas S. Huang. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. In *International Conference on Learning Representations*, 2019.

Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc V. Le. BigNAS: Scaling up neural architecture search with big single-stage models. In *European Conference on Computer Vision*, 2020a.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, 2020b.

Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: A baseline for network adaptation via additive side networks. In *European Conference on Computer Vision*, pp. 698–714, 2020. doi: 10.1007/978-3-030-58580-8_41.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems*, 2020.

Donglin Zhuang, Xingyao Zhang, Shuaiwen Leon Song, and Sara Hooker. Randomness in neural network training: Characterizing the impact of tooling. In *Proceedings of Machine Learning and Systems*, 2022.

A Proof of Conditional Recoverability

Proof of Proposition 1. Induct on protected steps. Equality holds at $t = 0$ by assumption. Assume $s_t^{\mathcal{I}} = s_t^0$. Because the two processes select the same protected batch and apply the same protected objective, arithmetic, randomness, and control policy, their protected transition maps receive identical inputs. The construction therefore produces $T_0(s_t^0; x_t) = s_{t+1}^0$ at its next protected boundary. Every optional transition changes only branch-owned state, whether it executes before or after the protected transition or reads a published copy of an earlier boundary, so the protected boundary remains $s_{t+1}^C = s_{t+1}^0$. Induction establishes equality for every finite $t \le T$; exact serialization preserves the premise across interruption and resume. Equation (6), Equations (7) and (8), and Equation (9) are the pre-transition, post-transition, and checkpoint-assisted instances. □

The proposition establishes sufficiency, not a converse for every realized run. Violating an assumption invalidates the induction but need not change every endpoint or task metric.

Proposition 2 (Prefix-preserving branch abandonment)**.** *Under Proposition 1, if the branch is stopped or replaced at boundary $\tau \le T$ and subsequent optional work remains identity on protected state, then every later matched protected boundary equals standalone. No replay of the protected prefix is required.*

Proof of Proposition 2. At boundary τ, Proposition 1 gives $s_\tau^{\mathcal{I}} = s_\tau^0$. After stopping or replacing the branch, the ownership condition keeps every subsequent branch action identity on protected state. The induction step from Proposition 1 therefore applies for $t = \tau, \dots, T-1$, yielding $s_t^{\mathcal{I}} = s_t^0$ at every later boundary. The prefix needs no replay because it already satisfies the standalone invariant. □

B Why the Contract Has This Form

Proposition 1 proves that ownership separation suffices. Two questions remain: whether exactness could instead arise by cancelling optional influence, and whether any component of Equation (1) can be omitted. The following results answer both.

B.1 Exactness requires non-influence, not cancellation

Lemma 3 (Uniform exactness removes optional dependence)**.** *Let an optional procedure be recoverable at step t for every admissible pair (u_t, z_t) in its reachable set. Then $\pi_s T_{\mathrm{opt}}(s_t, u_t; x_t, z_t)$ is constant in (u_t, z_t) and equals $T_0(s_t; x_t)$.*

Proof. By Equation (3), $\pi_s T_{\mathrm{opt}}(s_t, u_t; x_t, z_t) = T_0(s_t; x_t)$ for every admissible (u_t, z_t). The right-hand side does not depend on (u_t, z_t), so neither does the left-hand side. □

The elementary lemma exposes the key distinction between cancellation and non-influence. Cancellation can hold on a realized trajectory, whereas uniform exactness must hold over every admissible optional state. A uniformly exact construction must therefore make its protected projection independent of optional variables. Gradient surgery illustrates the difference: it reduces realized interference without removing dependence, and Table 22 records the residual—a near-zero task-level gap on one task, yet $0/20$ endpoint equality and divergence by protected step three.

Sensitivity of the protected update turns this observation into a necessity result. Let v_t denote the protected update input (for example, a gradient after clipping), and write $T_0(s_t; v_t)$ for the resulting transition. For fixed s_t, suppose $T_0(s_t; v)$ is injective on two admissible values $v \neq v'$. A coupled procedure that changes v_t across admissible branch states cannot then be uniformly exact: the same protected output would have to equal both distinct values $T_0(s_t; v)$ and $T_0(s_t; v')$. This optimizer-sensitive impossibility rules out cancellation whenever the transition distinguishes the changed input. It remains conditional and does not imply that every realized coupling changes every endpoint.

Proposition 4 (Coupled-update necessity under sensitivity)**.** *Fix s_t and x_t. Suppose the standalone protected transition factors as $T_0(s_t; v_t)$, and there are two admissible optional states for which a coupled procedure supplies $v_t \neq v_t'$. If $T_0(s_t; \cdot)$ is injective on these two values, then the coupled procedure cannot satisfy the uniform recovery condition for both optional states. Thus two protected*

inputs distinguished by the transition cannot both be supplied while preserving the same standalone output for every optional state.

Proof. Uniform recovery would require $T_0(s_t; v_t)$ and $T_0(s_t; v'_t)$ both to equal the fixed matched standalone output. Injectivity makes those two outputs distinct, a contradiction. □

Remark 5 (Exactness does not determine the observation boundary). A branch that reads s_t and writes only branch-owned state satisfies the same projection identity as a branch that reads the post-transition state or a published checkpoint. The conditional guarantee therefore covers each policy. Their optional updates receive different views and may interact differently with the selected training batches; recoverability alone ranks neither their task quality nor their execution cost. A comparison of observation policies must specify the update phase, publication rule, batch selection, and the protected state used at evaluation.

B.2 Continuation, overlap, and the designated base

Three requirements must be distinguished: preserving the clean training continuation, overlapping branch learning with that run, and attaching the branch to a designated clean release. Clean plus joint satisfies the first with a separate clean process; transferring its branch to the clean endpoint also satisfies the final-host requirement, with task-dependent quality (Table 1). A branch observing a frozen copy can train while the clean writer continues; its observation remains fixed. Terminal two-stage learning waits for the clean endpoint. Isolated checkpoint readers observe versions of the clean trajectory under the ownership contract. Table 5 records these choices without ranking quality or cost. Matrix VII tests frozen-base training and evaluation compatibility (Appendix K).

B.3 No component of the contract is redundant

Proposition 6 (Contract minimality)**.** *For each component of Equation* (1)*, there is an execution model and an admissible optional procedure that changes only that component at a protected boundary, and a subsequent protected transition that distinguishes the two states. An audit omitting that component can therefore accept a non-recoverable boundary. Exact serialization must preserve the same information across restoration. This is an inventory obligation across execution models.*

Proof. Fix identical states at a boundary and let the optional operation perform one deterministic assignment to the named component, leaving all other components unchanged. The following assignments and consuming transitions give witnesses. *Parameters θ:* add one to a scalar parameter before the protected map $\theta \mapsto \theta/2$. *Buffers b:* change only a stored normalization mean before a protected forward pass that uses it. *Optimizer state o:* change only a momentum accumulator m before the update $\theta^+ = \theta - \eta(\beta m + g)$, with $\eta\beta \neq 0$; no optimizer step is executed by the optional operation. *Scheduler state q:* change only a schedule counter before the optimizer reads a counter-dependent learning rate and applies a nonzero gradient. *Precision and scaler state a:* change only a pending overflow flag before a precision controller decides whether to apply a nonzero protected update. *Randomness r:* advance only a generator whose next two draws differ before a protected operation consumes the next draw. *Data position d:* increment only a cursor before selection of two batches with different protected gradients. *Control state c:* change only an accumulation counter before a controller chooses to apply or defer a nonzero update. At the audited boundary the retained components are identical, yet the omitted component differs; at the consuming transition the difference affects protected computation. Each assignment is compatible with a removable forward route. For serialization, restoring one such component to the altered value while restoring the others exactly gives the same counterexample across a checkpoint. □

The inventory covers execution models in which the listed fields are independently mutable and transition-relevant; a particular stack may admit a coarser sufficient statistic or a vacuous field. The proof's single-component assignments establish this obligation. The controlled suites test practical coupling operations: these can propagate within one transition, as global clipping and optimizer reuse change both model and optimizer in Table 25. Those multi-component interventions diagnose leakage; the single-component witnesses establish why omission can miss it.

Table 5: **Clean continuation and branch observation are separate design choices.** Entries describe the stated constructions, not measured resource advantages. A continuing clean writer may coexist with a frozen branch base. Correctly isolated live and checkpoint readers both implement ORA.

Alternative	Clean continuation	Branch observation / fusion target
Clean + frozen branch	Clean writer may continue independently	Fixed copy / its matched frozen base
Terminal two-stage	Clean run finishes before branch training	Final clean state / that state
Clean + joint	Separate clean process is retained	Evolving joint host / joint endpoint
C branch transferred	Separate clean process is retained	Evolving joint host / clean endpoint
Checkpoint and replay	Reconstructed from an eligible checkpoint	Protocol-dependent; count replay work
Joint / PCGrad	Generally changes the protected process	Coupled host / coupled endpoint
Isolated live or checkpoint reader	Clean writer continues under the contract	Clean trajectory views / designated clean release

B.4 Composition and the boundary of one-state claims

Disjoint ownership makes the identity construction compositional. This is a correctness result, not a claim of faster multi-branch training or a new storage advantage. Let $u^{(1)}, \dots, u^{(K)}$ be branch states, with C_j updating only $u^{(j)}$ while reading a common protected snapshot. If each C_j is identity on s, their sequential composition is also identity on s.

Proposition 7 (Disjoint-branch composition)**.** *Suppose K branch transitions have pairwise disjoint writable state, all read the same protected state $\bar{s}_{t+1}$, and satisfy $\pi_s C_j(s, u^{(j)}; \bar{s}_{t+1}, z_j) = \bar{s}_{t+1}$ for every admissible input. Then any composition $C_K \circ \cdots \circ C_1$ preserves $\bar{s}_{t+1}$ exactly, while each branch state changes only in its own ownership region.*

Proof. Apply the premise to C_1: its protected projection is $\bar{s}_{t+1}$. Because its writable state is disjoint from every other branch region, the inputs and protected read-only snapshot seen by C_2 are unchanged, so the premise applies again. Repeating this argument for $j = 1, \dots, K$ leaves the protected projection equal to $\bar{s}_{t+1}$ and changes no branch-owned region other than the one assigned to that transition. □

Independently coupled protected forks fall outside the proposition because each fork has its own protected trajectory, violating the shared-state premise. The result defines a compositional correctness boundary rather than empirical multi-branch utility.

Proposition 8 (Conditional retention lower bound for independent forks)**.** *Consider K independently coupled protected forks with pairwise distinct final protected artifacts $a_1, \dots, a_K$. Suppose deployment must select any one artifact by direct lookup from retained state, without recomputation or changing the exactness target. Then the retained representation must contain at least K distinguishable protected representations; in particular, a single shared protected representation cannot provide all K exact fallbacks.*

Proof. If two artifacts shared the same retained representation, direct lookup would return the same protected state for both selections, contradicting $a_i \neq a_j$. Thus the representation map must be injective on the K artifacts and contain at least K distinguishable values. The conclusion is conditional on direct lookup: recomputation, compression with a separately exact decoder, or a relaxed artifact target changes the model and is outside this lower bound. □

B.5 Lifecycle decisions and the protected artifact

The application target is a declared training artifact and its continuation. For example, an optional attachment may consume a data stream excluded from an ongoing protected run. Removing the attachment should leave the protected run at its clean boundary, with the state needed for its next update. The provenance study tests exclusion of branch-only contamination; the abandonment studies

test continuation after removing the attachment and restoring a checkpoint. These controlled examples establish consequences of the requirement. They do not establish its prevalence, deployment necessity, or a general privacy guarantee.

Table 6: **Choosing a lifecycle design depends on what may change or wait.** Each row states a sufficient reason to consider an alternative. Resource and quality preferences still require measurement.

Requirement or flexibility	Candidate design	What must be retained or measured
Branch observation and fusion may use a fixed base	Frozen-base adaptation	Preserve that base and isolate task-specific state; a separate clean writer may continue.
Attachment training can wait until protected training ends	Terminal two-stage	Retain the terminal model and budget the subsequent branch training.
A second evolving state is acceptable	Clean plus joint	Retain the clean process for fallback; count both executions and compare enabled quality.
Earlier protected computation may be repeated	Checkpoint and replay	Retain the required clean checkpoint and execution information; count replay work.
Protected learning and optional observation must overlap with one protected writer	Isolated live or checkpoint reader	Check protected transitions; measure branch quality, copy storage, total work, and actual readiness separately.
Only enabled quality matters	Coupled training	Evaluate that objective directly; exact recovery is an additional requirement.

This decision structure follows the alternatives already present in the literature. DeeBERT freezes the inherited model before training additional exits, Side-Tuning permits discarding side networks while its base stays fixed, and replay-based deletion reconstructs a declared counterfactual from an eligible checkpoint (Xin et al., 2020; Zhang et al., 2020; Abdullah X, 2025). ORA addresses continuing protected learning with optional observation and fusion on a designated clean release. This is one logical trajectory: read-only copies may span devices and processes. The one-writer construction does not imply one GPU, no model copy, or reduced total computation. A separate clean process is a legitimate competitor whenever its cost is acceptable. Algorithmic provenance must also be assessed separately: even matching parameters or recorded trajectories does not by itself authenticate the historical training data (Thudi et al., 2022).

C Evidence Levels

Table 7: Recoverability evidence hierarchy. Each observation supports only its stated equality level.

Observation	Supports	Does not establish
Path-off graph equality	architectural removability	training-process or artifact equality
Equal metric/predictions	agreement on tested examples	parameter or state equality
Equal model-state endpoint	tested endpoint parameters and buffers	optimizer/RNG/cursor identity or intermediate trace
Equal full-run protected observable	step-by-step agreement of a transition-determined quantity over entire runs	complete-state equality; unobserved channels
Equal finite recorded-state trace	exact equality of recorded components for the tested horizon and stack	complete-state equality, arbitrary-system exactness, or unrecorded-state equality
Conditional proposition + contract tests	theorem/implementation boundary under stated assumptions	satisfaction by an untested implementation

The experiments deliberately occupy different levels of this hierarchy. BERT and the Qwen task matrix provide protected model-state endpoint evidence. Per-step protected loss supplies a full-run observable: a difference witnesses a changed protected computation, but agreement leaves unobserved channels unconstrained. Finite recorded-state evidence comes from the controlled traces, the eight-step Qwen interventions, the uninterrupted two-epoch Qwen2.5-0.5B and GSM8K Qwen2.5-1.5B traces in Appendix T, and the protected-only restore and continuation in Table 17. The lifecycle also

establishes projected token-sequence equality on its complete locked holdout, but only for that pinned stack. Trajectory replay compares the tested implementation's complete recorded endpoint inventory. Equality of the theoretical complete protected transition state, and its finite-horizon sufficiency, remain conditional on Proposition 1 and the stated execution assumptions.

D Positioning and Coverage of Existing Evidence

D.1 An established preservation goal and its extension

Existing removable adaptation methods establish preservation as a useful design objective. DeeBERT freezes its fine-tuned backbone to protect final-exit quality while fitting intermediate exits (Xin et al., 2020); Side-Tuning keeps its base fixed and permits discarding side networks during continuous adaptation (Zhang et al., 2020). Ladder Side-Tuning uses a frozen backbone to reduce memory cost (Sung et al., 2022). ORA extends the object to be preserved from inherited state to its ongoing standalone evolution. The cited precedents motivate preservation, while demand for attachment to a continuing clean release remains unmeasured.

Table 8: **The contribution changes the studied object and the empirical question.** This comparison records explicit designs in the cited works; it is not an exhaustive novelty search or a resource ranking.

Precedent	Established design / target	Question studied here
DeeBERT / Side-Tuning	Fixed inherited backbone or base during branch training	Preserve the matched evolution of a continuing host, including after abandonment and restoration.
Ladder Side-Tuning	Frozen backbone and side-network training to reduce activation storage	Audit continuation under optional learning; total resource savings require separate evidence.
Noninterference / disjoint-state reasoning	Isolation as a correctness principle	Make optional-write obligations executable and test predicted first divergence in training stacks.
EasyScale	Exact execution under resource elasticity, including implicit state	Test optional-training influence, then vary what isolated readers observe from a matched trajectory.

Noninterference supplies the isolation foundation (Goguen & Meseguer, 1982; O'Hearn, 2007), and EasyScale identifies implicit state required for reproducible training (Li et al., 2023). Our contribution makes a continuing-process target executable and uses the preserved trajectory to control an empirical learning question. With the host and final fusion base fixed, version assignment remains free to change branch inputs. Replay, prospective confirmation, and direct batch intervention establish a repeatable effect and its boundary. The recovery proposition supplies the admissible construction; it cannot determine the quality ordering within that construction.

D.2 Version coordination is adjacent, not interchangeable

Several systems coordinate versions because their workers update a shared or pipelined model. Stale Synchronous Parallel bounds the age of worker reads while workers contribute updates to the shared parameter server (Ho et al., 2013); PipeDream stashes model versions so a minibatch's forward and backward passes use a consistent stage-local version (Narayanan et al., 2019); MoCo derives a slowly changing key encoder from the query encoder to maintain a representation dictionary (He et al., 2020). These mechanisms establish that version age and consistency can affect optimization. Their shared or source parameters evolve through those learner updates. ORA's writer follows the standalone clean transition while the optional branch writes separate state and reads designated snapshots. The contribution is the reader-progress assignment and its quality/batch boundary under that preserved trajectory, not the invention of versioned execution.

The consequential distinction is what the experiment holds fixed. The cited systems coordinate versions while training coupled or pipelined parameters. ORA preserves the host's standalone evolution, allowing reader inputs to change without altering that source trajectory. At a common terminal base, the matched terminal-only intervention fixes initialization, recipe, and batch order

while replacing the observed versions (Table 1). The P/H comparisons test two assignments during concurrent execution; the batch intervention tests their dependence on the accompanying samples. Separate fusion controls test attachment to the evaluation base. This combination establishes observation design as a learning choice left open by recovery, without claiming that version clocks or stable inputs are new concepts.

Worker progress and schedule equivalence. A latest-publication reader starting after the writer finishes observes s_T throughout; with the matched initialization, recipe, batch sequence, and random stream, this is the terminal-only input sequence. Equivalence to a checkpoint reader instead requires equality of the version assigned at every update and of those other inputs. Observing every publication, or matching an average worker speed, does not suffice. P uses $K\lfloor(j-1)/K\rfloor$ before reader update j, whereas the serial checkpoint construction in Equation (9) publishes after the protected update. The measured policy means do not establish monotonic quality in worker speed or bounds between endpoint policies. P specifies the version sequence independently of relative worker progress; wall-clock cost and numerical behavior across execution environments require their own evidence.

D.3 Which changes of setting have been tested?

The existing studies test several distinct forms of robustness and transfer. Table 9 keeps their method, changed setting, and supported conclusion separate. In particular, the secondary evaluation on standard splits uses the earlier ORA_{post} and its original two-stage comparator, not the later P/H policy pair or the re-tuned Matrix II comparator. It changes evaluation data without retraining or selecting new checkpoints; no practical margin was prespecified (Appendix G).

Table 9: **Recovery transfer, quality sensitivity, and policy confirmation have different evidence.** Every row summarizes an existing study; no result is transferred from one method or evidence level to another.

Method / setting change	Existing evidence	Supported conclusion and boundary
ORA_{post}: Qwen LoRA to BERT exits	Matched removal and fresh-process continuation (Appendix T.5)	Recovery lifecycle across tested path types; does not establish P quality transfer.
ORA_{post}: larger Qwen model	Complete recorded GSM8K trace at Qwen2.5-1.5B (Appendix T)	Recovery implementation at larger scale; no quality result in that trace study.
ORA_{post}: standard evaluation splits	Positive differences from original two-stage on GSM8K test and SQuAD validation (Table 27)	Evaluation-split sensitivity of existing checkpoints; no new-model training transfer.
P/H: fresh seeds and two executions	Both primary task tests pass separately in live and shared studies (Table 2)	Internal confirmation on the existing recipe; execution studies do not identify a causal load effect.
P/H: independent branch order	Preregistered replay on ten live seeds/task (Appendix N); earlier six-trajectory exploration retained	Batch dependence confirmed on both tasks; residual superiority established on SQuAD only; no unique mediator identified.
pre/post and periodic readers: batch / fusion changes	Independent batches and fixed-branch fusion controls (Sections 4.4 and 4.6)	Quality ordering depends on protocol; not a universal phase or publication rule.
ORA_{post}: full-model protection	Recovery gates hold; two-stage has higher quality (Table 31)	A negative quality boundary; partition, learning rate, and selected branch settings change together.
P: full-model observation	Terminal-only exceeds P on both tasks, eight new seeds/task (Table 4)	Direct negative transfer test; recipe differs from one-block; serial execution measures quality only.
P: shared vs. host-disjoint data	Equal-size halves, eight seeds/task; shared gains and disjoint reversals (Table 3)	Data relationship modifies the version effect; examples and relative order change together.

The evidence answers distinct questions across settings. Cross-family and larger-model checks support recovery; standard-split evaluation extends the earlier method's tested data; two prospective

studies confirm the specific $P - H$ contrast. Their combination supports a recovery capability and a bounded observation finding, not a general quality-transfer claim for P. Keeping these estimands separate lets the cross-setting evidence carry its actual meaning.

D.4 Reading the observation controls together

The controls explain why an unchanged host can yield different branch outcomes. First, recovery leaves the version assigned to each update unspecified even at fixed branch order. Replacing that assignment with terminal-only observation produces a quality difference while initialization, recipe, and batch order are held fixed (Table 1). Two fresh-seed P/H studies separately confirm the difference between prescribed and latest-publication observations at a common terminal base. Thus the recovery constraint defines a class of learning processes with measurably different outcomes.

Second, the samples paired with those versions affect the result. The preregistered replay on ten live seeds per task passes original-order reproduction gates on one seed per task before replaying retained version sequences with independent branch orders (Appendix N). The P/H gap shrinks in nine of ten GSM8K pairs and all ten SQuAD pairs, and the interaction passes on both tasks. Residual superiority under the prespecified rule is established only on SQuAD. The earlier three-seed exploration provides separate supporting evidence (Appendix M). The separate pre/post experiment finds practical equivalence under independent batches and a SQuAD interaction; its GSM8K interaction is unresolved. Third, the same learned branch can perform differently at another fusion base. Crossing never and periodic branches with s_0/s_T changes their ranking at fixed weights.

The equal-size data intervention adds a complementary boundary: P improves on terminal-only in the shared control but loses on host-disjoint data on both tasks (Table 3). Because example identity and relative order change together, this study alone cannot separate their contributions.

These interventions constrain interpretation without supplying a unique mechanism. The direct factorial P/H comparison changes example order, batch composition, and version–example pairing together. The fusion control concerns other policy members. The resulting knowledge is that preserved host training permits observation choices with repeatable quality consequences and measured dependencies. A useful next explanation must account for both the confirmed original-order gain and its attenuation, rather than infer quality from recovery or final-state exposure alone.

E Experimental Details

E.1 Protocol chronology and reproducibility records

The canonical and fresh hypothesis sets, statistical rules, and thresholds were fixed before their outcomes were opened. Selection used separate tune seeds, and fresh seeds 420–439 were opened after the lock. The valid full-horizon, scale-contract, and lifecycle protocols were locked before their retained results; the lifecycle seed correction was outcome-blind and is disclosed in Appendix E. The outcome-triggered decay extension treats 800–804 as pilot, 805–819 as the sole inferential sample, and pooled $n = 20$ as descriptive. The family and provenance matrices were preregistered on 2026-09-02 with write-once locks over protocol, code, data, environment, and reference runs, passed their bitwise replication gates, and were analysed once each after completion; their fresh seeds (15200–15219; 5100–5109 and 5150–5155) were opened only by the locked analyses. A third matrix (full-model protection) was preregistered on 2026-09-05 after its 2026-09-02 predecessor was voided by its own feasibility gate before any formal coordinate ran, and was analysed once on 2026-09-06. A fourth matrix (checkpoint-assisted observation) was preregistered on 2026-09-06 with its cadence, margin, and latency limit fixed before any outcome, passed a bitwise replication gate against three frozen Matrix II runs, and was analysed once on 2026-09-07; its fresh seeds (15300–15309) were opened only by the locked analysis. A fifth matrix (the staleness sweep) was preregistered on 2026-09-08 with a lock that binds the Matrix IV lock, plan, receipts, and analysis and every reused Matrix IV coordinate by file digest, passed a bitwise replication gate against two frozen Matrix IV runs, and was analysed once on 2026-09-10 (local time) after one coordinate whose trace log had been polluted by an execution incident was re-executed and reproduced its receipted digest bitwise (Appendix E.9). The frozen family analyser rejected standalone-arm receipts, which carry no fused evaluation; a two-line amendment that admits them was applied outside the locked tree and is disclosed with its diff and hashes in Appendix E.5 and the supplement.

The observation-sequence study is a post-outcome intervention on the six retained concurrent trajectories. Its source sequences, implementation, and inputs were frozen before execution; all historical replays passed before intervention training, and all training passed before evaluation. It uses the original holdout and provides descriptive retained-seed evidence. Its shared-GPU execution supplies no performance estimate (Appendix K.7).

Complete recipes, prompts, scoring, splits, and environment records are given in Appendix E; analysis roles and statistics are in Appendices E.4 and I; equality levels and formal boundaries are in Appendices A to C. The training code, the exact-online runners, and the eight-component audit suite of Appendix T (synthetic contract cells and real-stack falsification tests, run as reference/candidate roles from one seed and reporting the first divergent component) are released as anonymized supplementary material, with the environment lock and the scripts that regenerate every table from recorded run artifacts.

Table 10: Index of the preregistered matrices and follow-up studies referenced by Roman numeral in the main text.

Matrix	Question	Arms	Where
I	Canonical Qwen optional-LoRA: ORA_{post} vs. two-stage	post, two-stage, joint	Table 27
II	Family: pre/post vs. re-tuned two-stage, 20 fresh seeds	base, pre, post, two-stage	Appendix E.5
III	Full-model protection	base, post, two-stage	Appendix E.7
IV	Checkpoint-assisted observation, $K{=}250$	base, post, ckpt, two-stage	Appendix E.8
V	Staleness sweep over publication interval K	pre, $K \in \{1, 10, 1000\}$, never	Appendix E.9
VI	Independent branch batches for pre/post	pre_{own}, post_{own}	Appendix K
VII	Fusion base for fixed branches	never, $K{=}250$ at s_0/s_T	Table 34
VIII	Concurrent reader on six retained trajectories	H, P replay, C	Appendix K.7
—	Prospective P/H confirmations (live, shared)	H, P, C	Appendix L
—	Direct P/H batch-order intervention	original vs. independent order	Appendix M
—	Independent-order replay, ten live seeds	P_{own}, H_{own} vs. live P, H	Appendix N
—	Admissibility of C; two-stage at fixed batches	C branch on s_T, host alone, two-stage	Appendix P
—	Full-model observation, eight new seeds	P, matched terminal-only	Appendix O

E.2 Canonical BERT early-exit matrix

Table 11: Canonical BERT early-exit matrix (Section 2): disabling all exits does not recover the matched standalone artifact. Accuracy is on a 0–100 scale; paired differences and confidence intervals are in percentage points.

Task	Plain	Joint off	Gap	Paired 95% CI	Holm p
QNLI	91.37	89.94	$+1.431$	$[+1.304, +1.559]$	$1.91{\times}10^{-6}$
SST-2	92.77	91.20	$+1.565$	$[+1.361, +1.770]$	$1.91{\times}10^{-6}$

The canonical matrix compares four BERT-base-uncased systems with 12 exits: plain final-path training, PABEE-style depth-weighted joint training, a harmonized DeeBERT-structure two-stage baseline, and online ORA_{post} ramps. QNLI and SST-2 use their complete training and validation splits. Every method uses maximum length 128, batch size 32, learning rate 2×10^{-5}, no warmup, FP32, and five epochs per trainable stage. Seeds 200–219 determine matched initialization and epoch permutations; no development-set early stopping or threshold selection occurs. The primary canonical hypothesis set is plain minus joint final-path accuracy on both tasks with Holm correction.

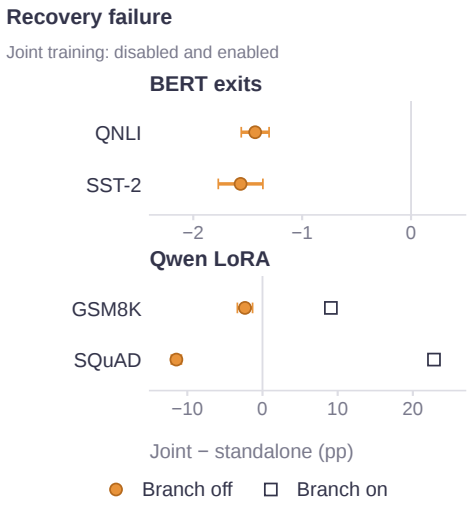

Figure 4: Joint minus standalone scores. BERT branch-off differences have confirmatory paired 95% intervals; Qwen branch-off and enabled comparisons are descriptive. BERT's adaptive enabled policy is omitted.

The ORA_{post} row supplies same-setting failure–repair evidence rather than an additional post-hoc primary hypothesis.

The two-stage row reproduces DeeBERT's structural separation rather than its original hyperparameters. It initializes independent intermediate pooler/classifier ramps from the matching plain final pooler, freezes the protected endpoint, and trains the ramps under the common five-epoch budget. Complete patience and entropy routing sweeps are secondary and do not select a headline threshold.

E.3 Qwen optional-LoRA matrix

The Qwen matrix uses Qwen2.5-0.5B-Instruct in eager FP32 execution. Its protected trainable partition contains decoder layer 23 and the final normalization; rank-8, scale-8 LoRA modules on query and value projections in decoder layers 0–22 form the branch-owned partition. The standalone/projected route disables these modules, while the fused route enables them. Protected and branch parameter sets are disjoint.

Protected training uses two epochs, batch size 8, learning rate 10^{-4}, a constant schedule, zero weight decay, and gradient-norm cap 1. GSM8K uses exact-answer accuracy; SQuAD uses F1 as primary and exact match as secondary. Before fit/tune/holdout assignment, examples are grouped by normalized question for GSM8K and normalized context for SQuAD, preventing group leakage across internal partitions. Assigned fit/tune/holdout counts are $5978/748/747$ for GSM8K and $18000/2003/5002$ for SQuAD; complete-group assignment makes the latter tune and holdout counts slightly exceed their $2000/5000$ targets. Two assigned GSM8K fit examples exceed the training-length limit, so the materialized tune-stage fit counts are $5976/18000$ for GSM8K/SQuAD. Fresh training uses materialized fit+tune counts $6724/20003$. These materialized counts determine optimizer-step totals. The standard GSM8K test and SQuAD validation splits remain reserved for the locked inference-only secondary evaluation.

The literal prompts are `Question: {question}\nAnswer:` for GSM8K and `Context: {context}\nQuestion: {question}\nAnswer:` for SQuAD; supervised targets begin with one space. Evaluation uses greedy single-beam decoding with no sampling, repetition penalty 1, and at most 256 new tokens for GSM8K or 64 for SQuAD. GSM8K scoring extracts the last complete, finite numeric token from the generation and gold answer and tests decimal equality. SQuAD takes the first nonempty generated line, applies the standard lowercase, punctuation, article, and whitespace normalization, and reports the maximum exact match and token F1 over reference answers.

The tuning grid is symmetric in branch opportunity. At a 1x budget, ORA_{post} executes one branch update after every protected update; 2x and 4x add independent branch-only updates without changing

protected transition count. Two-stage uses the corresponding total branch-update budgets after completing the protected endpoint. Both tune branch learning-rate multiplier $\{0.5, 1, 2\}$. The fixed cross-task rule selects multiplier 2 and the 1x branch budget for both methods: one branch update after every protected update for ORA_{post}, and two branch-only epochs after the protected endpoint for two-stage. The selection score averages each task-specific fused metric over tune seeds and then across tasks, with lower wall time and lower budget as fixed tie-breakers.

E.4 Derived secondary analyses of the fresh matrix

Every result in this subsection is a read-only derivation from completed fresh runs. No training, evaluation, or selection was rerun, and the frozen experiment tree remains unchanged.

Projected recovery gap. Base-minus-joint path-off comparisons on GSM8K and SQuAD belonged to an earlier prespecified LLM hypothesis set. Before fresh outcomes were opened, the selection lock replaced that set with fused ORA_{post} minus two-stage as the sole fresh primary set. Because every locked seed still evaluates the projected routes, we report the original comparisons descriptively rather than revive a second confirmatory set. Intervals, exact sign-flip tests, and Holm-adjusted values retain pairing and uncertainty information. Compared rows have identical protected optimizer-step counts: 1682 on GSM8K and 5002 on SQuAD. The PCGrad and exact rows in Table 22 are also descriptive.

Full-run protected observable. Every training step logs protected loss before the protected update. A step enters the comparison only when its optimizer update touches protected parameters: dedicated protected updates for standalone, ORA_{post}, and two-stage, and coupled updates for a joint-style row such as PCGrad. Excluding branch-only updates aligns sequences across branch budgets; logged values are compared by exact equality. Each method contributes 33,640 GSM8K and 100,040 SQuAD paired comparisons. Full-run loss equality is necessary for complete-state equality but does not substitute for it; uninterrupted full-horizon traces provide stronger positive evidence for their pinned configurations.

Branch generalization diagnostic. Terminal branch training loss averages logged fused loss over the final 200 branch updates and then over 20 fresh seeds. It is 0.236 for ORA_{post} versus 0.113 for two-stage on SQuAD, and 0.357 versus 0.282 on GSM8K; corresponding holdout scores are 77.01 versus 67.11 and 43.04 versus 39.55. The comparison holds branch update count, branch data, and the bitwise-identical protected endpoint fixed. It remains a post-outcome descriptive analysis without a preregistered hypothesis, threshold, or hypothesis-set membership.

Tuning-grid separation. The frozen tune-only artifact over seeds 400–409 shows complete grid separation. Nine ORA_{post} variants score 0.585–0.600 and nine two-stage variants score 0.528–0.542 on the cross-task tune score. No fresh, internal-holdout, or standard-split data enter this calculation.

QNLI operational workflow verdict. The preregistered QNLI workflow meets its identity and quality requirements (ORA_{post} minus two-stage macro -0.177 *pp*, inside the locked -1 *pp* margin) but misses the locked time-ratio 0.90 and 900-second thresholds (Appendix E); where no lifecycle constraint binds, nothing argues for ORA on cost.

E.5 Preregistered family matrix (Matrix II)

Protocol `ORA_FAMILY_EXACT_ONLINE_TWO_STAGE_20260902_V1` was frozen on 2026-09-02 before any of its outcomes existed. A write-once lock binds the protocol document, configuration, runner, evaluator, orchestrator, analysis code, the frozen source tree, task configurations, data tensors and manifests, the environment, and the eleven Matrix I runs used as bitwise references; every entry point re-verifies the lock. Five arms share seeds and the protected stream of Appendix E: *base* (standalone), *pre* (pre-transition read-only, Equation (6)), *post* (ORA_{post}), *ts_ref* (the Matrix I two-stage cell: multiplier 2, two branch epochs, no weight decay, constant schedule), and *ts_sel*, a two-stage cell selected on the tune partition from a 24-cell grid over multiplier $\{2, 4, 8\}$, branch epochs $\{1, 2\}$, branch weight decay $\{0, 0.1\}$, and schedule {constant, linear decay} (2 tasks $\times$ 4 tune seeds 400–403, 192 runs). The exact-online arms keep their Matrix I configuration; additional tuning

opportunity is allocated to two-stage. Three gates preceded fresh confirmation: a twelve-coordinate smoke, bitwise replication of the frozen Matrix I GSM8K seed-420 rows (protected digests, every trainable tensor, every generated token, every score), and bitwise replication of the eight Matrix I two-stage reference cells inside the tune stage; all passed. The selection objective (unweighted mean over tasks of the four-seed tune metric; tie-breaks fewer epochs, lower decay, constant schedule, lower multiplier) chose multiplier 8, two epochs, no weight decay, linear decay. This multiplier is the search boundary, and selection optimizes a cross-task mean rather than either task separately. The grid tests shorter integer-epoch training, decay, and weight decay; it does not cover fractional-epoch early stopping or establish a globally optimal two-stage recipe. Fresh seeds 15200–15219 are disjoint from every earlier seed; 200 fresh runs completed on 2026-09-05, before the locked deadline of 2026-09-20.

The exactness precondition requires each of pre, post, ts_ref, and ts_sel to equal base on every fresh seed in per-step protected loss digests, endpoint component digests, checkpoint components, and the protected parameter hash; all four arms are exact in $40/40$ seed–task pairs. Table 12 reports the four locked hypothesis sets under the rules of Appendix I. Descriptive held-out means (fused route, 0–100 scale) are pre $44.18/78.27$, ORA_{post} $42.72/76.58$, ts_sel $39.52/71.50$, and ts_ref $40.29/66.07$ on GSM8K/SQuAD; the re-tuned two-stage cell gains $5.4\,pp$ of SQuAD F1 over the Matrix I cell, which is why the ORA_{post} advantage over two-stage is $+5.09\,pp$ here against $+9.90\,pp$ in Matrix I, while the Matrix I cell itself replicates (family R). Two-stage was re-tuned over multiplier, epochs, weight decay, and schedule (192 tune runs), and the selected cell (multiplier 8, two epochs, linear decay) loses to both members on both tasks. The re-tuned two-stage cell is a lower bound on tuned two-stage: its multiplier sits at the grid boundary, and cross-task selection makes it weaker than the Matrix I cell on GSM8K (39.52 vs. 40.29). In Matrix I, ORA_{post} gains $+3.487/+9.902\,pp$ under symmetric tuning (CIs $[+2.553, +4.421]/[+9.155, +10.649]$; Table 27), and its reference cell replicates ($+2.430/+10.511\,pp$) on Matrix II seeds, so the GSM8K margin over the strongest tested two-stage is $+2.430\,pp$.

Batch coupling. With branch budget 1, every online branch update (ORA_{pre}, ORA_{post}, and the checkpoint readers) trains on the protected batch x_t of the transition it accompanies (`batches = [base_batch]` in the family and checkpoint runners; the checkpoint reader is trained on `base_batch`), whereas the two-stage branch draws its own permutation from the branch data generator. ORA_{post}'s update therefore always fits a batch the protected step has just fit and ORA_{pre}'s one it has not, a batch-freshness difference that Matrix II does not separate from the boundary read. Matrix VI tests independent branch streams and reports their phase comparison and quality changes (Section 4.4 and Appendix K).

Table 12: Preregistered family matrix: locked hypothesis sets on fresh seeds 15200–15219, $n = 20$ per task. Paired differences in percentage points with two-sided 95% t-intervals and Holm-adjusted exact one-sided sign-flip p over the two tasks; H2 is the symmetric hypothesis set, shown in the direction that passed. A directional family is SUPPORTED only if the mean exceeds $1\,pp$, the interval excludes 0, and Holm $p < 0.05$ on both tasks.

Family	Contrast	Task	Diff.	Paired 95% CI	Holm p	Verdict
H1	$\text{ORA}_{\text{post}} - \text{ts_sel}$	GSM8K acc.	$+3.199$	$[+2.269, +4.130]$	2.86×10^{-6}	SUPPORTED
		SQuAD F1	$+5.088$	$[+4.677, +5.499]$	1.91×10^{-6}	
H1′	pre − ts_sel	GSM8K acc.	$+4.652$	$[+3.661, +5.643]$	1.91×10^{-6}	SUPPORTED
		SQuAD F1	$+6.775$	$[+6.380, +7.169]$	1.91×10^{-6}	
H2	$\text{pre} - \text{ORA}_{\text{post}}$	GSM8K acc.	$+1.452$	$[+0.973, +1.932]$	1.34×10^{-5}	pre favoured
		SQuAD F1	$+1.687$	$[+1.468, +1.906]$	1.91×10^{-6}	
R	$\text{ORA}_{\text{post}} - \text{ts_ref}$	GSM8K acc.	$+2.430$	$[+1.388, +3.472]$	2.67×10^{-5}	SUPPORTED
		SQuAD F1	$+10.511$	$[+9.436, +11.585]$	1.91×10^{-6}	

Analysis amendment. When the locked analysis first ran, its fresh-matrix loader validated every coordinate through the evaluator's completed-run check, whose final clause requires a fused evaluation; the standalone *base* arm has none by design, so the run stopped before reading any score, wrote no outcome artifact, and left the lock valid. The frozen tests had exercised the fresh analysis only on its refusal path. We applied a two-line amendment outside the locked tree: the completed-run check gains an `allow_standalone` flag that returns a complete base run after the receipt, artifact-hash,

spec, and full-horizon checks, and the fresh-cell loader passes it. No hypothesis, decision rule, statistic, selection, deadline handling, or claim-boundary sentence changed; the tune-stage loader is untouched; the original suite passes against the amended code together with one added regression test. The analysis then ran once. The amendment record, its diff, the digests of the original and amended files, and the analysis log are included in the supplement.

E.6 PREREGISTERED PROVENANCE MATRIX

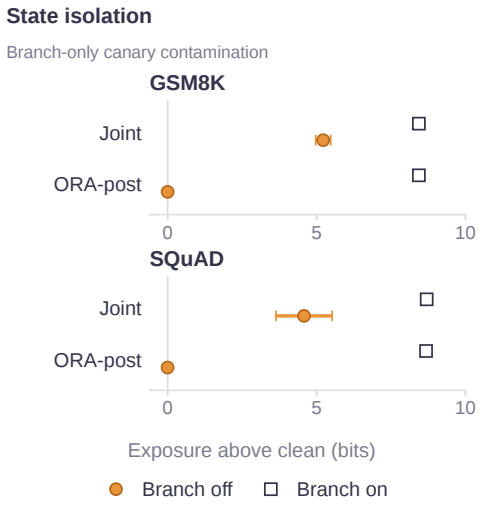

Figure 5: Branch-only contamination. Both enabled routes learn canaries; ORA-post has zero excess exposure after switch-off. Bars are paired 95% intervals across 10 GSM8K and 6 SQuAD seeds. State equality is checked independently.

Protocol `ORA_PROVENANCE_NO_TRACE_20260902_V1` was frozen on 2026-09-02 with the same lock discipline. Three arms share seeds, the protected stream of Appendix E in its fresh configuration, and one contaminated stream D_c: *base* never reads D_c; *post* (ORA_{post}) takes one branch-only step per protected transition on a D_c batch, reading s_{t+1} (branch learning-rate multiplier 2); *joint* takes one shared AdamW step per transition on the protected loss plus the fused D_c loss. D_c is built once on CPU and bound by the lock: 64 canary labels with distinct eight-digit secrets that never appear in a prompt, four identical copies each (256 rows), plus the complete held-out partition (747 GSM8K, 5,002 SQuAD rows); post and joint draw the same D_c batches in the same order. Canary exposure is $\log_2 1000 - \log_2(\text{rank})$ of the secret's summed NLL among 999 length-matched alternatives, averaged over the 64 canaries (maximum 9.97 bits); held-out NLL is the mean supervised-token NLL over the held-out partition; greedy generation is scored descriptively. Seeds are 5100–5109 (GSM8K) and 5150–5155 (SQuAD), 48 coordinates.

The locked precedence is exactness, then exposure control, then trace. *Exactness* (post = base on the per-step protected loss digest sequence, every endpoint component, the protected parameter hash, and every shared branch-disabled score) holds on $16/16$ seeds; joint's protected parameters differ from base on every seed. *Exposure control* (branch-enabled exposure minus base exposure ≥ 1 bit for each contaminated arm) passes with $8.44/8.44$ bits (GSM8K, ORA_{post}/joint) and $8.68/8.70$ (SQuAD). *Trace* (joint minus base with the branch disabled) passes on both tasks: exposure $+5.223$ bits $[4.973, 5.474]$ and $+4.579$ $[3.636, 5.523]$; held-out NLL -0.217 $[-0.221, -0.214]$ and -0.336 $[-0.346, -0.326]$; independently enumerated Holm-adjusted exact sign-flip $p = 0.001953125 < 0.002$ on GSM8K and $p = 0.03125$ on SQuAD for both metrics. Descriptively, joint's branch-disabled held-out accuracy/F1 exceeds base by $3.90/9.53\,pp$ because the held-out set was in D_c. Table 13 lists the per-arm means.

Numerical audit. The frozen analyser records GSM8K exposure raw $p = 0$ and adjusted values 0 (exposure) and 0.0009765625 (NLL). Its dot-product enumeration and separate observed-sum reduction can differ by more than the fixed 10^{-15} comparison tolerance, excluding the observed

assignment itself. Exact enumeration must include that assignment. Independent enumeration from the released seed records gives raw 2^{-10} for each metric and Holm-adjusted $2^{-9} = 0.001953125$ for both. The frozen artifacts are preserved; this numerical correction changes neither the decision nor the reported $p < 0.002$ bound. The manuscript audit checks these discrepancies explicitly.

Table 13: Preregistered provenance matrix: per-arm means over seeds (10 GSM8K, 6 SQuAD). Exposure in bits over 64 canaries (maximum 9.97); held-out NLL per supervised token; metric is held-out GSM8K accuracy / SQuAD F1 on a 0–100 scale. "Off"/"on": branch disabled/enabled; branch-disabled columns describe the protected process alone. ORA_{post}'s branch-disabled SQuAD generation is not scored because its protected parameters equal base's bitwise.

Task	Arm	Exposure (off)	Exposure (on)	Held-out NLL (off)	Metric (off)
GSM8K	base	1.53	–	0.452	37.0
	ORA_{post}	1.53	9.97	0.452	37.0
	joint	6.75	9.97	0.234	40.9
SQuAD	base	1.27	–	0.891	55.4
	ORA_{post}	1.27	9.94	0.891	–
	joint	5.85	9.97	0.555	64.9

E.7 Preregistered full-model matrix (Matrix III)

Protocol `ORA_FULL_MODEL_PROTECTED_20260905_V2` was frozen on 2026-09-05 with the lock discipline of Appendix E.5. Its predecessor (`. . . _20260902_V1`, frozen 2026-09-02) was voided on 2026-09-04 by its own feasibility gate before any formal coordinate ran: the locked projection rule timed smoke coordinates that hash the full protected state around every branch step and projected 443 GPU-hours against a 200-hour limit. V2 keeps the scientific protocol unchanged and adds timing coordinates that run the formal evidence settings; its gate passed at a projected 112 GPU-hours. Both protocols, locks, and the gate record are in the supplement.

The protected process is the entire Qwen2.5-0.5B-Instruct model (every non-LoRA parameter trainable) at learning rate 2×10^{-5}, otherwise the recipe of Appendix E; the rank-8 LoRA branch is unchanged. Arms are *base*, *post* (ORA_{post}), and *two_stage*. Tune: 36 coordinates (2 tasks $\times$ seeds 5300–5301 $\times$ 9 cells: ORA_{post} at branch multiplier $\{5, 10, 20\}$ with one branch update per protected step; two-stage at multiplier $\{5, 10, 20\}$ $\times$ branch epochs $\{1, 2\}$). Selection, written once before any fresh coordinate, maximizes the unweighted task mean of the seed-mean tune metric (tie-breaks: lower multiplier; fewer epochs) and refuses if the tune cells of any task–seed disagree on the protected endpoint; it chose multiplier 10 with one update per step for ORA_{post} and multiplier 10 with two epochs for two-stage, and all 36 tune coordinates shared their protected endpoints. Fresh: 60 coordinates (2 tasks $\times$ seeds 5400–5409 $\times$ three arms), completed on 2026-09-06 before the locked deadline; every coordinate ran as one uninterrupted invocation. Decision rules: exactness (no tolerance) precedes completeness, which precedes the utility rule (mean $> 1\,pp$, interval above 0, Holm-adjusted one-sided sign-flip $p < 0.05$ on both tasks).

Verdict. Exactness SUPPORTED, utility NOT SUPPORTED. The licensed text: "With the entire base model protected (full-parameter fine-tuning of the base path), the exact-online construction and the two-stage construction reproduced the standalone protected process bit for bit on all 10 fresh seeds of both tasks: identical per-step protected loss digests, identical endpoint protected components, and identical protected parameter hashes. On the held-out split, the selected exact-online coordinate did not exceed the selected two-stage coordinate by the preregistered margin on either task (gsm8k: mean -5.96 pp, 95% CI $[-7.25, -4.67]$, Holm-adjusted one-sided $p = 1.0000$; squad: mean -4.15 pp, 95% CI $[-4.85, -3.44]$, Holm-adjusted one-sided $p = 1.0000$); the paper may report exactness under full-model protection and must state that the utility advantage observed with [the block-protected setting of Matrices I–II] did not replicate here." The bracketed phrase replaces the locked wording "adapter-protected bases": in Matrices I–II the protected partition is the final decoder block and normalization and the LoRA modules are the optional branch, so the earlier setting is block-protected. The lock is unchanged and the correction is recorded in the supplement. Per-arm held-out fused means (descriptive, recorded by the protocol outside its rules) are ORA_{post} $27.15/69.19$ and two-stage $33.11/73.33$ on GSM8K/SQuAD; the standalone arm's projected route was not scored in this matrix. Against the block-protected arms of Table 31, ORA_{post}'s fused route is $15.6/7.4\,pp$ lower

and two-stage's $6.4\,pp$ lower on GSM8K but $1.8\,pp$ higher on SQuAD; on GSM8K both lie below the block-protected standalone branch-disabled route of the provenance matrix (37.0, Table 13); the matrix does not identify whether the protected partition, the protected recipe (learning rate 2×10^{-5}, constant schedule, no warm-up), or the selected branch configuration produced the lower level or the reversal.

Operational accounting. From the 60 run receipts (descriptive; outside the hypothesis sets): median training wall time per coordinate is $26.3/43.8/45.5$ min for base/ORA_{post}/two-stage on GSM8K and $78.4/130.6/136.0$ min on SQuAD, plus $48/17$ min of fused evaluation for each non-base coordinate; peak reserved memory is 0.338 of the 80 GiB device for base on both tasks and two-stage on GSM8K, and 0.367 for ORA_{post} on both tasks and two-stage on SQuAD. The matrix does not test the pre-transition member or joint training, and its exactness evidence is per-step loss digests plus endpoint components and hashes, not a complete intermediate-state trace.

E.8 Preregistered checkpoint matrix (Matrix IV)

Protocol `ORA_CHECKPOINT_BRANCH_BASELINE_20260906_V1` was frozen on 2026-09-06 with the lock discipline of Appendix E.5; its lock additionally binds the Matrix II lock, the Matrix II two-stage selection, and three Matrix II runs used as bitwise references. The question is operational: every training recipe already retains periodic checkpoints, so a branch could train against them off the protected run's critical path instead of paying ORA_{post}'s serial branch update inside the protected pass. The requirement variables were fixed before any outcome: the recipe's own checkpoint cadence $K = 250$ protected steps, one branch update per protected step for every online arm, equal branch budgets for all branch arms, a practical margin of $1\,pp$, and a protected-latency requirement of at most $1.05\times$ the standalone protected pass.

Five arms share seeds and the fresh protected stream of Appendix E: *base* (standalone); *post* (ORA_{post} at its Matrix I–II anchor, multiplier 2, constant schedule); *ckpt* (Equation (9) with $K = 250$): the live instance runs the standalone process and publishes its protected state after every K transitions into a second model instance, the reader, which holds the most recently published state and owns the branch, so the branch update after transition t reads $s_{K\lfloor (t+1)/K \rfloor}$ (publication is a tensor copy, the in-process form of a second process polling the checkpoint directory); *ckpt_sensitivity* ($K = 50$, secondary, entering no primary decision); and *two_stage*, the Matrix II selection (multiplier 8, two branch epochs, no weight decay, linear decay), whose $2\lceil n/8 \rceil$ branch updates equal the online arms' T. Release snapshots of post and ckpt at $\lfloor 0.5T \rfloor$ and $\lfloor 0.75T \rfloor$ (steps $841/1261$ on GSM8K, $2501/3751$ on SQuAD) materialize the fused route available at that moment, the live protected state together with the branch state of a reader that has kept up (one update per protected step); a release does not advance the reader. Only the ckpt branch multiplier was tuned ($\{2, 4, 8\}$, tune seeds 400–401, 12 coordinates; objective the unweighted task mean of the seed-mean tune metric, tie-break lower multiplier), selecting 2; post and two-stage keep their earlier selections, an asymmetry that can only favour the new baseline.

Three gates preceded the fresh matrix. A ten-coordinate smoke (seed 15299, four protected steps, publication intervals 2 and 1, per-step evidence) requires every non-standalone arm to reproduce base bitwise per step, each reader to read $s_{K\lfloor (t+1)/K \rfloor}$ with its protected hash equal to base's hash at that step before every branch update, and the $K = 1$ arm to coincide with post. A reference gate requires GSM8K seed-15200 base, post, and two-stage to reproduce the frozen Matrix II runs bitwise: every trainable tensor, protected hash, endpoint components, per-step loss digests, and, for the branch arms, every fused generation and score. The tune completion receipt precedes the write-once selection. All passed. Fresh seeds 15300–15309 are disjoint from every earlier seed; 100 coordinates completed on 2026-09-07, before the locked deadline of 2026-09-20, and the analysis ran once.

Decision rules. Exact paired sign-flip tests over ten seeds (2^{10} enumeration), two-sided 95% t-intervals, Holm over the two tasks within a hypothesis set. A symmetric set is LEFT FAVOURED if the left arm exceeds the right by more than $1\,pp$ with interval above zero and Holm $p < 0.05$ on both tasks, RIGHT FAVOURED for the reverse, PARITY if both intervals lie inside $(-1, 1)\,pp$, and otherwise MIXED, reported per task without a summarized direction. H_{C0}: post, ckpt, ckpt_sensitivity, and two_stage equal base on every fresh seed in per-step protected loss digests, endpoint components, checkpoint components, and protected parameter hash, the ckpt arms additionally recording that the

branch never wrote protected state and read only published states; any mismatch voids every family involving that arm. H_{C1}: post − ckpt, terminal fused metric. H_{C2}: ckpt − two_stage. H_{C3}: post − ckpt at the 75% release. R1: wall-clock of the work on the protected pass's critical path (protected steps plus serial branch updates for post; protected steps plus publications for ckpt, whose branch runs on the reader; protected steps for two_stage) divided by the standalone pass of the same seed and task, median over seeds and tasks, requirement ≤ 1.05; post was expected to violate it and ckpt to meet it. Descriptive: the 50% release, the sensitivity arm against ckpt and post, post against two_stage, staleness, the reader's keep-up ratio (branch wall time over protected wall time), and the bytes of one published protected state.

Verdict. H_{C0} holds for all four arms in $20/20$ seed–task pairs; H_{C1} and H_{C3} are MIXED; H_{C2} favors checkpoint training over selected two-stage (Table 14). The mixed outcomes establish neither a task-general ordering nor equivalence within $\pm 1\,pp$ at the observed precision. R1 is met by ckpt, ckpt_sensitivity, and two_stage and violated by post under the serial component accounting defined above. This evaluates protected-path work, with separate reader progress considered below. The matrix includes post; pre is added on these seeds in Matrix V. Descriptive held-out fused means are post $43.48/76.55$, ckpt $43.63/77.30$, ckpt_sensitivity $43.23/77.86$, and two_stage $39.75/71.88$ on GSM8K/SQuAD; post minus two-stage is $+3.735/+4.665\,pp$. The original machine-generated verdict and wording remain in the released analysis records; scientific interpretations here retain the comparison and execution boundaries.

Numerical audit. The frozen analyser enumerates the 2^{10} sign assignments with a floating-point comparison whose tolerance excludes some assignments whose sum exactly ties the observed sum; GSM8K differences are multiples of $100/747$, so ties occur. Independent enumeration that counts exact ties gives larger raw and Holm-adjusted values in five GSM8K cells: H_{C1} R>L $0.354 \to 0.373$, H_{C3} L>R $0.777 \to 0.781$, the 50% release L>R $0.201 \to 0.209$, ckpt_{50}−ckpt R>L $0.406 \to 0.430$, and ckpt_{50}−post R>L $0.605 \to 0.621$. Table 14 reports the tie-inclusive values; every other cell agrees to machine precision, no value is within 0.15 of the 0.05 threshold, no verdict changes, and the frozen artifacts are preserved. The manuscript audit checks both sets.

Table 14: Preregistered checkpoint matrix: locked hypothesis sets (H_C) and descriptive contrasts (D) on fresh seeds 15300–15309, $n = 10$ per task. Paired differences in percentage points with two-sided 95% t-intervals and Holm-adjusted exact one-sided sign-flip p over the two tasks for each direction (L>R: left exceeds right). ckpt_{50} is the $K = 50$ sensitivity arm; release contrasts compare the fused route materialized at $\lfloor 0.75T \rfloor$ or $\lfloor 0.5T \rfloor$. Protected latency is the median ratio to standalone over 20 seed–task pairs against the locked limit 1.05.

Contrast	Task	Diff.	Paired 95% CI	Holm L>R	Holm R>L	Verdict
H_{C1}: post − ckpt	GSM8K	−0.147	[−1.053, +0.758]	1.000	0.373	MIXED
	SQuAD F1	−0.755	[−1.635, +0.125]	1.000	0.084	
H_{C2}: ckpt − two-stage	GSM8K	+3.882	[+1.938, +5.826]	2.93×10^{-3}	1.000	ckpt favoured
	SQuAD F1	+5.420	[+4.560, +6.280]	1.95×10^{-3}	1.000	
H_{C3}: post − ckpt at 75%	GSM8K	−0.388	[−1.596, +0.819]	0.781	0.477	MIXED
	SQuAD F1	+1.351	[−2.529, +5.231]	0.771	0.615	
D: post − ckpt at 50%	GSM8K	+0.576	[−0.361, +1.512]	0.209	1.000	descriptive
	SQuAD F1	+0.098	[−0.550, +0.747]	0.371	1.000	
D: ckpt_{50} − ckpt	GSM8K	−0.402	[−1.481, +0.678]	0.801	0.430	descriptive
	SQuAD F1	+0.555	[−0.493, +1.603]	0.283	0.859	
D: ckpt_{50} − post	GSM8K	−0.254	[−1.374, +0.866]	0.709	0.621	descriptive
	SQuAD F1	+1.310	[+0.638, +1.982]	5.86×10^{-3}	0.998	
D: post − two-stage	GSM8K	+3.735	[+2.131, +5.339]	1.95×10^{-3}	1.000	descriptive
	SQuAD F1	+4.665	[+4.191, +5.138]	1.95×10^{-3}	1.000	

R1 latency (× standalone, median): post 3.18 (violates 1.05); ckpt 1.01; ckpt_{50} 1.01; two-stage 1.00.

Release-point caveats. Three descriptive facts qualify H_{C3} and the 50% contrast. First, the SQuAD interval at 75% is carried by one seed: seed 15309's checkpoint snapshot scores 57.17 F1 against ORA_{post}'s 73.59 (its 50% snapshot 75.91 and terminal 76.56 are unremarkable), and the other nine seeds average $-0.32\,pp$; a one-in-ten collapse of the materialized fused route is itself a property of

the checkpoint member at that step that the matrix records but does not explain. Second, the release steps sit at different phases of the publication cadence: SQuAD's 2501 and 3751 fall one update after a publication, so the snapshot pairs the live state with a branch adapted to a state 250 steps older, whereas GSM8K's 841 and 1261 fall 91 and 11 updates after one; the locked release fractions did not control phase. Third, at $n = 10$ the PARITY verdict required both intervals inside $(-1, 1)\,pp$, which with the observed spread meant a GSM8K mean within about $0.1\,pp$ (half-width 0.906; observed -0.147), so the MIXED verdicts are also statements of power. Descriptively, both members peak at the 50% release and decline to the terminal state (ORA_{post} $45.27/78.30 \to 43.48/76.55$, ckpt $44.70/78.20 \to 43.63/77.30$), as does the standalone projection under the constant schedule (Appendix E.10); whether terminal two-stage is disadvantaged by fitting a degraded endpoint is untested here.

Table 15: Historical cost ledger (Matrices I, IV, and V). Metrics come from distinct protocols and are not a matched end-to-end ranking. GSM8K/SQuAD values; actual concurrent costs are in Appendix K.5.

Construction	Critical path	Wall-clock (s)	Device time	Peak (GiB)	Extra state (MiB)	Staleness
Standalone	1.00	642/1978	1.0	11.9	0	—
ORA_{pre} (V)	2.89	—	—	—	—	live (s_t)
ORA_{post}	3.18	2026/7254	3.2	21.7	1.98	live (s_{t+1})
ORA_{ckpt}, K=1 (V)	2.74	—	—	—	read-only copy	0.0/0.0
ORA_{ckpt}, K=10 (V)	1.81	—	—	—	read-only copy	4.5/4.5
ORA_{ckpt}, K=50	1.01	—	—	23.7	read-only copy	24.3/24.5
ORA_{ckpt}, K=250	1.01	—	3.2	23.7	read-only copy	120.9/124.5
ORA_{ckpt}, K=1000 (V)	1.00	—	—	—	read-only copy	435.4/499.3
ORA_{ckpt}, never (s_0) (V)	1.00	—	—	—	read-only copy	841.5/2501.5
Two-stage	1.00	—	—	21.7	—	terminal (s_T)
Joint	—	1063/3194	—	—	—	—
Clean + joint	—	1704/5172	—	—	58.87	—

Notes. Qwen2.5-0.5B block-protected unless stated (GSM8K/SQuAD). Critical path: accounted protected-path work relative to the standalone pass, using serial component timings (median over seeds and tasks; Matrix IV receipts; ORA_{pre} and K=1, 10, 1000, and never from Matrix V receipts against the same-session standalone pass, measured on a shared node, Appendix E.9). Wall-clock: training seconds per coordinate (Matrix I two-checkpoint study, Table 24; standalone 641.62/1977.90 s). Device time: total device time in standalone passes (Matrix IV). Peak memory: reserved GiB (Matrix IV medians). Rows marked V use pooled shared-node ratios and do not establish speedups. Extra state: saved trainable-state delta in MiB beyond the immutable snapshot (Table 24); the checkpoint member keeps one read-only protected copy resident instead. Staleness: mean protected steps between the state read and the live state. "—": not measured in the locked protocol that produced the row. The QNLI operational workflow (Appendix E) records a time ratio of 0.9281 against the locked 0.90 and 296.39 s against 900 s.

Operational accounting. From the run receipts (descriptive): the standalone protected pass takes a median 10.6/31.6 min on GSM8K/SQuAD; the T branch updates take 22.5/69.5 min for post and 22.9/69.8 min for ckpt, so the reader's keep-up ratio is 2.20 at $K = 250$ and 2.10 at $K = 50$ (medians over 20 coordinates): the protocol's check that a second process would keep pace (ratio below one) fails, a matched-hardware reader taking one update per protected step finishes about 1.2 protected passes after the live run and would hold about $0.34T$ updates at the 75% release, and total device time (protected, branch, and publication) is 3.1–3.2× the standalone pass for post and ckpt alike. Publication costs below one minute per coordinate. Peak reserved memory is 11.9 GiB for base, 21.7 GiB for post and two-stage, and 23.7 GiB for the ckpt arms (medians), so the reader instance adds about 2 GiB beside the live process. Mean staleness is 120.9/124.5 protected steps at $K = 250$ and 24.3/24.5 at $K = 50$; one published protected state occupies 56.9 MiB of parameters and buffers in this partition, and the reader instance holds it beside the live process. The matrix does not test the pre-transition member's latency (serial by construction, hence expected to violate the requirement), cadences other than 250 and 50, a reader that falls behind the live process (the realised release-point and terminal scores are those of a caught-up reader), a reader on a second device, or a file-system publication path; the release snapshots rely on the recipe's checkpoint at the release step; and ten seeds do not establish equivalence between post and ckpt, whose terminal interval reaches $-1.6\,pp$ on SQuAD.

E.9 PREREGISTERED STALENESS SWEEP (MATRIX V)

Protocol `ORA_STALENESS_SWEEP_20260908_V1` was frozen on 2026-09-08 (lock `99371530...`) with the lock discipline of Appendix E.5; its lock additionally binds the Matrix IV lock, fresh plan, completion receipt, selection, analysis, and verdict, and every reused Matrix IV fresh coordinate by file digest (100 cells: spec, result, receipt, trainable state, protected endpoint checkpoint, components, trace, and for evaluable arms the fused predictions, scores, and evaluation receipt). The registered questions concern per-step-publication identity, live versus periodic observation, and periodic versus fixed-initial-state observation. Their answers are interpreted under the phase, same-batch, and fusion conditions of Sections 3 and 4.4. The original question and automatic interpretation text remain in the frozen protocol and verdict records.

Six new arms run on the Matrix IV seeds 15300–15309 with the fresh protected stream of Appendix E, branch multiplier 2 and budget 1 throughout (the Matrix IV selected cell, also the Matrix I–II anchor; nothing is tuned): *base* (standalone, re-run for a same-session wall-clock reference), *pre* (Equation (6)), and Equation (9) at $K{=}1$ (publication after every transition, predicted to reproduce $\mathrm{ORA}_{\mathrm{post}}$ bit for bit), $K{=}10$, $K{=}1000$, and *never* ($K{=}0$ in the runner: the reader holds s_0 for the whole horizon, so the branch trains against the initial protected state and is fused with s_T). The frozen Matrix IV coordinates supply $\mathrm{ORA}_{\mathrm{post}}$, $K{=}250$, $K{=}50$, two-stage, and the exactness anchor for the re-run base. No release snapshots are taken.

Three gates preceded the fresh matrix. A twelve-coordinate smoke (seed 15299, four protected steps, cadences 1, 2, and never, per-step evidence) requires every non-standalone arm to reproduce base bitwise per step, $\mathrm{ORA}_{\mathrm{pre}}$ to read s_t and $\mathrm{ORA}_{\mathrm{post}}$ to read s_{t+1} at every transition, each reader to follow $\kappa(t)$ exactly (all zero for never), the $K{=}1$ branch state to equal $\mathrm{ORA}_{\mathrm{post}}$'s, and every other pair of branch states to differ. A reference gate requires the extended runner to reproduce the frozen Matrix IV $\mathrm{ORA}_{\mathrm{post}}$ and $K{=}250$ coordinates of GSM8K seed 15300 bitwise (trainable state, protected hash, loss digests, every fused generation and score); it passed ($747/747$ generations equal for both). The fresh stage ran 120 coordinates on eight GPUs; the deadline was 2026-09-14T12:00Z.

Decision rules. Statistics as in Appendix E.8. H_{D0} (exactness precondition): every new arm equals the re-run base, and the re-run base equals the frozen Matrix IV base, on the initial protected hash, the per-step loss digests, endpoint components, endpoint checkpoint components, and protected parameter hash; no tolerance. H_{D1} (identity): on every seed of both tasks, the $K{=}1$ coordinate and the frozen $\mathrm{ORA}_{\mathrm{post}}$ coordinate have identical trainable-state tensors, identical fused generations and per-example scores, and an identical aggregate metric (IDENTICAL only if $20/20$). H_{D2} ($\mathrm{ORA}_{\mathrm{pre}} - K{=}250$; symmetric) and H_{D3} ($K{=}250 -$ never; symmetric) on the terminal held-out fused metric. R1: protected-latency ratio to the same-session standalone pass, median ≤ 1.05, per new arm. The sweep curve, the remaining pairwise contrasts, and the per-seed Spearman correlation between cadence rank and quality are descriptive and carry no direction. Predictions recorded at freeze: H_{D1} IDENTICAL; H_{D2} LEFT FAVOURED or MIXED; H_{D3} LEFT FAVOURED; $\mathrm{ORA}_{\mathrm{pre}}$ and $K{=}1$ at or above the latency limit, the coarser cadences below it.

Execution note. The node was shared. Twelve coordinates failed at their first step with CUDA out-of-memory when another user's jobs took 42–50 GiB per GPU, and the orchestrator's locked three-consecutive-failures rule retired four devices; the run was re-invoked on the same immutable plan (completed coordinates reused by receipt, interrupted ones resumed from their 250-step checkpoints, failed ones restarted from scratch). A second interruption was caused by our own supervisor's watchdog. A third followed from it: one coordinate (SQuAD, $K{=}1000$, seed 15301) had been killed before its first checkpoint, and on restart the runner appended to the earlier trace log, so the locked analysis refused the matrix; the directory was quarantined, the coordinate re-executed through the orchestrator, and the re-run reproduced the receipted trainable-state digest bitwise before the analysis ran. The retained coordinate reproduces its receipted artifact; wall-clock is accumulated from per-step traces. Load changes the comparability of timing records even though it does not change the matched artifact, as discussed below. All three incidents are recorded in the experiment root.

Latency under a shared node. The R1 records use the same-session standalone pass as their denominator; the original pooled-rule outcomes are retained as protocol records. The node was shared with other users' training for most of the fresh stage, so per-step protected wall time differed by up to $2.2\times$ between arms by time of day: on GSM8K the median protected step was 0.373 s for

base, ORA_{pre}, $K{=}1000$, and never (run on an idle device) but $0.789/0.837$ s for $K{=}1/K{=}10$ (run under foreign load); on SQuAD the base itself ran under load (0.814 s against 0.375 s in Matrix IV), which is why the $K{=}1000$ and never ratios fall below one there. Within one SQuAD $K{=}1$ run the protected step rose from 0.379 to 0.545 s and a publication from 0.569 to 0.662 s when another job started. A publication in this harness clones the protected partition into the reader and computes a verification hash; it costs 0.23–0.65 s per event here (0.6–1.5 protected steps) against 0.21–0.37 s in Matrix IV's quieter session, where $K{=}50$ and $K{=}250$ met the requirement at $1.01\times$. The original R1 rule uses medians pooled over both tasks; per task (GSM8K/SQuAD) the ratios are ORA_{pre} $2.89/2.61$, $K{=}1$ $3.99/1.42$, $K{=}10$ $2.51/1.01$, $K{=}1000$ $1.01/0.73$, and never $1.00/0.73$, so the pooled median conceals task-specific load, and the SQuAD ratios below one cannot establish reduced protected work. Matrix V did not time a file-based or asynchronous publication path; Matrix VIII measures actual concurrent execution (Appendix K.5). The Matrix V records do not bound that cost; the freeze-time prediction that $K{=}10$ would meet the requirement failed under load, and no cadence was re-timed under the locked protocol. Matrix IV provides the quieter-session component accounting: post exceeds the limit and $K{=}50$ and 250 meet it. These outcomes still do not measure concurrent reader availability.

Verdict. H_{D0} holds for every new arm in $20/20$ seed–task pairs against the re-run base, and the re-run base matches the frozen Matrix IV base in $20/20$; H_{D1} IDENTICAL ($20/20$ cells: trainable tensors $20/20$, fused generations and scores $20/20$); H_{D2} MIXED; H_{D3} LEFT FAVOURED; The original pooled R1 rule is met by $K{=}1000$ ($1.00\times$) and never ($1.00\times$) and violated by ORA_{pre} ($2.89\times$), $K{=}1$ ($2.74\times$), and $K{=}10$ ($1.81\times$) (Table 16). The matrix completed on 2026-09-09 UTC (2026-09-10 local time), before the deadline. Reader keep-up ratios (branch time over protected time) are $2.12/1.89/1.97/2.01$ at $K{=}1/10/1000/\text{never}$, and mean staleness is $0.0/4.5/467.4/1671.5$ protected steps (medians over 20 coordinates). The never-published member's terminal scores spread widely across seeds (31.73–41.63 GSM8K accuracy, 48.85–76.98 SQuAD F1; descriptive), which is why its intervals are wide; its contrast against two-stage is MIXED ($-2.048/-7.465\,pp$, CIs $[-5.084, +0.988]/[-15.520, +0.590]$). Its SQuAD scores are bimodal: six seeds at 65.4–77.0 and four at 48.8–58.6. Per-seed Spearman correlation between cadence rank ($1 < 10 < 50 < 250 < 1000 <$never) and terminal quality: mean $-0.52/-0.40$, two-sided sign-flip $p = 0.0039$ on both tasks; descriptive; the rank includes the fixed-initial-state endpoint and does not isolate an effect of publication cadence. Terminal fused means (descriptive, GSM8K/SQuAD): ORA_{pre} $44.42/78.37$, ORA_{post} $43.48/76.55$, $K{=}10$ $43.83/78.33$, $K{=}50$ $43.23/77.86$, $K{=}250$ $43.63/77.30$, $K{=}1000$ $42.76/76.73$, never $37.70/64.42$, two-stage $39.75/71.88$ (Figure 7).

Interpretation of the registered comparisons. The $K{=}1$ identity establishes the post-phase equivalence in the tested implementation. H_{D2} is MIXED, with a GSM8K interval half-width of $1.23\,pp$, so equivalence within $\pm 1\,pp$ is unattainable at this observed precision. H_{D3} establishes the periodic-versus-never comparison, but the never branch trains on s_0 and is evaluated with s_T. Matrix V evaluates the never branch only on s_T. Matrix VII subsequently evaluates both fixed branches on s_0 and s_T, finding that the aligned frozen-base configuration exceeds the tested periodic configuration (Section 4.6 and Table 34). This control establishes sensitivity to the fusion base; the original Matrix V contrast alone does not isolate the effect of access to evolution. The original automatic interpretation text is preserved in the supplement's frozen verdict file. This manuscript reports the statistical outcomes with these identification boundaries; the latency verdicts describe the pooled rule applied to the recorded runs, with no operational conclusion from the shared-node ratios.

E.10 ANYTIME CO-EVOLUTION STUDY

Constant-schedule study. The anytime study is independent of the fresh matrix. Before opening seeds 700–704, its record fixes task, model, partitions, arms, release points, seeds, evaluation plan, predictions, and decision rule. It reuses the Qwen optional-LoRA configuration in Appendix E: two epochs, 1682 protected steps, batch size 8, and the selected ORA_{post} and two-stage branch settings. At protected steps 421, 841, 1262, and 1682, the runner records protected transition state and materializes trainable state for evaluation on the 747-example internal holdout. Because two-stage starts branch training only after step 1682, its fused route is unavailable—not zero—at these release points.

Table 16: Preregistered staleness sweep: locked hypothesis sets (H_D) and descriptive contrasts (D) on fresh seeds 15300–15309, $n = 10$ per task; Matrix IV cells reused by digest. Paired differences in percentage points with two-sided 95% t-intervals and Holm-adjusted exact sign-flip p. D rows are descriptive and carry no verdict; the $\text{ckpt}_1 - \text{ckpt}_{250}$ GSM8K R>L cell prints the tie-inclusive 0.373 as in Table 14 (frozen analyser 0.354).

Contrast	Task	Diff.	Paired 95% CI	Holm L>R	Holm R>L	Verdict
H_{D2}: pre − ckpt_{250}	GSM8K	+0.790	[−0.442, +2.022]	0.091	1.000	MIXED
	SQuAD F1	+1.066	[+0.198, +1.935]	3.91×10^{-3}	1.000	
H_{D3}: ckpt_{250} − never	GSM8K	+5.930	[+4.242, +7.618]	1.95×10^{-3}	1.000	ckpt_{250} favoured
	SQuAD F1	+12.885	[+5.049, +20.722]	1.95×10^{-3}	1.000	
D: pre − post	GSM8K	+0.937	[−0.085, +1.960]	0.031	1.000	descriptive
	SQuAD F1	+1.822	[+1.612, +2.031]	1.95×10^{-3}	1.000	
D: pre − two-stage	GSM8K	+4.672	[+2.905, +6.439]	1.95×10^{-3}	1.000	descriptive
	SQuAD F1	+6.486	[+6.003, +6.969]	1.95×10^{-3}	1.000	
D: ckpt_1 − ckpt_{250}	GSM8K	−0.147	[−1.053, +0.758]	1.000	0.373	descriptive
	SQuAD F1	−0.755	[−1.635, +0.125]	1.000	0.084	
D: ckpt_{10} − ckpt_{250}	GSM8K	+0.201	[−0.984, +1.386]	0.357	1.000	descriptive
	SQuAD F1	+1.028	[+0.209, +1.847]	5.86×10^{-3}	1.000	
D: ckpt_{1000} − ckpt_{250}	GSM8K	−0.870	[−2.015, +0.275]	1.000	0.119	descriptive
	SQuAD F1	−0.573	[−1.919, +0.774]	1.000	0.204	
D: ckpt_{1000} − two-stage	GSM8K	+3.012	[+0.762, +5.262]	9.77×10^{-3}	1.000	descriptive
	SQuAD F1	+4.847	[+3.605, +6.089]	1.95×10^{-3}	1.000	
D: never − two-stage	GSM8K	−2.048	[−5.084, +0.988]	1.000	0.088	descriptive
	SQuAD F1	−7.465	[−15.520, +0.590]	1.000	0.084	

R1 latency (× same-session standalone; GSM8K/SQuAD medians, pooled median in parentheses): pre 2.89/2.61 (2.89, violates 1.05); ckpt_1 3.99/1.42 (2.74, violates); ckpt_{10} 2.51/1.01 (1.81, violates); ckpt_{1000} 1.01/0.73 (1.00) and never 1.00/0.73 (1.00) meet the pooled rule; SQuAD ratios below one reflect a loaded standalone run.

All three prespecified predictions hold. P1 requires ORA_{post} and two-stage to equal matched standalone protected state at every release point while joint does not. P2 requires ORA_{post} and joint, but not two-stage, to expose trained fused routes before the terminal protected point. P3 requires ORA_{post} fused accuracy at release points two and three to exceed terminal two-stage accuracy. The decision rule further requires P1 in 5/5 seeds and a positive paired interval for P3 at point three. At that point, the difference is $+5.17\,pp$, interval $[+4.06, +6.27]$, with 5/5 positive differences; exact one-sided sign-flip $p = 0.03125$ is the minimum attainable at $n = 5$. At point two, the difference is $+8.03\,pp$. The ORA_{post} and two-stage projected routes require no regeneration because bitwise state equality determines them exactly. Joint's terminal path-off route loses $1.18\,pp$ descriptively.

Descriptively, ORA_{post} fused accuracy under this constant schedule peaks at 50% (46.40) and is lower at the terminal point (42.97); standalone projection ($37.54 \to 36.25$) and joint ($45.52 \to 45.19$) also decline over the same interval. The locked comparison is against terminal two-stage at each release point.

Linear-decay schedule and independent extension. The schedule follow-up changes protected and branch schedules from constant to linear decay while retaining model, task, data, arms, release points, and evaluation. Seeds 800–804 were observed first, producing a point-three difference of $+1.954\,pp$, interval $[-0.628, +4.537]$, 4/5 positive differences, and sign-flip $p = 0.0625$. Because this observation triggered a power extension, it is treated as a pilot.

Only seeds 805–819 form the independent extension. At point three, their paired mean is $+1.892\,pp$, interval $[+0.861, +2.923]$, with 11/15 positive differences and exact one-sided sign-flip $p = 0.001434$. ORA_{post} and two-stage each match standalone protected state at all four release points in all 15/15 fresh seeds. These seeds are the inferential source for the schedule follow-up. Pooling seeds 800–819 gives a descriptive mean of $+1.908\,pp$, interval $[+1.036, +2.779]$, and 15/20 positive differences, without ordinary confirmatory status.

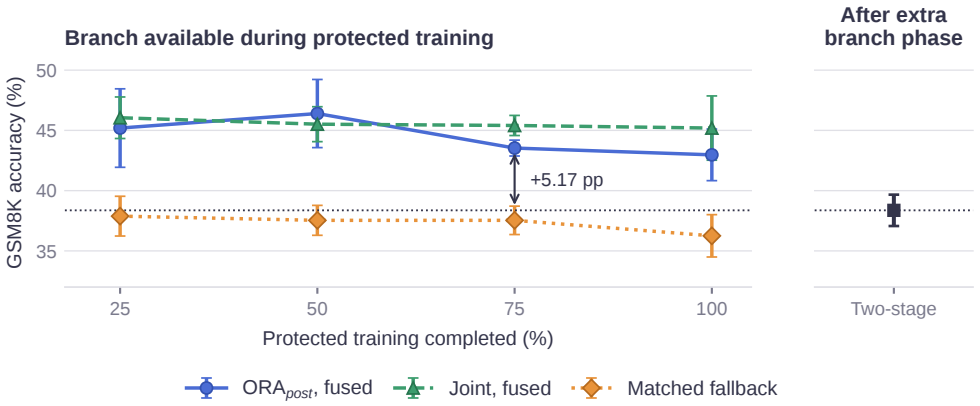

Figure 6: **Online availability under the constant schedule.** Five locked GSM8K seeds. $\mathrm{ORA}_{\mathrm{post}}$ is exact throughout; two-stage has no fused route during protected training. At 75%, $\mathrm{ORA}_{\mathrm{post}}$ exceeds terminal two-stage by $5.17\,pp$ (CI $[+4.06, +6.27]$) after 2524 versus 3364 updates. Error bars are 95% t intervals for each mean across five seeds; the quoted gain uses a separate paired-difference interval. The two-stage endpoint is shown on a separate categorical axis after its extra branch phase.

Corrections and scope. The initial power-extension analysis incorrectly combined the outcome-triggering pilot and fresh extension into an ordinary $n = 20$ confirmatory sample. The corrected analysis separates all three roles. Two earlier constant-schedule measurements were recomputed after correcting a phase label that included cursor position and an availability predicate that excluded joint coupled updates. Neither correction changes the point-three quality comparison. Together, the studies support online availability and a consistent quality direction under two schedules, but neither a schedule-invariant effect size nor a claim that co-evolution alone causes the full difference. Their comparator has no fused route before T by construction; the fair release-point comparator is a branch trained against the checkpoint available at that step, and the preregistered test against it (H_{C3}, Appendix E.8) is MIXED on both tasks, so these studies establish availability over terminal-only access, not an advantage over checkpoint-assisted access.

E.11 EIGHT-BRANCH OPERATIONAL WORKFLOW

The operational study evaluates BERT early exits with QNLI $K \in \{1, 2, 4, 8\}$ and SST-2 $K \in \{1, 8\}$ over seeds 500–509, totaling 260 task–exit–seed runs. The QNLI $K = 8$ primary requires a macro-quality lower bound above $-1\,pp$, worst-tier lower bound above $-2\,pp$, exact protected endpoints, mean $\mathrm{ORA}_{\mathrm{post}}$/two-stage time ratio at most 0.90, one-sided ratio upper bound below 1, and mean saving of at least 900 seconds. Observed macro and worst-tier differences are $-0.177\,pp$ and $-0.544\,pp$, with lower bounds $-0.214\,pp$ and $-0.616\,pp$; mean ratio is 0.9281 with one-sided 95% upper bound 0.9409; and mean saving is 296.39 seconds. The identity, quality, and ratio-uncertainty requirements are met, but the time-ratio and saving thresholds are not. Separately, validation reuse avoids 2198.0 and 829.4 seconds of mean base validation on GSM8K and SQuAD.

E.12 TRAJECTORY-REPLAY EXACT COMPARATOR

For GSM8K and SQuAD seeds 420–439, replay restarts from the same pretrained model and seeded initialization as the selected $\mathrm{ORA}_{\mathrm{post}}$ run, executes the same two-epoch protected recipe, and applies one selected branch update after each protected update. The completed standalone run remains the released fallback; replay does not replace it. Direct recursive comparison covers protected parameters and buffers, full trainable state, base and branch AdamW state, both constant schedulers, all explicit generators, and protected and branch cursors. Every coordinate must match on all nine of these comparisons, which cover protected and branch state together; neither tolerance, a task metric, nor generated-output equality substitutes for state equality.

E.13 REAL-STACK FINITE TRACE

The Qwen intervention trace uses the GSM8K materialized fit+tune tensor, seeds 610–614, batch size 8, and eight protected transitions, resuming after transition 4. Every positive or negative cell pairs a standalone reference with a candidate. Protected snapshots include parameters and persistent buffers, protected AdamW and constant-scheduler state, protected RNG, and data permutation/cursor/generator state. Positive cells must match at every boundary; negative cells pass only when both the first unequal boundary and the complete set of first unequal components match the prespecified intervention.

E.14 FULL-HORIZON AND SCALE CONTRACT TRACES

The uninterrupted positive trace extends this paired comparison across complete locked two-epoch horizons. Qwen2.5-0.5B uses seeds 620–622 on GSM8K (1,682 protected steps per seed) and SQuAD (5,002 per seed), totaling 40,110 recorded boundaries. The scale-contract check uses Qwen2.5-1.5B with rank-32 LoRA on GSM8K and fresh seeds 720–722; each coordinate contains 3,365 recorded boundaries. All comparisons use eager FP32, direct `torch.equal` tensor checks, and the seven protected components enumerated in the locked protocols (parameters, buffers, optimizer, scheduler, protected RNG, data stream, control); the precision/scaler component carries no live state under the fixed eager-FP32 policy recorded in each run manifest. These coordinates test implementation-level exactness, not independent metric samples or unrecorded runtime state.

E.15 MIDPOINT BRANCH-ABANDONMENT LIFECYCLE

The lifecycle study pins Qwen2.5-0.5B on GSM8K for 1,682 protected transitions and abandons the optional branch after transition 841. Standalone, $\mathrm{ORA}_{\mathrm{post}}$, and joint share protected data order and RNG sequence. Before abandonment, standalone and $\mathrm{ORA}_{\mathrm{post}}$ apply the isolated protected transition, followed by the $\mathrm{ORA}_{\mathrm{post}}$ branch-owned transition; joint applies a fused transition with global clipping. Each role is then serialized as a protected-only checkpoint and restored in a fresh runtime for projected-only continuation. Checkpoints contain protected parameters and buffers, projected protected AdamW state, protected scheduler, base RNG, data permutation/generator/epoch/cursor, and protected step counter. They contain no branch parameters, optimizer, scheduler, or RNG.

The outcome-blind V2 lock uses seeds 9000–9002. An initial seed assignment overlapped the previous anytime study; those processes were stopped before the abandonment boundary and before any result or checkpoint was written. The original lock, corrected lock, and deviation record are all retained. The formal gate requires every recorded $\mathrm{ORA}_{\mathrm{post}}$/standalone state component and protected loss to match exactly, exact restoration of that inventory in a fresh runtime, joint divergence at step one and non-identity at both the abandonment and final boundaries, and independent token equality over every example in the locked 747-example holdout.

Table 17: Locked midpoint branch-abandonment lifecycle. "ORA cont." requires exact agreement of the recorded state, protected loss, fresh-runtime restore, and post-abandonment continuation. "Branch absent" means that no branch-owned component is present in the restored checkpoint. Joint deviation is final protected-parameter L2 normalized by standalone displacement from initialization; accuracies are projected GSM8K percentages for standalone/ORA/joint.

Seed	ORA cont.	Branch absent	Joint first div.	Joint norm. L2	Acc. S/O/J	S=O tokens
9000	exact	yes	step 1	0.3916	35.74/35.74/35.88	747/747
9001	exact	yes	step 1	0.3982	36.14/36.14/39.09	747/747
9002	exact	yes	step 1	0.3895	39.09/39.09/39.76	747/747

All three coordinates pass the locked lifecycle gate. At abandonment, joint differs from standalone in 13 protected parameter tensors and 26 protected optimizer-moment entries, while scheduler, base RNG, data cursor, and control still match. This localizes the tested mismatch to fused writes. Its mean absolute continuation-loss gap is 0.01315. Because joint projected accuracy is higher in all three coordinates, neither non-identity nor the lifecycle pass implies universal task-metric harm.

E.16 CROSS-ARCHITECTURE BERT EARLY-EXIT LIFECYCLE

A separately locked lifecycle tests whether the abandonment construction transfers from optional LoRA to early exits. It uses the pinned eager-FP32 BERT-base/SST-2 stack, all $67{,}349$ training examples, batch size 32, five epochs, and fresh seeds 12001–12003. Standalone, ORA_{post}, and PABEE-style joint training share one complete initialization per seed and the same deterministic data order. At protected update $5{,}262$ of $10{,}525$, each role saves protected parameters and buffers, protected AdamW state, scheduler, protected RNG, cursor, and control state. Joint's AdamW is projected onto protected parameters so that historically coupled moments remain, while every optional classifier and optimizer entry is excluded. Fresh model/runtime instances load the checkpoints, physically remove all ramps, and continue with protected-only transitions.

Table 18: Fresh BERT-base/SST-2 branch-abandonment lifecycle. "ORA exact" covers 50 recorded state boundaries, all $10{,}525$ protected-loss values, protected-only restore, and final logits/predictions. Accuracies are standalone/ORA/joint percentages on all 872 validation examples.

Seed	ORA exact	Branch absent	Joint first div.	Joint non-exact final	Acc. S/O/J
12001	yes	yes	step 1	yes	92.66/92.66/91.28
12002	yes	yes	step 1	yes	93.23/93.23/91.97
12003	yes	yes	step 1	yes	92.89/92.89/91.17

All three formal coordinates pass both locked gates. In aggregate, ORA_{post} matches standalone at $150/150$ recorded boundaries and all $31{,}575$ protected-loss updates; final logits and predictions match on all $2{,}616$ seed–example coordinates. Joint first diverges after update one in every seed and remains non-exact after branch removal. Its validation accuracy is $1.45 \pm 0.24\,pp$ lower than standalone (mean $\pm$ sample standard deviation), but quality harm was neither required nor predicted. The result establishes lifecycle transfer for this pinned BERT/SST-2 implementation, not arbitrary hardware exactness, universal early-exit behavior, or production value.

E.17 CONTROLLED EXACTNESS–QUALITY RELAXATION

The controlled relaxation reuses the lifecycle experiment's Qwen2.5-0.5B GSM8K eager-FP32 stack, $1{,}682$ protected transitions, and complete 747-example internal holdout. At each transition, projected and fused protected-parameter gradients are evaluated from the same pre-update state and explicit RNG seed, then mixed as $g_\alpha = (1-\alpha)g_{\text{proj}} + \alpha g_{\text{fused}}$ for $\alpha \in \{0, .1, .25, .5, 1\}$. Protected AdamW, clipping, scheduler, data stream, and RNG remain separate from branch-owned state. After the protected step, every cell applies the isolated branch-only transition used by ORA_{post}. Thus $\alpha = 0$ is the ordinary exact construction; positive α cells are controlled non-exact probes rather than ORA_{post} variants. The $\alpha = 1$ endpoint also differs from the ordinary joint and PCGrad comparators.

Before execution, the sequential protocol fixes exploratory seed 9100, confirmation seeds 9101–9103, the complete grid, and all gates. Exploration returns PROMOTE FRESH CONFIRMATION and serves only as selection evidence. Confirmation repeats all five values without selecting an α. Its paper-level gate requires complete evaluation, exact $\alpha = 0$, divergence for every positive cell, nondecreasing mean normalized deviation, an interior mean non-dominated point, and at least $1\,pp$ fused gain for the mean-best α in every seed.

Table 19: Per-seed controlled relaxation. "First div." is the first unequal protected boundary; scores use all 747 holdout examples. Every $\alpha = 0$ row matches standalone throughout.

Seed	α	First div.	Norm. L2	Projected	Fused
9101	0	none	0.00000	37.22	42.57
	0.1	step 1	0.06646	36.41	43.37
	0.25	step 2	0.15361	36.55	42.84
	0.5	step 2	0.30909	37.08	45.11
	1	step 2	0.68530	30.92	46.18
9102	0	none	0.00000	37.48	44.58
	0.1	step 1	0.06471	38.55	45.78
	0.25	step 2	0.15258	37.22	44.98
	0.5	step 2	0.30445	38.42	46.99
	1	step 2	0.68060	31.73	44.04
9103	0	none	0.00000	36.55	39.63
	0.1	step 1	0.07217	35.21	40.56
	0.25	step 2	0.16518	36.55	41.77
	0.5	step 2	0.31564	38.42	43.78
	1	step 2	0.68649	30.92	43.78

Under the prespecified confirmation rule, mean normalized deviation is strictly ordered with α, increasing from zero at $\alpha = 0$ to 0.6841 at $\alpha = 1$. Mean fused accuracy peaks at 45.29 for $\alpha = .5$, and the non-dominated mean points are $\alpha \in \{0, .1, .5\}$. At $\alpha = .5$, accuracy improves by $+3.03\,pp$ over exact $\alpha = 0$, with seed-wise gains $+2.54, +2.41, +4.15\,pp$. This establishes a controlled quality–state-deviation frontier for the pinned stack. No positive-α cell is recoverable, and the sweep supplies neither a universal utility function nor an unconstrained joint/PCGrad comparison.

Post-outcome behavioral analysis. Without retraining or reevaluation, each positive-α projected output is compared with the same-seed $\alpha = 0$ output over all 747 holdout examples. The analysis measures complete token sequence disagreement, extracted-answer disagreement, normalized token edit distance, and correctness flips. Counts below pool the $3 \times 747 = 2{,}241$ paired examples; rates and edit distances average over seeds.

Table 20: Behavioral consequences of protected-state relaxation. Accuracy differences are projected-route changes from same-seed $\alpha = 0$; flip columns are pooled counts. This post-outcome analysis is descriptive.

α	Norm. L2	Token disagree	Answer disagree	Mean edit	C→W	W→C	Δ acc.
.1	.068	28.34%	16.38%	8.88%	58	50	$-0.36\,pp$
.25	.157	53.32%	31.19%	17.40%	107	100	$-0.31\,pp$
.5	.310	75.95%	45.16%	27.70%	139	159	$+0.89\,pp$
1	.684	99.96%	73.76%	60.12%	383	251	$-5.89\,pp$

Normalized L2 has descriptive Spearman correlation 0.918 with token disagreement and 0.916 with both answer disagreement and mean token edit distance across the twelve positive-α coordinates. The coordinates share seeds and α levels, task harm is non-monotonic, and no risk threshold was prespecified. The supported conclusion is therefore narrower: aggregate accuracy can conceal extensive sample-level behavioral replacement, while L2 is not calibrated as a universal behavioral-risk metric.

E.17.1 BERT/QNLI CONTROLLED FRONTIER

This independent replication prospectively fixes BERT-base-uncased on QNLI, all $104{,}743$ training examples, eager FP32, batch size 32, five epochs (16,370 updates), AdamW at 2×10^{-5}, no warmup, and gradient clipping at 1.0. The $5{,}463$ validation examples are deterministically label-stratified into disjoint calibration and evaluation partitions. Exploration uses only seed 13000 and calibration labels; it selects the smallest qualifying interior point, $\alpha = .1$. Confirmation then evaluates the full $\{0, .1, .25, .5, 1\}$ grid on the held-out partition with fresh seeds 13001–13003. The primary route,

fixed beforehand from historical QNLI runs, is patience 2. Here $\alpha = 0$ invokes the canonical ORA_{post} transition; positive α interpolates final-path and depth-weighted all-exit objective gradients before the protected step and is deliberately non-recoverable. The $\alpha = 1$ endpoint retains separate state ownership and is not ordinary PABEE joint training, which is run separately as unconstrained quality context.

Table 21: BERT/QNLI controlled frontier on the prospectively held-out evaluation partition (2,731 examples). Accuracy and mean exit layer use the fixed patience-2 route. "Exact" denotes equality of the recorded protected trajectory to the paired standalone reference; all positive α rows first diverge at step one. Normalized L2 is a within-stack coordinate only.

Seed	α	Norm. L2	First div.	Acc. (%)	Mean layer	Final acc. (%)	Exact
13001	0	0.000	none	75.83	3.924	91.47	yes
	.1	0.753	step 1	80.81	3.953	91.18	no
	.25	0.837	step 1	82.53	3.964	90.66	no
	.5	0.945	step 1	82.79	3.952	90.26	no
	1	1.175	step 1	83.01	3.956	89.31	no
13002	0	0.000	none	76.20	3.963	91.18	yes
	.1	0.704	step 1	81.25	3.992	91.25	no
	.25	0.831	step 1	82.39	3.971	91.36	no
	.5	1.005	step 1	83.30	3.978	90.66	no
	1	1.213	step 1	83.96	3.970	89.60	no
13003	0	0.000	none	74.88	3.899	90.92	yes
	.1	0.822	step 1	80.52	3.943	91.29	no
	.25	0.920	step 1	82.13	3.950	90.66	no
	.5	1.009	step 1	83.01	3.965	90.26	no
	1	1.202	step 1	83.78	3.970	89.34	no
Mean	0	0.000	none	75.64	3.929	91.19	3/3
	.1	0.760	step 1	80.86	3.963	91.24	0/3
	.25	0.863	step 1	82.35	3.962	90.89	0/3
	.5	0.986	step 1	83.03	3.965	90.39	0/3
	1	1.197	step 1	83.58	3.965	89.42	0/3

Under the prospectively fixed decision rule, all ten confirmation criteria are met. At the preselected $\alpha = .1$, patience-2 gains over exact $\alpha = 0$ are $+4.980, +5.053, +5.639\,pp$; the mean is $+5.224 \pm 0.361\,pp$ and the mean exit-layer change is $+0.034 \pm 0.009$, where both $\pm$ values are sample standard deviations across the three confirmation seeds. Mean normalized L2 is nondecreasing from 0 to 1.197, every positive cell diverges, and $\alpha = .1$ remains non-dominated. Ordinary joint yields patience-2 accuracies $83.78, 83.49, 83.63\%$ at mean layers $4.018, 3.988, 4.017$ (final accuracies $89.89, 89.56, 89.97\%$) and normalized deviations $1.307, 1.296, 1.317$; every seed diverges at step one and is non-exact. It supplies context, not a controlled-family endpoint. At selected $\alpha = .1$, final predictions flip on $76, 76, 90$ examples and patience-2 predictions on $316, 320, 344$ examples relative to same-seed $\alpha = 0$ (out of 2,731 per seed). Thus the boundary is observed in two tested model/optional-path pairs. It does not establish recoverability for positive α, universal quality monotonicity, cross-stack calibration of normalized L2, fixed-FLOP matching, arbitrary task/scale transfer, or operational value.

E.18 Exact-fallback validation reuse

The release workflow uses seeds 420–424 and complete internal projected suites of 747 GSM8K and 5002 SQuAD examples. It fixes the projected evaluator, protected-route implementation, configuration, tokenizer, environment, and projected-path suite while allowing optional-extension code and weights to differ. Standalone is the validated reference, ORA_{post} the exact candidate, and joint training the non-exact control. The identity predicate compares serialized protected parameters and persistent buffers; its timing includes loading both trainable states and recursively comparing those fields. This predicate does not establish evaluator premises, disabled-route inertness, or complete trace equality. The LoRA wrapper disables the path by returning the base linear output before optional residual computation. Full base and joint validation includes model/tokenizer loading, greedy generation over

every example, and equality to all recorded token lists. Hence $20/20$ replay paths denotes full-suite paths, not examples. Net avoided time subtracts ORA_{post} identity checking from base validation wall time. Locked gates require ORA_{post} identity and joint non-identity in all groups, exact base/joint token replay, identity time at most 1% of base validation in every group, and at least 600 seconds mean net avoided time per task. Timings exclude training, quality, storage, distribution, and serving costs.

E.19 Retained-clean-checkpoint boundary accounting

This boundary has a concrete memory–latency trade-off. Standalone training alone takes $641.62/1977.90$ s on GSM8K/SQuAD, so ORA_{post}'s $2025.91/7254.24$ s is $3.16/3.67\times$ standalone and $1.91/2.27\times$ joint. Beyond an existing clean state, ORA_{post} adds a 1.98 MiB branch payload and clean plus joint adds 58.87 MiB, a difference of 56.89 MiB. Dual residency uses 3.69 rather than 1.84 GiB and switches in 0.007–0.008 ms; single residency stays at 1.84 GiB but reloads in 12.9–16.0 ms.

This post-outcome descriptive analysis reuses standalone, ORA_{post}, and joint artifacts for GSM8K and SQuAD seeds 420–424. M1 keeps one ORA_{post} instance; M2 keeps clean and joint instances resident; M3 keeps one model plus clean and joint trainable states in CPU memory, copying the selected state on each route change. Five independent processes measure model-and-state loading per mode and coordinate. Separate processes measure resident memory and 100 switches after 10 setup alternations. Because OS page cache is uncontrolled, the former is process-start load rather than cold load. Correctness directly compares named tensors and generated token-ID lists for one fixed holdout prompt per task and direction. The analysis defines neither a new hypothesis family nor a stakeholder-weighted utility score; M2 latency cannot be combined with M3 memory.

F Closure Results

Table 22: Exactness–quality trade-off on fresh internal splits ($n = 20$). Path-off Δ is relative to standalone scores $36.96/55.50$; a negative value is a RecoveryGap. Joint/PCGrad fused means are descriptive.

	GSM8K acc.			SQuAD F1		
Method	Fused	Path-off (Δ)	Exact	Fused	Path-off (Δ)	Exact
Joint	**46.06**	34.63 (-2.33)	0/20	**78.31**	44.05 (-11.45)	0/20
PCGrad	45.29	36.77 (-0.19)	0/20	78.26	54.95 (-0.55)	0/20
ORA_{post}	43.04	36.96 (± 0.00)	20/20	77.01	55.50 (± 0.00)	20/20
Two-stage quality	39.55	36.96 (± 0.00)	20/20	67.11	55.50 (± 0.00)	20/20

Table 23: Direct replay comparator ($n = 20$ per task). Exact counts require all nine mandatory state comparisons (protected and branch model, optimizer, scheduler, generator, and cursor state). Offline total includes standalone and replay passes; wall-time direction differs by task.

Task	Exact	Replay pass (s)	Offline total (s)	ORA_{post} (s)	Offline/ORA_{post}
GSM8K	20/20	1667.86	2309.48	2025.91	1.1416
SQuAD	20/20	4959.61	6937.51	7254.24	0.9631

Table 24: Retained-clean-checkpoint boundary. G/S denotes GSM8K/SQuAD; training is candidate-only wall time. Delta uses logical tensor payload, and switch values average five coordinate medians.

Mode	Fused G/S	Train G/S (s)	Delta (MiB)	GPU (GiB)	Switch G/S (ms)
M1 ORA_{post}	43.04/77.01	2025.91/7254.24	1.98	1.84	0.124/0.092
M2 two resident	46.06/78.31	1062.84/3193.74	58.87	3.69	0.007/0.008
M3 state reload	46.06/78.31	1062.84/3193.74	58.87	1.84	15.978/12.908

All $150/150$ process-start measurements and $30/30$ memory/switch coordinates complete; named state and token lists pass in all $60/60$ route checks. Mean process-start coordinate medians for

M1/M2/M3 are $3.415/4.880/3.544$ seconds on GSM8K and $2.642/4.106/2.663$ seconds on SQuAD. The immutable snapshot is 0.93 GiB. Counting it once, greenfield ORA/two-checkpoint bundles are $0.99/1.04$ GiB; duplicating immutable files would make the latter 1.97 GiB but is not required. M3 CPU RSS is $1.07/0.93$ GiB on GSM8K/SQuAD. Device queues share host I/O and finish at different times, so absolute process-start values are not compared across tasks; mode comparisons are paired within task–seed–device. These measurements establish a Pareto boundary, not an operational winner.

Table 25: Real-stack Qwen trace. Positive cells require complete recorded agreement; negative cells require the prespecified first boundary and components in all five seeds.

Cell	Expected first boundary	Expected components	Pass	Total
Isolated FP32	none	none	5	5
Isolated BF16	none	none	5	5
Complete resume	none	none	5	5
Global clipping	step 1, after optional	model, optimizer	5	5
Protected optimizer reuse	step 1, after optional	model, optimizer	5	5
RNG leakage	step 1, after optional	protected RNG	5	5
Data-cursor leakage	step 1, after optional	data stream	5	5
Incomplete resume	resume after step 4	data stream	5	5

Table 26: Exact-fallback validation reuse ($n = 5$ per task). Identity/base is the worst coordinate ratio; avoided time excludes identity loading and comparison.

Task	ORA_{post} identity	Joint non-id.	Token replay	Identity/base	Avoided (s)
GSM8K	5/5	5/5	10/10	0.0166%	2198.00
SQuAD	5/5	5/5	10/10	0.0248%	829.38

G Locked Standard-Split Secondary

Table 27: Locked standard-split secondary ($n = 20$). "Proj. match" counts identical projected output sequences, not a new endpoint test; no practical margin was defined.

Task	ORA_{post}	Two-stage	Diff.	Paired 95% CI	Holm p	Proj. match
GSM8K test acc.	31.83	29.82	$+2.013$	$[+1.350, +2.675]$	2.86×10^{-6}	20/20
SQuAD val. F1	82.47	71.73	$+10.737$	$[+9.878, +11.596]$	1.91×10^{-6}	20/20

H Complete Contract Interventions

Table 28: **State ownership predicts the first divergence.** Negative controls pass only at the declared first component; positives require equality at every checked boundary. Qwen counts cover five seeds per cell and eight protected steps. Synthetic controls test additional channels.

Intervention	Predicted first effect	Evidence / outcome
Isolated FP32, BF16, restore	none	Qwen $15/15$ exact
Shared clipping or optimizer	model / optimizer	Qwen $10/10$ predicted
Protected RNG reuse	protected RNG	Qwen $5/5$ predicted
Cursor reuse or partial restore	data stream	Qwen $10/10$ predicted
Buffer, scheduler, overflow	buffer / scheduler / control	synthetic: all predicted

Each cell pairs reference and candidate roles over five seeds in a linear–BatchNorm network with input dimension 8, hidden dimension 12, four classes, and a separate linear branch. Runs use 48 synthetic examples, batch size 8, and an eight-step trace. Positive cases must preserve every recorded component; negative cases must diverge first in the declared channel for every seed. Later divergence cannot rescue a failed prediction. A two-rank DDP secondary repeats the BF16-autocast separated-state positive cell and the RNG-leakage negative cell on both ranks with seeds 300–304 (40

Table 29: Controlled contract matrix. Negative cells pass only when the declared channel diverges first.

Cell	Paired intervention / expected observation	Result
Separated FP32 state	branch owns all transition state; protected trace remains equal	exact
Separated BF16 autocast state	same ownership under tested mixed-precision policy	exact
RNG leakage	branch consumes protected random stream; RNG diverges first	as predicted
Global clipping	branch gradients enter shared norm; protected optimizer input diverges	as predicted
Shared optimizer step	branch advances shared moments; optimizer state diverges first	as predicted
Data-cursor leakage	branch loading advances protected cursor	as predicted
Buffer contamination	branch forward changes persistent running statistics	as predicted
Shared overflow decision	branch overflow changes protected step/control state	as predicted
Fused-loss scheduler	branch-dependent signal changes protected scheduler	as predicted
Complete resume	all next-transition state is restored	exact
Incomplete resume	required transition state is omitted	as predicted

rank–seed–role executions); every positive execution preserves the recorded trace and every negative execution first diverges in the protected RNG channel. This suite isolates the transition contract rather than full BERT or Qwen training.

I Statistical Scope and Reporting Rules

All task comparisons pair explicit seeds; paired 95% intervals are two-sided Student-t intervals over seed-level differences. The canonical and fresh two-task hypothesis sets use exact one-sided sign-flip tests with Holm correction and enumerate all 2^{20} assignments. The fresh primary also requires mean advantage above $1\,pp$; standard-split results have no practical margin. Joint/PCGrad means and projected recovery gaps remain descriptive.

The constant-schedule anytime comparison uses its prespecified $n = 5$ paired interval and exact sign-flip test. In the outcome-triggered decay extension, seeds 800–804 are pilot and 805–819 are the independent $n = 15$ inferential sample, tested over all 2^{15} sign assignments. Pooled $n = 20$ is descriptive only.

The independent BERT/QNLI frontier uses calibration seed 13000 only to select $\alpha = .1$, then applies fixed per-seed $1\,pp$ gain, mean $1\,pp$ gain, 0.10-layer compute, non-dominance, exactness, and completeness gates to held-out seeds 13001–13003. These are prospective practical gates over three seeds, not a post-hoc significance test.

The preregistered family matrix applies the fresh rules with $n = 20$ (2^{20} enumeration), Holm over the two tasks within each hypothesis set (no correction spans its four sets), and a symmetric set tested in both directions; the provenance matrix pairs $n = 10$ GSM8K and $n = 6$ SQuAD seeds with Holm over its two trace metrics within a task, so the smallest attainable adjusted values are 2^{-9} and 2^{-5}; the full-model matrix applies the fresh rules with $n = 10$ per task (2^{10} enumeration) and Holm over the two tasks; the checkpoint matrix applies the same $n = 10$ rules to symmetric hypothesis sets, and its latency requirement is a median ratio against a locked limit rather than a test. QNLI, SST-2, and GSM8K use accuracy; SQuAD uses token-overlap F1. Scores use a 0–100 scale and differences use percentage points. Endpoint equality is reported as seed counts, never inferred from zero metric difference.

J CLAIM BOUNDARIES

The ownership construction preserves the matched protected transition under its explicit assumptions. Canonical BERT endpoints, Qwen full-horizon recorded-state traces, and the two lifecycle studies provide distinct implementation evidence at their stated horizons. The provenance study independently tests protected-state equality and retained canary exposure. Quality is conditional on the protected setting and batch protocol: both live members in Matrix II and periodic observation at $K{=}250$ in Matrix IV exceed their selected terminal baselines, while the full-model comparison reverses. VI finds practical pre/post equivalence under independent branch batches and lower quality for both members. Its interaction analysis supports a changed phase comparison on SQuAD while leaving GSM8K unresolved. VII finds that matching the fixed-base branch to s_0 rescues quality and exceeds the tested $K250@s_T$ configuration. These controls delimit the earlier batch and fusion explanations without identifying a universal mechanism. Live/checkpoint comparisons are MIXED at the recorded precision. Checkpoint training reduces accounted protected-path work but its reader fails the matched-hardware keep-up check. VIII measures actual writer progress, branch availability, and completed-branch quality. Exact historical replay then supports a post-outcome observation-sequence intervention on its six retained trajectories: replacing the sequence improves quality at a fixed terminal base (Appendix K.7). This descriptive offline result is followed by two complete fresh-seed confirmations (Appendix L). Both confirm $P - H$ on both tasks; neither establishes joint noninferiority to clean plus joint. The live timing audit remains descriptive (Appendix L.1). The preregistered ten-seed independent-order replay confirms batch dependence and a SQuAD-only residual advantage (Appendix N). The prospective full-model P test reverses its matched terminal-only comparison on both tasks (Appendix O). These results support audited recovery and conditional observation effects, with original protocols, deviations, and machine verdicts retained for verification.

The evidence does not establish post-hoc recovery, a causal account of the full quality gap or of its reversal under full-model protection, equivalence of live and checkpoint observation or a cadence-optimal K, universal schedule/scale transfer or exactness, quality dominance, uniform speed, or production value. Positive-α probes are non-recoverable; their within-stack normalized L2 values cannot be calibrated or compared across stacks, mean exit layer is not fixed FLOPs, and $\alpha = 1$ is not ordinary joint. The 1.5B trace tests implementation transfer only; historical quality pilots remain exploratory. Standard-split results measure sensitivity, the multi-branch practical gate fails, and the composition/retention propositions are conditional rather than empirical utility results.

K BATCH AND FUSION CONTROLS

K.1 HISTORICAL PHASE AND CHECKPOINT COMPARISONS

The following table retains the original II/IV/V comparisons under their same-batch protocol. VI and VII supply the additional controls needed to interpret them. The same one-block members, compared

Table 30: **Phase and checkpoint comparisons under the same-batch protocol.** Paired differences in pp and 95% intervals. Pre–post (II, $n = 20$) passes the prespecified rule. The other comparisons (IV/V, $n = 10$) are MIXED: neither a task-general ordering nor equivalence within $\pm 1\,pp$ is established. In pre–checkpoint, the GSM8K half-width $1.23\,pp$ prevents equivalence at the observed precision.

Contrast	GSM8K accuracy	SQuAD F1
$\mathrm{ORA}_{\mathrm{pre}} - \mathrm{ORA}_{\mathrm{post}}$ (II)	$+1.452\ [0.973, 1.932]$	$+1.687\ [1.468, 1.906]$
$\mathrm{ORA}_{\mathrm{post}} - \mathrm{ORA}_{\mathrm{ckpt250}}$ (IV)	$-0.147\ [-1.053, 0.758]$	$-0.755\ [-1.635, 0.125]$
$\mathrm{ORA}_{\mathrm{pre}} - \mathrm{ORA}_{\mathrm{ckpt250}}$ (V)	$+0.790\ [-0.442, 2.022]$	$+1.066\ [0.198, 1.935]$

with selected two-stage training, and the earlier full-model study appear below; the full-model row changes partition, protected learning rate, and branch settings together.

K.2 INDEPENDENT BRANCH STREAMS AND THEIR INTERACTION

VI retains seeds 15300–15309 and the frozen IV/V scientific configuration. Its two new arms draw the branch batch from an independent permutation; the protected stream is unchanged. Smoke and full reference gates precede 40 fresh coordinates, all completed before the deadline. Every new

Table 31: **Observation effects across training settings.** Fused-score differences from two-stage, in pp with paired 95% intervals. II/IV use the selected one-block comparator; III uses its full-model comparator. II: 20 seeds/task; III/IV: 10. III changes partition, protected learning rate, and branch settings together.

Protected setting / member	GSM8K accuracy	SQuAD F1
One block: ORA_{pre} (II)	$+4.65\ [3.66, 5.64]$	$+6.77\ [6.38, 7.17]$
One block: ORA_{post} (II)	$+3.20\ [2.27, 4.13]$	$+5.09\ [4.68, 5.50]$
One block: ORA_{ckpt}, K=250 (IV)	$+3.88\ [1.94, 5.83]$	$+5.42\ [4.56, 6.28]$
Full model: ORA_{post} (III)	$-5.96\ [-7.25, -4.67]$	$-4.15\ [-4.85, -3.44]$

coordinate matches the frozen standalone initial protected hash, protected loss digests, and endpoint components. Batch digests show no same-step protected/branch batch identity. These loss-trace and endpoint checks do not constitute a complete all-component trace at every update.

The locked VI analysis reports symmetric task comparisons, paired 95% t intervals, Holm correction within each two-task hypothesis set, and a 1 *pp* practical margin. Both pre/post intervals lie within $(-1, 1)$ *pp*. The separately frozen interaction addendum specifies a two-sided sign-flip test with ties included at 10^{-12} *pp* tolerance and Holm across two tasks. It was frozen before VI outcomes were opened, conditional on known IV/V coupled-arm results; it neither replaces the original VI rules nor supplies an independent replication. Its interpretation avoids exclusive attribution from a non-significant contrast.

Table 32: **Independent branch batches remove the phase ordering and lower both members' quality.** VI, 10 paired seeds per task; differences in pp with 95% intervals. "Own" denotes the independent branch stream. The first row meets practical equivalence; both other rows favor the same-batch comparator under the locked rules.

Contrast	GSM8K accuracy	SQuAD F1
$\text{pre}_{\text{own}} - \text{post}_{\text{own}}$	$-0.214\ [-0.756, 0.328]$	$-0.027\ [-0.158, 0.105]$
$\text{post}_{\text{own}} - \text{post}$	$-2.262\ [-3.136, -1.389]$	$-2.474\ [-3.121, -1.827]$
$\text{pre}_{\text{own}} - \text{pre}$	$-3.414\ [-4.825, -2.002]$	$-4.322\ [-4.839, -3.806]$

Table 33: **Phase–batch interaction.** $(\text{pre}_{\text{own}} - \text{post}_{\text{own}}) - (\text{pre} - \text{post})$, $n = 10$ paired seeds. A directional finding requires adjusted $p < 0.05$, a CI excluding zero, and absolute mean above 1 *pp*.

Task	Mean (pp)	95% CI	Holm p	Verdict
GSM8K	-1.151	$[-2.328, +0.025]$	0.0586	unresolved
SQuAD	-1.848	$[-2.085, -1.612]$	0.0039	negative

K.3 Fixed-branch evaluation intervention

VII reuses all ten IV/V seed identities and branch weights, with no new training, selection, or tuning. For each task it evaluates never and K=250 on s_0 and reuses their bound s_T evaluations. Two additional branch-disabled s_0 evaluations are shared only after initial-state hashes agree across sources. Smoke and complete token-level reference reproduction precede the 42 formal cells. All cells complete before the locked deadline and before analysis. This is a prospective evaluation follow-up on known training outcomes.

Define N_0, N_T as never evaluated on s_0, s_T, and C_0, C_T as K=250 evaluated on those bases. $\text{H}_{F1} = N_0 - N_T$ holds the never branch fixed. $\text{H}_{F2} = (C_T - N_T) - (C_0 - N_0)$ tests the policy–base interaction. $\text{H}_{F3} = C_T - N_0$ compares configurations with different final bases and training policies. All use paired two-sided 95% t intervals and exact two-sided sign-flip tests over 10 seeds, including ties at 10^{-12} *pp* tolerance; Holm correction spans all six tests. The directional threshold is adjusted $p < 0.05$, CI excluding zero, and absolute mean above 1 *pp*. The control establishes evaluation-base sensitivity; it does not estimate a mediated fraction or a universal cadence effect.

Table 34: **A matched frozen-base branch exceeds the tested periodic configuration.** VII, means over 10 retained seeds, one protected block. Evaluation holds branch weights fixed. Accuracy/F1 are on the 0–100 scale. The initial base without a branch scores $3.21/37.42$.

Branch policy	Evaluation base	GSM8K	SQuAD
never	s_0	50.13	80.59
never	s_T	37.70	64.42
$K{=}250$	s_0	3.43	19.60
$K{=}250$	s_T	43.63	77.30

Table 35: **Complete fixed-branch contrasts.** VII, differences in pp, 10 paired seeds. All six satisfy their directional rules.

Contrast	Task	Mean	95% CI	Holm p
H_{F1}	GSM8K	$+12.436$	$[+10.184, +14.689]$	0.0117
H_{F1}	SQuAD	$+16.176$	$[+8.417, +23.936]$	0.0117
H_{F2}	GSM8K	$+52.637$	$[+50.587, +54.687]$	0.0117
H_{F2}	SQuAD	$+73.883$	$[+65.349, +82.416]$	0.0117
H_{F3}	GSM8K	-6.506	$[-7.723, -5.289]$	0.0117
H_{F3}	SQuAD	-3.291	$[-4.174, -2.408]$	0.0117

K.4 Evidence binding and execution boundary

The interaction receipt binds the byte-level SHA-256 hashes of the final VI analysis file and completion receipt. VI also stores an internal hash of its canonical JSON payload with its own hash field omitted; this is a different digest scope. Manuscript verification checks each scope separately. VII's formal receipt binds all 42 cell receipts, and its completion receipt binds the final analysis file. The manuscript source ledger records all three completed analyses.

The real concurrent study (VIII) is a separate frozen protocol: three seeds per task, four workflows, and alternating roles and workflow order. Seeds 15300 and 15301 use sequential task groups on a quiet GPU pair; seed 15302 uses concurrent task groups on disjoint pairs under the execution amendment. It measures branch availability at the writer endpoint and after draining, then evaluates all retained artifacts outside training timing windows. All $24/24$ workflows and 24 artifact evaluations completed, and the final analysis binds all 48 receipts. Component accounting from IV and loaded timing from V remain distinct evidence. Complete VIII comparisons and per-seed execution conditions are in Appendix K.5.

K.5 Actual concurrent learning: capability and cost

VIII completes 24 workflows and 24 artifact evaluations across two tasks, three seeds, and four workflows. All protected roles pass their reference checks: protected loss digests at every update and named protected tensors and persistent buffers at the endpoint. These checks have a different scope from the complete state traces in the lifecycle studies. Async readers observe immutable publications every 250 writer steps and commit branch weights every 50 reader steps; both retained branch endpoints are fused with the final writer state for evaluation.

The async reader makes a trained branch available before the writer finishes in all six runs. Committed cutoff updates are $800, 800, 700$ of 1682 on GSM8K and $2350, 2400, 2400$ of 5002 on SQuAD. Median drain lags are 670.7 and 1984.3 seconds. Completing all scheduled reader updates does not recover the serial-post quality in this study. Update count alone therefore cannot explain the observed gap. The fixed-trajectory intervention in Appendix K.7 tests the effect of the observed version sequence at the same update budget and terminal fusion base.

These are descriptive comparisons of three retained seeds per task. The execution amendment can affect shared CPU/storage contention and the realized publication sequence; per-seed results retain this distinction. All workflow receipts report no detected foreign GPU load during training. A later evaluation-load incident is recorded separately from these training windows. Peak reserved memory is recorded per worker role; summing role-specific peaks would not measure simultaneous memory use.

Table 36: **All retained seeds in the fusion control.** Accuracy/F1 on a 0–100 scale. Branch-disabled s_0 is shared across seeds: GSM8K 3.213, SQuAD 37.418.

Task	Seed	N_0	N_T	C_0	C_T
GSM8K	15300	47.925	35.475	2.945	42.303
GSM8K	15301	48.327	40.027	5.890	44.311
GSM8K	15302	50.602	36.680	4.552	42.972
GSM8K	15303	50.870	31.727	4.552	42.972
GSM8K	15304	51.004	41.633	2.677	45.515
GSM8K	15305	49.799	39.090	1.473	42.035
GSM8K	15306	49.130	34.672	1.740	42.436
GSM8K	15307	50.602	40.295	1.740	46.319
GSM8K	15308	51.807	40.295	4.685	45.515
GSM8K	15309	51.272	37.082	4.016	41.901
SQuAD	15300	80.486	73.267	19.234	78.052
SQuAD	15301	80.760	76.980	13.069	77.500
SQuAD	15302	80.289	70.739	18.256	77.563
SQuAD	15303	81.019	58.597	12.925	78.226
SQuAD	15304	80.199	49.031	20.339	78.616
SQuAD	15305	80.935	65.413	29.018	78.376
SQuAD	15306	80.787	72.990	14.689	77.006
SQuAD	15307	79.813	48.849	27.872	76.543
SQuAD	15308	80.543	53.206	12.740	74.577
SQuAD	15309	81.105	75.099	27.815	76.565

Table 37: **Actual concurrent workflows (VIII).** Protected finish time relative to the matched standalone run and summed GPU worker occupancy are medians over three seeds. Quality is mean accuracy/F1 in percentage points. Async quality is reported at writer cutoff and after draining; other branch workflows report terminal quality. Occupancy sums worker lifetimes, including startup and waiting; it measures neither kernel-active time nor energy.

Task	Workflow	GPUs	Protected ratio	Occupancy (s)	Quality
GSM8K	Standalone	1	1.000	636.1	—
GSM8K	Serial post	1	3.087	1940.9	43.11
GSM8K	Async reader	2	1.014	1952.3	38.33/38.20
GSM8K	Clean + joint	2	1.009	1706.5	43.78
SQuAD	Standalone	1	1.000	1880.8	—
SQuAD	Serial post	1	3.044	5734.8	76.16
SQuAD	Async reader	2	1.010	5804.8	71.63/68.33
SQuAD	Clean + joint	2	1.004	5049.5	78.05

Clean + joint attains higher terminal quality and lower summed worker occupancy than historical async in every paired seed. The prescribed observation sequence approaches its mean quality in offline replay (Table 40); that intervention does not supply a live cost comparison. ORA supplies a different capability: optional state learns from the same continuing protected writer whose process is retained. The earlier deployment study (Table 24) prices a branch-only logical payload at 1.98 MiB versus 58.87 MiB for an additional joint trainable state, a 56.89 MiB difference when a clean state already exists. Both modes may share immutable model files. This payload saving and the earlier deployment memory/switching measurements are separate from VIII training costs. The two studies identify the resource and lifecycle conditions under which the constructions are useful.

K.6 Protected-path component accounting

Matrix IV executes protected and branch work serially in one process. The post protected-path ratio includes branch updates; the checkpoint ratio includes protected updates and publication, accounting for a separately served reader. These component ratios do not measure actual concurrent completion. Matrix V's shared-node timing, per-task ratios, and load incidents are retained in Appendix E.9.

Post exceeds the prespecified $1.05\times$ protected-path limit; $K{=}50$ and 250 meet it near $1.01\times$ in this quieter session. Reader work nevertheless takes about 2.2 protected-pass equivalents, failing the

Table 38: **Per-seed concurrent comparison.** Quality difference is drained async minus terminal clean + joint; occupancy ratio uses paired summed worker lifetimes. Seeds 15300 and 15301 use sequential task groups (S). Seed 15302 runs the two task groups concurrently on disjoint GPU pairs (P), under an execution amendment retaining the scientific configuration.

Task	Seed	Execution	Quality difference (pp)	Occupancy ratio	Reader lag (s)
GSM8K	15300	S	−5.49	1.148	670.7
GSM8K	15301	S	−6.16	1.142	661.3
GSM8K	15302	P	−5.09	1.210	773.0
SQuAD	15300	S	−7.00	1.143	1956.7
SQuAD	15301	S	−11.68	1.154	1984.3
SQuAD	15302	P	−10.46	1.139	1986.6

Table 39: **Low accounted protected-path overhead does not ensure a reader keeps pace.** Matrix IV one-block setting; descriptive medians over seeds and tasks. Device time is normalized to standalone; memory is peak reserved GiB. The $1.05\times$ rule applies to the accounted protected path. Full records are in Table 15.

Construction	Protected path	Device time	Memory	Branch/protected
Standalone	1.00	1.0	11.9	—
$\mathrm{ORA}_{\mathrm{post}}$	3.18	3.2	21.7	2.2
$\mathrm{ORA}_{\mathrm{ckpt}}$, $K{=}250$	1.01	3.2	23.7	2.2

keep-up check. On matched hardware this predicts completion roughly 1.2 protected passes after the writer. Total device time remains about $3.2\times$ standalone, and the checkpoint reader adds about 2 GiB of peak memory. VIII directly measures the lag and cutoff quality that this component accounting leaves unresolved.

Table 40: **Changing the observation sequence narrows the terminal quality gap.** Means over the same three retained seeds per task, on the 0–100 scale. H/P share the final base s_T; P is an offline intervention. Serial post and clean plus joint are completed VIII references.

Training / sequence	GSM8K accuracy	SQuAD F1
Historical H	38.20	68.33
Prescribed P	43.46	77.65
$P - H$	+5.27	+9.31
Serial post	43.11	76.16
Clean + joint	43.78	78.05

K.7 Historical and prescribed observation sequences

Question and fixed inputs. We test whether changing the observed host-version sequence changes branch quality with the protected trajectory and terminal fusion base held fixed. This is a post-outcome mechanism intervention on the six completed VIII trajectories, with seeds 15300–15302 for each task. The one-block Qwen2.5-0.5B setting, group-disjoint fit/tune/holdout partitions, model constructor, initial branch, batch order, branch stochastic stream, optimizer schedule, and number of updates are retained. Each GSM8K/SQuAD reader performs $1682/5002$ updates: two epochs, batch size eight, base learning rate 10^{-4}, branch multiplier two, no weight decay, and gradient-norm limit one. Original writer publications are immutable inputs; the study trains no new writer.

Observation policies. Historical H reads the publication recorded at each original reader update j. Prescribed P reads the largest retained publication step at most $j-1$, namely $\sigma_j = 250\lfloor (j-1)/250 \rfloor$. This is a pre-update convention indexed by reader progress; Matrix IV's checkpoint member instead uses the protected post-update convention. Both replay arms fuse the final branch with the same writer state s_T, including when T is not divisible by 250. The concrete sequences and publication hashes are frozen before execution. P is an offline controlled input sequence, whose live scheduling and retention costs require a separate measurement.

Historical reproduction and evaluation gates. Two 50-update smoke runs reproduce the initial and step-50 branches with full-horizon optimizer schedules. All six full H replays then reproduce named branch tensors and hashes at every retained boundary: initialization, multiples of 50, and the endpoint. There are 35 checks per GSM8K trajectory and 102 per SQuAD trajectory, totaling 411. Final branch, protected-state, and optimizer hashes equal their historical references, and terminal fused tensors equal the original drained artifacts. These are all retained branch boundaries, rather than a newly recorded all-component trace at every branch update.

Only after all six H receipts pass does P training begin. All twelve complete training receipts precede evaluation; all six P evaluation receipts precede the joint opening of quality values. Historical evaluations are reused after direct fused-state equality. The original VIII evaluator and holdout are used throughout. Source, model, prediction, score, gate, and final-analysis hashes bind the complete matrix. Shared-GPU training uses the original scientific settings and a fixed allocator ceiling; its elapsed times support execution monitoring only.

Table 41: **Every retained branch improves under the prescribed sequence.** Accuracy/F1 on the 0–100 scale. C is the terminal clean-plus-joint reference from VIII. All differences are descriptive paired differences on the same seeds.

Task	Seed	H	P	$P-H$	C	$P-C$
GSM8K	15300	38.420	42.436	+4.016	43.909	−1.473
GSM8K	15301	38.019	44.846	+6.827	44.177	+0.669
GSM8K	15302	38.153	43.106	+4.953	43.240	−0.134
SQuAD	15300	69.653	77.920	+8.267	76.649	+1.270
SQuAD	15301	66.769	77.483	+10.714	78.445	−0.962
SQuAD	15302	68.577	77.532	+8.955	79.041	−1.508

Effect and interpretation. The mean $P-H$ differences are 5.266 accuracy points on GSM8K and 9.312 F1 points on SQuAD. The corresponding mean gaps to clean plus joint shrink from $5.578/9.712$ to $0.312/0.400$ points. These comparisons locate the effect relative to a strong alternative; neither mean proximity nor mixed per-seed wins establishes equivalence or superiority to that alternative.

Historical H performs $879/871/933$ GSM8K updates and $2602/2593/2593$ SQuAD updates on s_T, in seed order. Prescribed P uses no s_T training updates and still improves after fusion with s_T. Thus more training exposure to the final host does not ensure better final quality on these trajectories. The intervention changes the allocation of updates across host versions and their pairing with batches and learning rates. It identifies the effect of the complete sequence replacement, without distinguishing those internal explanations or asserting a universal staleness law.

Inference boundary. The historical outcomes were known before this intervention was designed. The three seeds per task and original holdout are retained, and the study supplies no independent-seed confirmation. With three pairs, even unanimous signs give a minimum exact two-sided sign-flip $p = 0.25$; the protocol targets descriptive effects rather than significance. All six results are reported without seed or checkpoint selection. Testing a live policy requires measuring version availability, retained-state cost, writer and reader progress, cutoff quality, terminal quality, and total work in the same execution.

L Complete prospective observation-policy confirmations

Question and evidence level. The retained-trajectory intervention selected the reader-progress policy with its earlier outcomes already known. We then froze two prospective protocols before their new seed matrices: live confirmation (seeds 15400–15409) and shared-device quality confirmation (seeds 15500–15509), each on GSM8K and SQuAD. Each matrix has $80/80$ workflows, $20/20$ protected-state group gates, and $100/100$ artifact evaluations. All scheduled training and group gates precede formal evaluation; all artifact evaluations precede aggregate quality analysis. No seed was removed or substituted in either analysis. These are independent-seed confirmations conditional on the existing group-disjoint fit/tune/holdout partitions and Qwen2.5-0.5B recipe. They do not introduce untouched evaluation data, another model family, or full-model quality transfer.

Matched workflows and the strong comparator. The protected final decoder block and normalization use learning rate 10^{-4}; preceding query/value LoRA uses rank 8 and learning-rate multiplier 2. Training uses eager FP32, two epochs, batch size 8, zero weight decay, clipping norm 1, and a constant schedule. The protected and optional budgets are $1{,}682/5{,}002$ updates for GSM8K/SQuAD. Standalone supplies the seed's reference trajectory. H and P each have one protected writer and one reader on a second GPU, with identical initialization, branch batch order, stochastic stream, optimizer schedule, update count, and terminal fusion base s_T. H consumes the latest committed protected publication. Reader update j in P consumes

$$s_{\sigma_j}, \qquad \sigma_j = 250 \left\lfloor \frac{j-1}{250} \right\rfloor .$$

The reader waits for that version if needed; the writer never waits for the reader. Both policies retain all protected publications and save branch commits every 50 updates and at the endpoint. This symmetric instrumentation controls storage/commit behavior in their comparison, but is not a storage-optimized implementation of H.

Clean plus joint (C) runs clean standalone and ordinary joint training concurrently on two GPUs, supplying exact fallback from the clean run and enabled quality from joint training. It uses the same task, seed, data recipe, update horizon, and frozen evaluation procedure. Joint training may update its own host with optional loss, while ORA constrains the branch to preserve the single protected writer. Thus C is a meaningful strong operational alternative with an additional evolving host state. It is not constrained to the same protected-state ownership rule. No confirmation arm receives new tuning or intermediate quality-based selection.

Correctness and evaluation gates. For every group, the H/P writers and the clean role of C match the standalone protected loss trace and explicit endpoint components: named parameters/buffers, optimizer, scheduler, RNG, batch-order digest, and cursor. H/P publications also match; reader traces bind committed publications and verify P's source-step formula. These are recorded loss traces, publication checks, and endpoint checks, not every-component state comparisons after every update. Engineering smoke separately verifies live P against deterministic offline replay. Each group evaluates H/P writer-cutoff and fully drained artifacts plus C terminal, using the frozen single-example greedy evaluator on $747/5{,}002$ held-out examples. The branch is available at writer cutoff in all retained H/P workflows; cutoff scores remain descriptive.

Statistical unit and frozen decisions. The seed is the unit. For each study separately, primary inference uses the ten paired drained $P - H$ differences per task, all 2^{10} sign assignments, one-sided exact sign-flip tests, and Holm correction over the two tasks. A pass requires adjusted $p < 0.05$, paired 95% t-interval lower bound above zero, and mean above $1\,pp$. Noninferiority to C uses a separate two-task family, tests differences shifted by $+1\,pp$, and requires adjusted $p < 0.05$ plus interval lower bound above $-1\,pp$. These tests assume sign symmetry/exchangeability under the corresponding null. Correction is within the declared families, not across all paper hypotheses. With ten seeds the minimum one-sided exact p is $1/1024$; a fixed seed budget does not ensure power for a $1\,pp$ noninferiority margin. Passing that gate is not equivalence or superiority; failure does not prove absence of benefit.

Complete quality outcome. All 40 primary paired differences are positive and all four task-level superiority gates pass. The GSM8K noninferiority interval lies wholly below $-1\,pp$ in both studies. For live SQuAD, noninferiority passes; for shared SQuAD, its interval crosses the $-1\,pp$ boundary and the adjusted test fails. Therefore neither study establishes joint noninferiority. Tables 42–45 retain complete statistics, all seed pairs, and descriptive cutoff results. The studies are neither pooled nor treated as a randomized comparison of external load: H depends on relative worker progress, and both seed and execution conditions differ between matrices.

Realized observation sequences. The recorded reader traces show what the two policies actually read (Figure 2). Under H, the writer outpaces the reader, so the version sequence is set by relative progress: across the ten live seeds, 52–64% of GSM8K and 53–63% of SQuAD reader updates read the terminal publication s_T, and the sequence differs from seed to seed. Under P, the sequence is identical across seeds by construction, no update reads s_T, and the branch still fuses with s_T at evaluation. The confirmed $P - H$ gain therefore accompanies a policy with fewer terminal-state

updates and a load-independent version schedule; this description does not identify which of these features mediates the gain.

Table 42: **All prespecified quality decisions.** Paired differences in pp, 95% t intervals, and exact one-sided sign-flip p values. Holm correction is within each two-task family in each study. Primary superiority also requires a mean above 1 pp; noninferiority requires the interval lower bound above -1 pp and adjusted $p < 0.05$.

Study	Task	Contrast	Mean	95% CI	Raw p	Holm p	Pass
Live	GSM8K	$P - H$	$+3.989$	$[3.251, 4.728]$	0.00098	0.00195	Yes
Live	SQuAD	$P - H$	$+10.575$	$[9.184, 11.966]$	0.00098	0.00195	Yes
Live	GSM8K	$P - C$	-2.423	$[-3.561, -1.285]$	0.98730	0.98730	No
Live	SQuAD	$P - C$	-0.257	$[-0.666, 0.151]$	0.00195	0.00391	Yes
Shared	GSM8K	$P - H$	$+4.404$	$[3.433, 5.375]$	0.00098	0.00195	Yes
Shared	SQuAD	$P - H$	$+9.450$	$[8.343, 10.557]$	0.00098	0.00195	Yes
Shared	GSM8K	$P - C$	-2.303	$[-3.540, -1.066]$	0.99316	0.99316	No
Shared	SQuAD	$P - C$	-0.751	$[-1.338, -0.163]$	0.17773	0.35547	No

Table 43: **Live confirmation: all paired terminal scores.** Accuracy/F1 and differences in pp. No seed is excluded or replaced in the analysis.

Task	Seed	H	P	C	$P - H$	$P - C$
GSM8K	15400	37.885	42.704	44.444	$+4.819$	-1.740
GSM8K	15401	37.082	42.838	46.854	$+5.756$	-4.016
GSM8K	15402	40.696	45.917	45.248	$+5.221$	$+0.669$
GSM8K	15403	38.688	42.436	43.373	$+3.748$	-0.937
GSM8K	15404	39.224	43.373	45.917	$+4.150$	-2.544
GSM8K	15405	38.956	41.633	45.783	$+2.677$	-4.150
GSM8K	15406	39.090	42.570	44.177	$+3.481$	-1.606
GSM8K	15407	40.161	43.909	47.791	$+3.748$	-3.882
GSM8K	15408	40.562	43.106	46.988	$+2.544$	-3.882
GSM8K	15409	38.822	42.570	44.712	$+3.748$	-2.142
SQuAD	15400	69.097	77.665	77.617	$+8.568$	$+0.047$
SQuAD	15401	65.768	77.189	77.784	$+11.421$	-0.595
SQuAD	15402	68.324	77.074	77.495	$+8.751$	-0.421
SQuAD	15403	69.570	78.547	78.612	$+8.977$	-0.065
SQuAD	15404	65.672	77.515	78.344	$+11.843$	-0.829
SQuAD	15405	68.033	79.633	78.718	$+11.600$	$+0.915$
SQuAD	15406	68.635	77.249	77.854	$+8.613$	-0.605
SQuAD	15407	62.748	77.333	78.345	$+14.585$	-1.011
SQuAD	15408	67.485	78.798	78.535	$+11.313$	$+0.263$
SQuAD	15409	67.703	77.781	78.053	$+10.078$	-0.272

Table 44: **Shared confirmation: all paired terminal scores.** Accuracy/F1 and differences in pp. No seed is excluded or replaced in the analysis.

Task	Seed	H	P	C	$P-H$	$P-C$
GSM8K	15500	39.357	44.846	45.515	+5.489	−0.669
GSM8K	15501	40.696	43.106	49.264	+2.410	−6.158
GSM8K	15502	37.751	42.838	43.775	+5.087	−0.937
GSM8K	15503	37.617	42.838	44.578	+5.221	−1.740
GSM8K	15504	39.491	42.704	45.114	+3.213	−2.410
GSM8K	15505	40.027	43.775	44.846	+3.748	−1.071
GSM8K	15506	37.216	44.177	45.515	+6.961	−1.339
GSM8K	15507	40.295	44.980	46.319	+4.685	−1.339
GSM8K	15508	37.751	41.901	45.515	+4.150	−3.614
GSM8K	15509	37.885	40.964	44.712	+3.079	−3.748
SQuAD	15500	65.040	77.745	78.386	+12.705	−0.640
SQuAD	15501	69.566	77.974	78.752	+8.408	−0.777
SQuAD	15502	69.538	78.636	78.394	+9.098	+0.242
SQuAD	15503	67.839	76.568	77.469	+8.730	−0.901
SQuAD	15504	69.761	78.453	78.075	+8.692	+0.378
SQuAD	15505	68.379	76.386	78.118	+8.008	−1.731
SQuAD	15506	69.972	77.744	78.717	+7.772	−0.972
SQuAD	15507	66.616	77.892	78.195	+11.277	−0.302
SQuAD	15508	67.187	77.184	77.657	+9.997	−0.473
SQuAD	15509	64.546	74.356	76.685	+9.810	−2.328

Table 45: **Writer-cutoff quality, recorded live timing, reader lag, and retained publications.** Cutoff rows report mean accuracy/F1. Timing rows show geometric paired ratios and log-scale 95% intervals, retained as descriptive arithmetic only. They do not establish uncontended timing or a full cost comparison.

Measurement	GSM8K	SQuAD
Live H cutoff	40.054	71.424
Live P cutoff	43.454	77.670
Shared H cutoff	40.482	70.800
Shared P cutoff	43.521	74.816
Live P writer / standalone time	1.013 [1.006, 1.020]	1.017 [1.012, 1.023]
Live P/C worker occupancy	1.247 [1.198, 1.298]	1.214 [1.178, 1.250]
Live P reader lag / full writer training time (median)	1.25	1.21
Live P retained publications (GB, median)	0.48	1.31

L.1 Execution audit and the limits of cost inference

Training deviations and eligibility. The live protocol originally specified one fixed GPU pair and sequential task–seed groups. Versioned amendments permitted three simultaneous disjoint pairs while retaining the within-group role mapping and cyclic workflow order. This enables host contention even when a pair has no foreign GPU process. SQuAD seed 15403 has historical supervisor-monitor gaps for standalone and H. The H receipt explicitly records `load_confounded=null`; its occupancy was reconstructed from worker-observed lifetime rather than complete supervisor launch-to-exit observations. Unknown load is not quiet load, and the standalone receipt's false load flag does not remove its documented monitoring gap. Separately, external activity on GPU 6 interrupted a partial clean-plus-joint attempt for SQuAD seed 15406. Before any workflow in that group completed, the whole group moved from pair $5/6$ to $0/1$ and reran under the same scientific settings and seed. The earlier attempt, failures, and hashes remain retained. No trial is excluded from the final ten-seed comparisons because of its quality.

Evaluation recovery and prediction integrity. The adaptive-evaluation v12 incident ledger records 19 device resource interruptions and 27 prediction backups. These counts refer to v12; an earlier v11 all-queue interruption is retained separately. The recovery reader accepts the historical nested and flattened prediction schemas only after checking the full lock/manifest/state/task/training-gate binding, unique example identifiers, generated-token structure, and score structure. It reuses complete

legal rows, preserves old prediction files and failure records, and backs up a malformed terminal fragment before repair. Valid completed rows are neither overwritten nor duplicated. These evaluation interruptions affect evaluation completion and throughput; they are not themselves evidence that earlier training times were contaminated. Complete artifact counts, bound scores, and prediction digests support the quality analysis.

Descriptive timing and strong-baseline cost. Table 45 reports the original ten-seed arithmetic without upgrading its timing eligibility. The recorded geometric P/standalone writer-time ratios are $1.013/1.017$; their intervals lie below the nominal 1.05 threshold. The P/C summed worker-occupancy ratios are $1.247/1.214$, with intervals $[1.198, 1.298]/[1.178, 1.250]$, exceeding the same threshold. Occupancy sums process lifetimes including startup, waiting, publication, saving, and hashing; it is neither energy nor kernel-active time. Reader lag is the time from writer completion to reader completion, divided by that run's full writer training duration (two epochs); the reported $1.25/1.21$ medians use this duration as one unit. The monitoring gaps, concurrent groups, and hardware migration prevent an uncontended timing claim. The audit-qualified analysis records timing passes as unknown and retains the original threshold arithmetic separately. Shared-device confirmation was designated quality-only from the outset and supplies no timing or cost inference. Neither study establishes a combined quality–cost advantage.

Separate timing sensitivity. A prespecified execution supplement reruns only P and standalone at SQuAD seed 15403, with $5{,}002$ updates, pair $2/3$, and order $P \rightarrow$standalone. It passes protected equality and records writer/base training-time and occupancy ratios of 1.017018 and 1.017648. This single-seed descriptive result ran on a host shared with another training pair. It neither replaces the original trial nor enters the ten-seed inference and cannot repair the historical H monitoring gap. Detailed timing-eligibility reconstruction and comprehensive retry-waste accounting remain open; reported final-attempt occupancy is not a total experimental resource bill.

Evidence map and portable verification. The supplement includes each study's protocol, source locks, versioned execution amendments, all formal group/workflow receipts, evaluation receipts, and score summaries. The authoritative live result is the audit-qualified `audited_analysis_v13` analysis, not the earlier analyzer's unqualified timing verdict. Shared quality uses its completed root analysis. `confirmation_evidence.py` binds both analyses, reconstructs paired means, intervals, sign-flip tests, Holm decisions, and descriptive timing ratios, and checks that every original seed and evaluation is present. Local verification hashes all new prediction files. The portable package supplies their digests as explicit omitted-artifact records and reproduces statistics from bound score receipts; it does not regenerate tokens or re-establish protected tensor equality. Frozen experiment sources and prior releases are preserved. This distinction keeps receipt-level reproducibility separate from a full training/evaluation rerun.

M DIRECT P/H CONTROL OF BRANCH BATCH ORDER

Question and evidence level. Does the version-assignment effect persist with a different branch sample order? We cross archived H/P sequences with original and independent orders on the six trajectories of Appendix K.7, reusing seeds 15300–15302 per task and the existing holdout. Historical outcomes were known; the design, permutations, gates, and contrasts were fixed before new training. This is an exploratory intervention.

Fixed trajectory, different version–batch inputs. All four cells reuse each coordinate's protected publications, branch initialization, optimizer, update count, stochastic stream, and final fusion base s_T. The original Qwen2.5-0.5B configuration uses two epochs, batches of eight, constant branch learning rate 2×10^{-4}, and $1{,}682/5{,}002$ updates on GSM8K/SQuAD. Original order uses a CPU permutation generator seeded with $\text{seed} + 300007$; independent order uses $\text{seed} + 900011$. Both H/P receive the same new order, including incomplete last batches. Each epoch contains the same examples exactly once. Accidental positional matches are retained; no permutation is selected using quality. H replays its archived versions and P reads $250\lfloor (j-1)/250 \rfloor$ at reader update j.

Table 46: **Independent branch order attenuates the P/H gap.** Means over three retained seeds per task; accuracy/F1 differences in percentage points. The interaction subtracts the original-order gap from the independent-order gap. These are descriptive effects, with no residual-superiority claim.

Task	Original $P - H$	Independent $P - H$	Interaction
GSM8K	$+5.266$	$+0.803$	-4.462
SQuAD	$+9.312$	$+1.394$	-7.918

Table 47: **All six paired outcomes.** Scores are on the 0–100 scale. Orig./Ind. denote original and independent branch order. Every pair has a smaller $P - H$ gap under independent order; five residual gaps are positive and one is zero.

Task	Seed	Orig. H	Orig. P	Ind. H	Ind. P	Orig. $P - H$	Ind. $P - H$	Interaction
GSM8K	15300	38.420	42.436	36.546	38.822	$+4.016$	$+2.276$	-1.740
GSM8K	15301	38.019	44.846	41.499	41.499	$+6.827$	$+0.000$	-6.827
GSM8K	15302	38.153	43.106	38.956	39.090	$+4.953$	$+0.134$	-4.819
SQuAD	15300	69.653	77.920	71.733	74.296	$+8.267$	$+2.564$	-5.703
SQuAD	15301	66.769	77.483	73.092	74.242	$+10.714$	$+1.150$	-9.564
SQuAD	15302	68.577	77.532	73.503	73.971	$+8.955$	$+0.468$	-8.488

Table 48: **Independent order raises mean H and lowers mean P.** Three retained seeds per task; mean accuracy/F1 on the 0–100 scale and paired gaps in pp. Both orders use the same host trajectory, fixed H/P version sequences, and final s_T. Descriptive follow-up, not fresh-seed confirmation; all six paired gaps shrink.

	Original branch order			Independent branch order		
Task	H	P	$P - H$	H	P	$P - H$
GSM8K	38.197	43.463	5.266	39.000	39.804	0.803
SQuAD	68.333	77.645	9.312	72.776	74.170	1.394

Reproduction and execution audit. After four 50-update smoke runs, all twelve original readers match 822 archived checkpoints, endpoint branch/protected/optimizer hashes, and reader/fused tensors. Twelve independent-order readers then train, followed by 96 archived-example evaluator token checks and twelve full evaluations ($747/5{,}002$ examples per GSM8K/SQuAD artifact). Outcomes open as a complete matrix. The eight-GPU amendment preserves four running workers and records 48 reproduction/training/evaluator/evaluation jobs. Shared execution supports quality evidence only.

Interpretation and limits. Independent order lowers mean P scores and raises mean H scores. The remaining positive mean gaps do not establish residual superiority or population robustness. Their roughly 85% reduction is descriptive, not a mediated-effect fraction: shuffling changes example order,

batch composition, and version–example pairing together. It does not identify which is responsible or establish that the host advanced too quickly. This limits extrapolation of the original-order effect. Seeds, checkpoints, and permutations were fixed before observing the new scores.

Reproduction material. The supplement supplies the experiment's original protocol and lock, the eight-GPU execution amendment, all six fixed batch schedules, training traces and completion receipts, full new prediction records, scores, and analysis. The manuscript-side `ph_batch_evidence.py` verifies source bindings, schedule and RNG-stream agreement, recorded gates, and paired contrasts. Portable receipt verification checks supplied prediction aggregates and declares omitted model/data tensors; it does not regenerate tokens or re-establish tensor equality by training.

N Independent Branch Order on Ten Fresh Seeds

Question and evidence level. Does the confirmed $P - H$ gain survive a branch order that is independent of the host's, and how much of it depends on that order? The three-seed intervention of Appendix M was post-outcome and descriptive. This study preregisters the same intervention on all ten live confirmation seeds per task before any independent-order outcome existed (lock `7011d74931df5d2d`). Both readers are replayed offline from the live workflow's retained publications: H uses the version sequence its live reader recorded, P the prescribed sequence. Initialization, branch RNG stream, optimizer, recipe, budget, and fusion base s_T are the live readers'; only the branch batch order changes to one prespecified permutation per seed.

Reproduction gate. On seed 15400 per task, original-order replays of both arms reproduce the live branch hash at every retained commit (34 on GSM8K, 101 on SQuAD) and the live fused artifact bitwise; independent-order training for a task started only after both gates passed. All 84 jobs completed with no failure.

Decision rules and outcomes. Seed is the unit; one-sided exact sign-flip tests over all 2^{10} assignments use Holm correction over the two tasks per family. A task passes when Holm $p < 0.05$ and the paired 95% t-interval lower bound exceeds the null, together with any stated mean requirement. H_{G1} tests the residual advantage $P_{\mathrm{own}} - H_{\mathrm{own}}$ (null 0, mean $> 1\,pp$); H_{G2} tests the batch dependence $(P - H) - (P_{\mathrm{own}} - H_{\mathrm{own}})$ (null 0). The $1\,pp$ threshold applies to the mean; interval and significance tests use zero. H_{G1} passes on SQuAD ($+1.18$ $[+0.61, +1.76]$, Holm $p = 0.0020$, $10/10$ positive) and fails on GSM8K ($+1.02$ $[-0.21, +2.24]$, $p = 0.0352$, $7/10$ positive), so no family-level residual claim is made. H_{G2} passes on both tasks ($+2.97$ $[+1.54, +4.40]$ and $+9.39$ $[+8.14, +10.64]$, Holm $p = 0.0029/0.0020$).

Table 49: **Independent-order replay on ten live seeds per task: decisions and descriptive contrasts.** Differences in pp with paired 95% intervals; means on the 0–100 scale.

Contrast	GSM8K	SQuAD	Rule
H_{G1}: $P_{\mathrm{own}} - H_{\mathrm{own}}$	$+1.02$ $[-0.21, +2.24]$	$+1.18$ $[+0.61, +1.76]$	Holm p = $0.0352/0.0020$, pass false/true
H_{G2}: $(P - H) - (P_{\mathrm{own}} - H_{\mathrm{own}})$	$+2.97$ $[+1.54, +4.40]$	$+9.39$ $[+8.14, +10.64]$	Holm p = $0.0029/0.0020$, pass true/true
$P - P_{\mathrm{own}}$	$+2.18$ $[+0.77, +3.59]$	$+3.90$ $[+3.30, +4.50]$	descriptive, paired
$H - H_{\mathrm{own}}$	-0.79 $[-1.71, +0.13]$	-5.49 $[-6.70, -4.28]$	descriptive, paired
$P_{\mathrm{own}} - C$	-4.61 $[-5.86, -3.35]$	-4.16 $[-4.68, -3.63]$	descriptive, paired
mean P (live order)	43.11	77.88	$n = 10/10$
mean H (live order)	39.12	67.30	$n = 10/10$
mean P_{own}	40.92	73.98	$n = 10/10$
mean H_{own}	39.91	72.79	$n = 10/10$
mean C	45.53	78.14	$n = 10/10$

Interpretation and limits. Independent order lowers P by $2.18/3.90\,pp$ and changes H by $+0.79/+5.49\,pp$; $9/10$ and $10/10$ paired gaps shrink. The permutation changes sample order, batch composition, and version–example pairing together, so the study establishes batch dependence and a task-specific residual, not a mediator. The three-seed study remains as the exploratory precursor; its seeds (15300–15302) are disjoint from these.

Table 50: **Every seed of the independent-order replay.** Accuracy/F1 on the 0–100 scale; gaps in pp.

Task	Seed	H	P	H_{own}	P_{own}	$P - H$	$P_{\text{own}} - H_{\text{own}}$	Interaction
GSM8K	15400	37.88	42.70	40.16	40.29	+4.82	+0.13	+4.69
GSM8K	15401	37.08	42.84	37.35	38.42	+5.76	+1.07	+4.69
GSM8K	15402	40.70	45.92	38.82	40.83	+5.22	+2.01	+3.21
GSM8K	15403	38.69	42.44	40.03	39.89	+3.75	−0.13	+3.88
GSM8K	15404	39.22	43.37	40.03	38.96	+4.15	−1.07	+5.22
GSM8K	15405	38.96	41.63	40.43	41.77	+2.68	+1.34	+1.34
GSM8K	15406	39.09	42.57	40.96	40.96	+3.48	+0.00	+3.48
GSM8K	15407	40.16	43.91	39.36	44.58	+3.75	+5.22	−1.47
GSM8K	15408	40.56	43.11	41.37	42.17	+2.54	+0.80	+1.74
GSM8K	15409	38.82	42.57	40.56	41.37	+3.75	+0.80	+2.95
SQuAD	15400	69.10	77.66	73.09	74.27	+8.57	+1.17	+7.40
SQuAD	15401	65.77	77.19	72.48	73.67	+11.42	+1.19	+10.24
SQuAD	15402	68.32	77.07	73.39	74.05	+8.75	+0.66	+8.09
SQuAD	15403	69.57	78.55	74.15	75.15	+8.98	+0.99	+7.98
SQuAD	15404	65.67	77.51	72.27	74.59	+11.84	+2.31	+9.53
SQuAD	15405	68.03	79.63	71.62	74.25	+11.60	+2.63	+8.97
SQuAD	15406	68.64	77.25	72.76	72.87	+8.61	+0.11	+8.51
SQuAD	15407	62.75	77.33	71.88	72.82	+14.59	+0.95	+13.64
SQuAD	15408	67.49	78.80	72.31	73.85	+11.31	+1.55	+9.77
SQuAD	15409	67.70	77.78	73.97	74.25	+10.08	+0.29	+9.79

Reproduction material. The supplement supplies the protocol, lock, frozen schedules and version sequences, gate receipts, per-run traces, branch-hash checks, scores, and the one-shot analysis; `independent_order_evidence.py` rebinds the manuscript to it.

O Full-Model Reader-Progress versus Terminal Observation

Question and prospective scope. We test whether the matched one-block P advantage extends to full-model training. The protocol, code, inputs, and eight new seeds per task (15600–15607) were frozen before any outcome of this study (lock `66383ffc1aa25351`). Qwen2.5-0.5B-Instruct, the existing GSM8K/SQuAD partitions, and rank-8 LoRA on query/value projections in layers 0–22 are retained; all non-LoRA base parameters train. The fixed full-model recipe uses host learning rate 2×10^{-5}, branch rate 2×10^{-4}, AdamW, constant schedules, zero decay, clipping norm 1, batch size 8, and two epochs. There is no outcome-based retuning, seed enlargement, or optional quality stopping. This is a training-setting test, with no new model family, scale, or evaluation split.

Paired design and serial execution. Each of the 16 groups trains one clean host trajectory and two branches with the same initialization, sample batches/order, branch RNG stream, optimizer, learning rate, budget, and final fusion host. P observes $s_{250\lfloor (j-1)/250 \rfloor}$ at update j; terminal-only always observes s_T. At each publication, the serial implementation first executes up to 250 reader updates and then the corresponding host updates with the optional branch disabled. It realizes the specified version–batch assignment without retaining every full-model snapshot. The terminal arm resets the branch, optimizer, and RNG after saving P. The horizons are $1682/5002$ updates. This design tests completed quality; it does not measure asynchronous throughput, latency, or waiting.

Engineering and evidence levels. CPU checks test the version schedule, ownership and mutation detection, the branch-step reference, statistics, and completeness. A known one-block gate reproduces the retained live branch at update 50. A separate full-model smoke gate (32 updates per task, cadence 8, seed 15599) matches the protected loss trace and endpoint parameters/optimizer/scheduler/RNG to standalone and checks the optimized branch step against the reference. Formal runs check protected tensor version counters during branch steps, publication-boundary hashes, terminal noninterference, and identical protected parameters/buffers in paired artifacts. Standalone matching at the smoke horizon is distinct from a full-horizon standalone replay; the latter was not run for these new seeds. Formal training and its complete gate precede all held-out evaluation, and all 32 evaluations precede the frozen aggregate analysis. All scheduled jobs finish without failure.

Frozen inference and result. Seed is the unit. The primary estimand is P minus terminal-only in accuracy/F1 percentage points, with paired 95% t intervals and exact one-sided sign-flip tests over 2^8 assignments. Holm correction is across the two tasks. A task passes only if adjusted $p < 0.05$, the interval lower bound exceeds zero, and the mean exceeds $1\,pp$; a joint statement requires both tasks. The sign-flip test assumes sign symmetry under its null. Neither task passes. All eight differences per task are negative, and both intervals lie below zero (Table 4). The one-sided superiority tests give Holm $p = 1.0000/1.0000$; this value is not an equivalence test.

Table 51: **All full-model observation pairs.** Final fused accuracy/F1 (0–100); P minus terminal-only in pp.

Task	Seed	P	Terminal-only	Difference
GSM8K	15600	22.09	32.66	−10.58
GSM8K	15601	22.09	33.47	−11.38
GSM8K	15602	26.77	34.54	−7.76
GSM8K	15603	27.44	34.00	−6.56
GSM8K	15604	27.44	32.26	−4.82
GSM8K	15605	27.18	34.14	−6.96
GSM8K	15606	24.63	31.86	−7.23
GSM8K	15607	25.03	33.20	−8.17
SQuAD	15600	68.37	73.01	−4.63
SQuAD	15601	68.62	72.18	−3.56
SQuAD	15602	69.06	72.27	−3.20
SQuAD	15603	67.47	72.65	−5.18
SQuAD	15604	69.21	73.07	−3.86
SQuAD	15605	67.98	73.81	−5.83
SQuAD	15606	68.19	73.38	−5.19
SQuAD	15607	66.79	73.23	−6.44
GSM8K	mean	25.33	33.27	−7.93
SQuAD	mean	68.21	72.95	−4.74

Interpretation and reproduction. This direct test of P adds to the earlier full-model $\mathrm{ORA}_{\mathrm{post}}$ reversal (Appendix E.7). It establishes a negative transfer result for the specified recipe, not a causal effect of the protected fraction: the host learning rate and training configuration also differ from the one-block comparison. It does not identify the cause of the reversal or invalidate the earlier matched effect. The supplement includes the frozen protocol/code, engineering and training receipts, all paired scores, and formal observation traces and predictions. `full_observation_evidence.py` verifies their bindings, reconstructs statistics and aggregates recorded prediction scores, and generates these tables. Large tensors and external caches remain explicit omissions in the portable release; this verification does not regenerate tokens or re-establish tensor equality.

P Matched Terminal-Only Observation and Branch Transfer to the Clean Host

Question and frozen inputs. This follow-up asks whether the observed version sequence changes quality at a fixed batch order, and whether a jointly trained branch can transfer to the clean final host. The protocol and decision rules were frozen before any new-arm outcome (lock `7d82886d476f3c55`); the live $P/H/C$ results were already known. Inputs are the completed live groups (seeds 15400–15409): the clean host s_T, shown bitwise equal to the standalone and writer hosts, the joint run's branch tensors, the P reader's initial branch and sample-order digest, and the frozen evaluator. This is a prospective matched follow-up on retained trajectories.

Arms. *C branch on clean s_T*: the joint run's branch tensors fused with the clean host, evaluation only; this transfer satisfies the final-host requirement. *Host alone at s_T*: the clean host on its branchless route. *Two-stage, host order*: branch-only training on clean s_T with the same initial branch, two-epoch host sample order, branch RNG stream, multiplier 2, constant schedule, and clipping as the live P reader; only the observed versions change. The tuned secondary uses the

Matrix II recipe (multiplier 8, linear decay) on the same order, testing sensitivity to the comparator's recipe. Host hashes are checked against the live base endpoint before and after training.

Decision rules. Seed is the unit. Each family tests drained P minus the arm with one-sided exact sign-flip tests over all 2^{10} assignments, Holm over the two tasks; a task passes when the adjusted $p < 0.05$, the paired 95% t-interval lower bound exceeds 0, and the mean exceeds 1 pp. Failure to pass establishes neither equivalence nor noninferiority; this governs the interpretation of the unresolved GSM8K transfer comparison.

Table 52: **Admissibility study: paired decisions and descriptive arm means.** Differences in pp with paired 95% intervals; means on the 0–100 scale over ten seeds. Rows marked descriptive carry no inferential claim.

Contrast	GSM8K	SQuAD	Rule
H_{E1}: $P - C$ branch on clean s_T	-0.92 $[-2.21, +0.36]$	$+7.96$ $[+4.85, +11.07]$	Holm p = 0.9365/0.0020, pass false/true
H_{E2}: P − Two-stage, host order	$+4.67$ $[+3.50, +5.85]$	$+13.87$ $[+12.99, +14.74]$	Holm p = 0.0020/0.0020, pass true/true
secondary: P − Two-stage, host order, tuned recipe	$+4.31$ $[+2.91, +5.71]$	$+7.75$ $[+7.24, +8.26]$	Holm p = 0.0020/0.0020, pass true/true
C branch on clean s_T − C terminal	-1.50 $[-2.94, -0.06]$	-8.21 $[-11.27, -5.16]$	descriptive, paired
Two-stage, host order − H	-0.68 $[-1.43, +0.06]$	-3.29 $[-4.21, -2.37]$	descriptive, paired
C branch on clean s_T − H	$+4.91$ $[+3.75, +6.07]$	$+2.62$ $[-1.20, +6.43]$	descriptive, paired
P − host alone	$+6.08$ $[+4.95, +7.20]$	$+22.38$ $[+21.52, +23.23]$	descriptive, paired
P	43.11 ± 1.2	77.88 ± 0.8	mean ± SD, $n = 10/10$
H	39.12 ± 1.1	67.30 ± 2.0	mean ± SD, $n = 10/10$
C (joint host)	45.53 ± 1.4	78.14 ± 0.4	mean ± SD, $n = 10/10$
C branch on clean s_T	44.03 ± 1.8	69.92 ± 4.3	mean ± SD, $n = 10/10$
Host alone at s_T	37.03 ± 1.2	55.50 ± 0.7	mean ± SD, $n = 10/10$
Two-stage, host order	38.43 ± 1.5	64.01 ± 1.6	mean ± SD, $n = 10/10$
Two-stage, host order, tuned recipe	38.80 ± 1.3	70.13 ± 0.6	mean ± SD, $n = 10/10$

Table 53: **Admissibility study: every seed.** Accuracy/F1 on the 0–100 scale; blank cells are arms incomplete at the frozen deadline.

Task	Seed	P	H	C	C br. on s_T	Host alone	TS host	TS host tuned
GSM8K	15400	42.70	37.88	44.44	39.36	35.61	37.08	38.02
GSM8K	15401	42.84	37.08	46.85	44.18	36.41	37.35	39.49
GSM8K	15402	45.92	40.70	45.25	45.52	36.95	38.96	37.88
GSM8K	15403	42.44	38.69	43.37	44.44	34.94	37.88	39.22
GSM8K	15404	43.37	39.22	45.92	46.18	38.55	38.96	36.41
GSM8K	15405	41.63	38.96	45.78	44.04	38.15	38.96	38.69
GSM8K	15406	42.57	39.09	44.18	44.44	36.28	37.75	38.15
GSM8K	15407	43.91	40.16	47.79	44.04	38.42	39.22	38.96
GSM8K	15408	43.11	40.56	46.99	44.58	37.08	41.77	41.50
GSM8K	15409	42.57	38.82	44.71	43.51	37.88	36.41	39.63
SQuAD	15400	77.66	69.10	77.62	74.26	54.29	64.72	69.87
SQuAD	15401	77.19	65.77	77.78	74.01	55.45	62.05	68.82
SQuAD	15402	77.07	68.32	77.49	65.64	55.80	63.54	70.21
SQuAD	15403	78.55	69.57	78.61	73.61	55.72	65.65	70.99
SQuAD	15404	77.51	65.67	78.34	72.87	54.84	61.70	70.04
SQuAD	15405	79.63	68.03	78.72	70.64	55.47	66.20	70.40
SQuAD	15406	77.25	68.64	77.85	61.81	56.73	65.02	70.19
SQuAD	15407	77.33	62.75	78.34	72.76	55.37	62.29	70.19
SQuAD	15408	78.80	67.49	78.54	66.56	54.96	64.03	70.50
SQuAD	15409	77.78	67.70	78.05	67.06	56.38	64.92	70.06

Table 54: **Selected terminal-only comparison.** Accuracy/F1 on estimation half B (0–100), ten paired seeds per task. Differences are $P - T$ in pp with paired 95% intervals.

Comparison	Task	P	Selected T	$P - T$ [95% CI]	Holm p	Pass
Primary	GSM8K	45.71	40.46	$+5.25$ $[+3.94, +6.57]$	0.0020	Yes
Primary	SQuAD	77.03	70.97	$+6.06$ $[+5.37, +6.75]$	0.0020	Yes
Secondary	GSM8K	45.76	40.46	$+5.31$ $[+3.08, +7.54]$	0.0020	Yes
Secondary	SQuAD	77.76	70.97	$+6.79$ $[+6.06, +7.51]$	0.0020	Yes

Primary: full-budget P versus terminal-only selected from four recipes and five checkpoints on half A. Secondary: P also selects among five checkpoints; it has no new learning-rate search. Each family uses one-sided tests with Holm correction across the two tasks. Pass requires mean > 1 pp, interval lower bound > 0, and Holm $p < 0.05$. The secondary comparison does not replace the primary. A/B reuse historical holdout data; this is not fresh-data confirmation.

Q TERMINAL RECIPE AND CHECKPOINT SELECTION

Selection-sensitive terminal comparison. This follow-up asks whether terminal-only optimization and stopping-time selection explain the shared-stream observation advantage. The locked design uses the ten retained host trajectories per task, the same final deployment host, branch initialization, training stream, and maximum update budget. Terminal-only candidates combine four learning-rate recipes with five checkpoints: 2×10^{-4} with a constant schedule, and 8×10^{-4}, 1.6×10^{-3}, or 3.2×10^{-3} with linear decay. The checkpoint steps are $\{100, 400, 850, 1250, 1682\}$ for GSM8K and $\{250, 1250, 2500, 3750, 5002\}$ for SQuAD. AdamW, zero weight decay, gradient clipping at 1, batch size 8, and two epochs follow the retained recipe. All candidates deploy on the same final clean host.

Selection data and asymmetric search budgets. Fixed group-disjoint halves of the historical holdout separate selection (A) from estimation (B): GSM8K has 374/373 examples and SQuAD 2,501/2,501. Each seed selects the highest A score, breaking ties by later checkpoint and then larger learning-rate multiplier. Terminal-only has 20 candidates per seed; the primary comparison uses the original full-budget P without selection. A secondary comparison selects P from five retained checkpoints, with no new learning-rate search. This gives terminal-only more selection opportunities and does not compare globally optimized policies. The terminal grid entails four full training runs per trajectory, with intermediate checkpoints retained from each run; candidate count is not a count of independent training runs. Previously retained full-budget predictions are reusable only after source and scorer checks, and terminal reuse additionally requires reproduction checks against the original run. Actual execution and reuse counts are reported below.

Chronology and inference. The design was locked after the earlier data-separation results were known. Training uses the historical fit-plus-tune pool; neither that tune subset nor B is previously unseen research data. A/B separation protects this follow-up's selection rule, but it does not create fresh-data confirmation. The primary estimand is full-budget P minus selected terminal-only on B; selected P minus selected terminal-only is secondary. Each family uses ten paired seeds per task, paired 95% t intervals, exact one-sided sign-flip tests, and Holm correction across the two tasks. The prespecified descriptive superiority criterion requires a mean above 1 percentage point, a positive interval lower bound, and adjusted $p < 0.05$ on both tasks. The secondary comparison cannot replace a failed primary. All four terminal endpoints, terminal and secondary P selection gains or losses, and recipe/checkpoint counts are reported. Secondary P checkpoint counts retain all five candidates, including unselected ones; selection at the largest terminal learning rate leaves further tuning sensitivity unresolved.

Complete results and selection behavior. Full-budget P passes the primary rule on both tasks, with all ten paired differences positive per task. Secondary checkpoint selection also passes on both tasks. The terminal search improves SQuAD by $2.04\,pp$ over the original tuned endpoint, while the GSM8K selection gain is unresolved. GSM8K's selected terminal mean (40.46) is below its fixed multiplier-16 endpoint (41.39), illustrating selection noise across A and B; we retain the locked A-based choice. The multiplier-32 SQuAD endpoint falls to 22.26, so the larger learning rate does not monotonically improve quality. One GSM8K choice reaches this grid boundary. All endpoints and both arms' choices are retained below.

Table 55: **All terminal endpoints and selection.** Mean half-B accuracy/F1 across ten seeds.

Terminal comparator	GSM8K	SQuAD
Terminal $\times 2$	39.49	62.74
Terminal $\times 8$	40.24	68.93
Terminal $\times 16$	41.39	70.13
Terminal $\times 32$	41.07	22.26
Selected terminal	40.46	70.97

Each fixed recipe uses its full training budget; the selected comparator chooses recipe and checkpoint on A. The multiplier-2 recipe uses constant learning rate; multipliers 8, 16, and 32 use linear decay. These summaries are descriptive and do not constitute additional superiority tests.

Table 56: **Effect of terminal selection.** Selected terminal minus the original tuned endpoint on B (pp).

Task	Difference [95% CI]
GSM8K	$+0.21$ $[-2.12, +2.55]$
SQuAD	$+2.04$ $[+1.64, +2.44]$

Paired seed intervals are descriptive; selection can improve or worsen the B score. The original tuned endpoint uses multiplier 8 with linear decay.

Search and execution cost. Across two tasks and ten retained trajectories per task, terminal selection required 80 new full branch-training runs (four recipes per trajectory); intermediate checkpoints yielded 20 candidates per trajectory, rather than 20 independent runs. All 40 reruns of the original two recipes reproduced their retained endpoints, but these reruns still incurred training cost to retain intermediate checkpoints. Secondary P selection used five checkpoints from each retained trajectory: 80 intermediate candidates were assembled on CPU and 20 final endpoints were retained, with no new P training. The 80 terminal runs accumulated 156.77 training-process hours (39.38 on GSM8K; 117.39 on SQuAD). This sums elapsed times around the update loops and checkpoint writes under shared-GPU execution; it is neither exclusive GPU-hours nor campaign wall time. It excludes model/data initialization, evaluation, historical host/P training, CPU assembly, and engineering smoke runs. The completed evaluation inventory contains 629 distinct candidate/half receipts: 509 native evaluations and 120 rescored halves of retained full-holdout predictions. Selecting a previously evaluated endpoint reuses its receipt rather than adding another evaluation. These counts do not include historical prediction-generation cost and do not establish an end-to-end compute advantage.

R Host-Disjoint Branch Data

Question and prospective scope. In every earlier one-block quality study, host and branch train on the same pool and concurrent branches follow the host's sample order. This study removes the shared data. The protocol, code, inputs, data splits, and eight new seeds per task (15700–15707) were frozen before any outcome of this study (lock `aa981cf273d9b1a0`). For each seed the fit+tune pool is grouped by normalized question (GSM8K) or context (SQuAD); groups are randomly assigned to half A until it holds half of the pool, the remainder forms B, and both halves are trimmed to an equal multiple of 8 (3,360 GSM8K examples; 9,992–10,000 SQuAD). A and B share no group and no held-out example. The host trains on A for two epochs with the one-block live P recipe (host rate 10^{-4}, branch multiplier 2, constant schedules, AdamW, zero decay, clipping 1, batch 8).

Descriptive motivation. Earlier shared-stream results are consistent with overfitting but do not identify its cause. Two-stage reaches lower training loss than $\mathrm{ORA}_{\mathrm{post}}$ while scoring lower on holdout (0.113 versus 0.236 and 67.11 versus 77.01 on SQuAD; Appendix E.4). Live H falls from $40.05/71.42$ at host release to $39.12/67.30$ fully trained. These observations motivated the disjoint-data intervention; they are not additional confirmatory tests.

Paired design. In the primary (disjoint) condition the branch trains on B in its own order, so no host version ever fits a branch example. P and terminal-only share branch initialization, batches, order, random stream, optimizer, budget (horizon 840 GSM8K, 2,498–2,500 SQuAD), and final host

Table 57: **Terminal choices on selection half A.** Counts retain all recipes, including unselected ones.

Task	LR multiplier	Seeds	Step: count
GSM8K	2	1	1250: 1
GSM8K	8	3	850: 2, 1250: 1
GSM8K	16	5	850: 1, 1250: 1, 1682: 3
GSM8K	32	1	1682: 1
SQuAD	2	0	–
SQuAD	8	6	2500: 1, 3750: 5
SQuAD	16	4	3750: 2, 5002: 2
SQuAD	32	0	–

Each task has ten choices. Multiplier 32 is the upper search boundary; choosing it leaves further learning-rate sensitivity unresolved. Ties prefer later checkpoints, then larger multipliers. The comparison gives terminal 20 candidates per seed and secondary P five; neither policy is globally optimized.

Table 58: **Secondary P checkpoint choices on A.** Counts include every candidate stopping point.

Task	Step	Seeds
GSM8K	100	0
GSM8K	400	3
GSM8K	850	2
GSM8K	1250	4
GSM8K	1682	1
SQuAD	250	0
SQuAD	1250	0
SQuAD	2500	6
SQuAD	3750	0
SQuAD	5002	4

P selects among five checkpoints with its original learning-rate recipe. These choices belong to the secondary comparison; full-budget P remains the primary comparator.

s_T; P reads $s_{250\lfloor(j-1)/250\rfloor}$ and terminal-only reads s_T. A secondary (shared) condition trains the branch on A in the host's order, the pairing of the main studies, with a recomputed host that must match the disjoint condition's host endpoint bitwise. P is executed serially as in Appendix O.

Engineering and inference. CPU tests check split determinism, equal sizes, group disjointness, holdout exclusion, schedule coverage, and the version sequence. A known one-block gate reproduces the retained live P branch at update 50; smoke runs for both tasks and conditions match the host's loss trace and endpoint to standalone, match the optimized branch step to the reference step, and give identical host endpoints across conditions. Formal runs check protected tensor version counters during branch steps, publication hashes, the terminal endpoint, and the realized P sequence. The primary rule is that of Appendix O: Holm-adjusted exact one-sided sign-flip $p < 0.05$, interval above zero, and mean above $1\,pp$ per task; a joint statement requires both tasks. The secondary contrast and the interaction $(P - \text{terminal})_{\text{shared}} - (P - \text{terminal})_{\text{disjoint}}$ are descriptive, with two-sided exact sign-flip p.

Result. Neither task passes the primary rule. P minus terminal-only is -1.86 $[-3.13, -0.58]$ on GSM8K (Holm $p = 1.0000$, $1/8$ positive) and -1.16 $[-1.54, -0.78]$ on SQuAD (Holm $p = 1.0000$, $0/8$ positive). In the shared condition, P minus terminal-only is $+4.08$ $[+2.99, +5.18]$/$+13.51$ $[+12.31, +14.71]$; the interaction is $+5.94$ $[+4.37, +7.51]$ on GSM8K (two-sided $p = 0.0078$) and $+14.67$ $[+13.38, +15.95]$ on SQuAD ($p = 0.0078$). Host endpoints are bitwise identical across conditions for every seed.

Reproduction. The supplement includes the frozen protocol and code, engineering and training receipts, paired scores, branch traces, and predictions. `disjoint_evidence.py` verifies their bindings, recomputes the statistics from the recorded scores, and generates these tables. Model tensors are omitted from the portable release; verification does not regenerate predictions.

Table 59: **Effect of P checkpoint selection.** Selected P minus full-budget P on B (pp).

Task	Difference [95% CI]
GSM8K	$+0.05$ $[-1.71, +1.82]$
SQuAD	$+0.73$ $[+0.15, +1.31]$

Intervals pair the same ten seeds and are descriptive. A choice on A can improve or worsen the B score; this diagnostic adds no superiority test and does not replace the primary comparison.

Table 60: **All selected-terminal pairs on B.** Scores are accuracy/F1 on a 0–100 scale; differences are in pp. Terminal recipe multipliers and both checkpoint choices are selected only on A.

Task	Seed	P final	T selected	$P - T$	T LR/step	P step	$P_{\text{sel}} - T$
GSM8K	15400	45.04	38.61	+6.43	16/1250	1250	+4.56
GSM8K	15401	46.11	45.58	+0.54	8/1250	400	+1.07
GSM8K	15402	48.79	43.70	+5.09	8/850	1250	+1.88
GSM8K	15403	46.11	40.21	+5.90	16/850	400	+4.29
GSM8K	15404	46.11	41.55	+4.56	8/850	1682	+4.56
GSM8K	15405	43.16	37.53	+5.63	16/1682	1250	+9.12
GSM8K	15406	45.58	40.21	+5.36	16/1682	1250	+2.14
GSM8K	15407	46.11	40.21	+5.90	16/1682	400	+7.77
GSM8K	15408	45.84	38.34	+7.51	2/1250	850	+8.85
GSM8K	15409	44.24	38.61	+5.63	32/1682	850	+8.85
SQuAD	15400	76.57	70.84	+5.72	8/3750	2500	+7.41
SQuAD	15401	76.19	69.68	+6.51	16/5002	5002	+6.51
SQuAD	15402	76.35	71.32	+5.03	16/5002	5002	+5.03
SQuAD	15403	78.08	72.58	+5.50	8/3750	5002	+5.50
SQuAD	15404	76.45	70.41	+6.04	16/3750	2500	+7.09
SQuAD	15405	79.01	71.53	+7.48	8/2500	5002	+7.48
SQuAD	15406	76.28	71.28	+5.01	8/3750	2500	+6.02
SQuAD	15407	76.94	71.59	+5.35	8/3750	2500	+6.88
SQuAD	15408	78.01	70.21	+7.80	8/3750	2500	+7.83
SQuAD	15409	76.42	70.30	+6.12	16/3750	2500	+8.10

S Separating Data Sharing from Order Alignment

Separating branch data from order. The overlap/order extension adds a shared-data, independent-order condition to the retained eight-seed data-separation design. The host keeps its original training half and order; the branch uses the same half with the independent permutation generator also used by the disjoint condition. Relative to shared-aligned training, the intervention changes branch order and batch composition. Relative to disjoint-independent training, it changes branch data half while retaining the permutation rule, sample count, update budget, recipe, initialization, randomness, and deployment host. The design uses one-block Qwen2.5-0.5B, rank-8 LoRA, batch size 8, two epochs, host/branch learning rates $10^{-4}/(2 \times 10^{-4})$, constant schedules, and P cadence 250. Each new host endpoint and complete recorded host loss trace must match the retained reference before evaluation. Evaluation uses the fixed greedy scorer on all 747 GSM8K and 5,002 SQuAD holdout examples.

Conditional contrasts. Let D_a, D_i, and D_d be paired P-minus-terminal differences for shared-aligned, shared-independent, and disjoint-independent conditions. The two contrasts are $D_a - D_i$ and $D_i - D_d$, whose sum is $D_a - D_d$. Their four task-by-contrast tests use two-sided exact sign flips and one Holm family, with paired 95% intervals and individual seed results. The three condition summaries are descriptive. These contrasts locate dependence under the tested stream construction; they do not isolate a unique mediator or constitute a full overlap-by-order factorial. A nonsignificant contrast does not establish equivalence. The extension reuses seeds and historical partitions from the earlier study and is not independent fresh-seed confirmation.

Order and data contribute distinct conditional contrasts. A locked post-E5 extension adds the missing shared-data, independent-order condition on the same eight seeds per task. On the same final host, the $P-$terminal difference becomes $+0.64\,pp$ on GSM8K (paired 95% interval $[-1.02, +2.29]$) and $+8.48\,pp$ on SQuAD ($[+6.18, +10.77]$). Relative to shared-aligned training, changing branch order and batch composition reduces the difference by $3.45/5.03\,pp$. At independent ordering, changing from shared to disjoint branch data reduces it by a further $2.49/9.63\,pp$. All four prespecified two-sided contrasts pass Holm correction as one family ($p = 0.03125$ each). Thus order alignment contributes on both tasks, but does not alone explain the SQuAD shared-data advantage.

Table 61: **All host-disjoint and shared-data pairs.** Final fused accuracy/F1 (0–100); P minus terminal-only in pp; mean branch training loss over the first and last fifth of updates (descriptive).

Task	Data	Seed	P	Terminal	Diff.	P loss first/last	Terminal loss first/last
GSM8K	disjoint	15700	44.04	45.38	−1.34	0.510/0.431	0.453/0.422
GSM8K	disjoint	15701	46.59	47.39	−0.80	0.513/0.438	0.455/0.429
GSM8K	disjoint	15702	44.58	47.12	−2.54	0.510/0.430	0.449/0.419
GSM8K	disjoint	15703	45.65	46.59	−0.94	0.513/0.436	0.457/0.427
GSM8K	disjoint	15704	46.32	46.05	+0.27	0.505/0.441	0.449/0.430
GSM8K	disjoint	15705	43.51	47.12	−3.61	0.516/0.432	0.458/0.423
GSM8K	disjoint	15706	44.04	48.33	−4.28	0.517/0.432	0.455/0.422
GSM8K	disjoint	15707	46.32	47.93	−1.61	0.507/0.432	0.448/0.424
SQuAD	disjoint	15700	78.18	79.37	−1.20	0.554/0.343	0.443/0.324
SQuAD	disjoint	15701	78.73	79.09	−0.36	0.573/0.330	0.446/0.314
SQuAD	disjoint	15702	77.79	79.02	−1.23	0.552/0.337	0.448/0.322
SQuAD	disjoint	15703	78.42	79.79	−1.37	0.577/0.318	0.462/0.298
SQuAD	disjoint	15704	78.22	79.60	−1.38	0.554/0.338	0.446/0.319
SQuAD	disjoint	15705	77.71	78.41	−0.70	0.565/0.329	0.466/0.311
SQuAD	disjoint	15706	78.05	79.22	−1.17	0.574/0.344	0.460/0.327
SQuAD	disjoint	15707	76.07	77.93	−1.86	0.552/0.339	0.444/0.319
GSM8K	shared	15700	42.57	39.49	+3.08	0.514/0.376	0.300/0.262
GSM8K	shared	15701	43.64	38.82	+4.82	0.516/0.366	0.300/0.252
GSM8K	shared	15702	43.51	40.16	+3.35	0.516/0.367	0.299/0.254
GSM8K	shared	15703	44.44	38.55	+5.89	0.513/0.371	0.297/0.253
GSM8K	shared	15704	42.30	38.69	+3.61	0.509/0.370	0.293/0.256
GSM8K	shared	15705	42.17	40.16	+2.01	0.513/0.371	0.295/0.257
GSM8K	shared	15706	44.04	38.55	+5.49	0.516/0.378	0.300/0.260
GSM8K	shared	15707	45.52	41.10	+4.42	0.519/0.363	0.304/0.251
SQuAD	shared	15700	75.32	61.92	+13.40	0.546/0.261	0.159/0.077
SQuAD	shared	15701	77.53	63.00	+14.53	0.556/0.259	0.143/0.069
SQuAD	shared	15702	75.44	64.51	+10.93	0.573/0.266	0.153/0.074
SQuAD	shared	15703	77.03	62.98	+14.05	0.544/0.269	0.148/0.075
SQuAD	shared	15704	75.97	60.28	+15.69	0.547/0.257	0.137/0.073
SQuAD	shared	15705	77.07	64.29	+12.78	0.550/0.256	0.149/0.076
SQuAD	shared	15706	76.10	63.47	+12.62	0.548/0.254	0.145/0.075
SQuAD	shared	15707	76.42	62.33	+14.09	0.543/0.253	0.158/0.070

Table 62: **Observation differences across data/order conditions.** $P-$terminal-only in pp with paired 95% intervals, eight matched seeds per task. These condition summaries are descriptive.

Condition	GSM8K	SQuAD
Shared, aligned	$+4.08$ $[+2.99, +5.18]$	$+13.51$ $[+12.31, +14.71]$
Shared, independent	$+0.64$ $[-1.02, +2.29]$	$+8.48$ $[+6.18, +10.77]$
Disjoint, independent	-1.86 $[-3.13, -0.58]$	-1.16 $[-1.54, -0.78]$

Aligned and disjoint rows reuse the completed equal-size E5 study. The new shared-independent condition keeps the same host trajectory and data half while changing branch order. All conditions have the same branch sample count and update budget.

The GSM8K shared-independent interval includes zero; this does not establish equivalence. These retained-seed, historical-data contrasts identify dependence on the tested stream construction, not a unique latent mechanism or a full overlap-by-order factorial.

T Recovery Validation and Additional Observation Policies

T.1 Matched failure and repair

Task metrics witness recoverability failure rather than define recovery. For metric M at matched epoch e, $\mathrm{RG}_e = M(A_e^0) - M(A_e^{\mathrm{off}})$ compares standalone and branch-disabled artifacts. Either sign of a nonzero gap demonstrates different behavior; a zero gap cannot establish state identity.

Table 63: **Contrasts between data/order conditions.** Differences of $P-$terminal effects in pp with paired 95% intervals; two-sided exact tests, Holm correction over all four contrasts.

Contrast	Task	Difference [95% CI]	Holm p
Aligned minus independent	GSM8K	$+3.45$ $[+1.71, +5.18]$	0.0312
Aligned minus independent	SQuAD	$+5.03$ $[+2.03, +8.04]$	0.0312
Shared minus disjoint	GSM8K	$+2.49$ $[+1.01, +3.98]$	0.0312
Shared minus disjoint	SQuAD	$+9.63$ $[+7.25, +12.02]$	0.0312

The order contrast is shared-aligned minus shared-independent. The data contrast is shared-independent minus disjoint-independent. Their sum equals aligned minus disjoint. This decomposition is conditional on the tested stream construction; it does not identify a unique mediator or replace a full data-overlap-by-order factorial.

Table 64: All paired condition differences and contrasts (pp). Aligned and disjoint values are inherited from E5; independent values are the new condition.

Task	Seed	D_a	D_i	D_d	$D_a - D_i$	$D_i - D_d$
GSM8K	15700	$+3.08$	$+1.47$	-1.34	$+1.61$	$+2.81$
GSM8K	15701	$+4.82$	$+4.69$	-0.80	$+0.13$	$+5.49$
GSM8K	15702	$+3.35$	-2.01	-2.54	$+5.35$	$+0.54$
GSM8K	15703	$+5.89$	$+1.07$	-0.94	$+4.82$	$+2.01$
GSM8K	15704	$+3.61$	$+0.94$	$+0.27$	$+2.68$	$+0.67$
GSM8K	15705	$+2.01$	-0.13	-3.61	$+2.14$	$+3.48$
GSM8K	15706	$+5.49$	-0.27	-4.28	$+5.76$	$+4.02$
GSM8K	15707	$+4.42$	-0.67	-1.61	$+5.09$	$+0.94$
SQuAD	15700	$+13.40$	$+11.31$	-1.20	$+2.09$	$+12.50$
SQuAD	15701	$+14.53$	$+5.23$	-0.36	$+9.30$	$+5.59$
SQuAD	15702	$+10.93$	$+11.34$	-1.23	-0.42	$+12.57$
SQuAD	15703	$+14.05$	$+3.97$	-1.37	$+10.08$	$+5.34$
SQuAD	15704	$+15.69$	$+10.15$	-1.38	$+5.54$	$+11.53$
SQuAD	15705	$+12.78$	$+8.77$	-0.70	$+4.01$	$+9.47$
SQuAD	15706	$+12.62$	$+9.67$	-1.17	$+2.96$	$+10.84$
SQuAD	15707	$+14.09$	$+7.37$	-1.86	$+6.72$	$+9.23$

The failure is measurable in matched BERT-base (Devlin et al., 2019) early-exit training on QNLI and SST-2 (Wang et al., 2019). Across 20 seeds, plain and PABEE-style joint training (Zhou et al., 2020) share protected initialization, data order, and recipe. Disabling every exit after five epochs leaves positive gaps in every pair: $1.431/1.565\,pp$, paired 95% intervals $[1.304, 1.559]/[1.361, 1.770]$, Holm $p = 1.91\times10^{-6}$ (Table 11). Under the same conditions, $\mathrm{ORA}_{\mathrm{post}}$ matches plain training at all five recorded epoch endpoints for every seed and task, closing the failure–repair comparison at those boundaries.

Qwen optional-LoRA training shows why enabled-route quality cannot certify the fallback: joint training improves the enabled route while lowering disabled GSM8K/SQuAD scores by $2.33/11.45\,pp$ (descriptive; Figure 4 and Appendix E.4).

T.2 State completeness and conditional necessity

The state in Equation (1) is complete relative to the execution environment. The main stack uses eager FP32 on one device; BF16 is tested over eight steps and multi-device only in a synthetic DDP secondary. Sufficiency depends on the stack, some fields can be vacuous, and Proposition 6 gives a model in which each field is necessary. Restoring a checkpoint must recover the same transition-relevant information. If optional work changes an input that the protected transition distinguishes, uniform recovery fails (Lemma 3 and Proposition 4); this converse rests on a sensitivity premise unverified for clipped AdamW.

T.3 Opening a write channel predicts the first divergence

The audit tests a prediction that can fail: opening a particular write channel must produce the declared first divergence in the live fields of Equation (1); a later mismatch cannot rescue an incorrect first-component prediction, and isolated positives must remain exact (Table 28).

All 110 synthetic linear–BatchNorm tests (11 cells, five seeds, two roles) satisfy their declared positive or negative prediction; a two-rank DDP secondary adds 40 passing executions (Table 29). Qwen2.5-0.5B tests five seeds over eight protected steps: isolated FP32, BF16, and complete restoration are exact in $15/15$ cases. Global clipping, optimizer reuse, RNG leakage, cursor leakage, and incomplete restoration produce the predicted first divergence in $25/25$ (Table 25). BF16 evidence is restricted to this eight-step horizon. These matched interventions locate optional influence that gradient isolation alone misses; longer traces and lifecycle studies test preservation beyond the intervention horizon.

T.4 Preservation extends through complete recorded trajectories

Full-horizon checks compare standalone and $\mathrm{ORA}_{\text{post}}$ after protected and branch updates: on Qwen2.5-0.5B, three fresh seeds per task match all seven live recorded components and 20,052 protected losses at 40,110 boundaries (eager FP32 makes precision/scaler state vacuous), and Qwen2.5-1.5B/rank-32 GSM8K checks match 10,095 boundaries over three fresh runs; branch quality at that scale is unmeasured.

The larger quality matrices check full-horizon loss digests and endpoint state rather than every component after every update: $\mathrm{ORA}_{\text{post}}$ and two-stage match protected losses on all 20 fresh seeds per task, whereas PCGrad differs by protected step three. We distinguish endpoint hashes, loss traces, and complete recorded-state traces throughout (Appendix C).

T.5 Abandonment and optional-data contamination

Abandonment and restoration. Midpoint removal and fresh-process restoration preserve the tested continuation (Tables 17 and 18): removing $\mathrm{ORA}_{\text{post}}$ at GSM8K step $841/1{,}682$ retains recorded states, losses, and 747 predictions in all three seeds while joint differs; BERT-base/SST-2 matches 150 boundaries and 3×872 predictions.

Contaminated optional data. Across 10 GSM8K and 6 SQuAD seeds, enabled branches learn 64 canaries. Under $\mathrm{ORA}_{\text{post}}$, protected digests, endpoint components, parameter hash, and disabled scores equal clean standalone in all 16 seeds; joint training retains $5.22/4.58$ bits of disabled excess exposure, intervals $[4.97, 5.47]/[3.64, 5.52]$ (Figure 5 and Appendix E.6).

T.6 Serial and checkpoint observation rules

$\mathrm{ORA}_{\text{pre}}$ reads the boundary before the protected update and changes only u:

$$u_{t+1}^{\text{pre}} = T_{\text{br}}^{\text{pre}}(u_t; \text{readonly}(s_t), z_t), \qquad s_{t+1}^{\text{pre}} = T_0(s_t; x_t). \tag{6}$$

$\mathrm{ORA}_{\text{post}}$ first performs the protected update, then exposes its result:

$$\bar{s}_{t+1} = T_0(s_t; x_t), \qquad u_{t+1}^{\text{post}} = T_{\text{br}}^{\text{post}}(u_t; \text{readonly}(\bar{s}_{t+1}), z_t), \tag{7}$$

$$s_{t+1}^{\text{post}} = \bar{s}_{t+1}. \tag{8}$$

$\mathrm{ORA}_{\text{ckpt}}$ reads the latest state published after a multiple of K protected transitions:

$$u_{t+1}^{K} = T_{\text{br}}^{K}(u_t; \text{readonly}(s_{\kappa(t)}), z_t), \quad \kappa(t) = K\lfloor (t+1)/K \rfloor, \quad s_{t+1}^{K} = T_0(s_t; x_t). \tag{9}$$

Never publication keeps the initial view s_0; $K = 1$ can reproduce post. P uses the same publication interval as $\mathrm{ORA}_{\text{ckpt}}$, but assigns versions before reader updates; their publication-boundary indexing differs. Main policy comparisons fuse with s_T, including when training used only s_0. A separate control evaluates fixed branches on both bases.

Batch selection is part of the training protocol. Serial pre/post each use protected batch x_t for one branch update, observing the host before/after it fits that batch; concurrent H/P follow protected sample order by reader index, which can pair a batch with a different writer step. Independently permuting branch batches changes these inputs without changing recovery (Section 4.4).

Table 65: **Observation policies and deployment hosts.** Terminal accuracy/F1 (0–100), mean ± seed SD, $n = 10$. Cross-study comparisons are descriptive; matched effects are in Table 1. Exec.: serial, one process; concurrent, separate branch process; after, branch training follows host completion. Tuned rows use the recipe selected for two-stage on tune data. Branchless: $3.21/37.42$ at s_0, $37.03/55.50$ at s_T.

Policy	Branch observes	Deployed on	Exec.	Study	GSM8K	SQuAD
Frozen base (never)	s_0	s_0	—	ctrl	50.13 ±1.3	80.59 ±0.4
Clean plus joint (C)	joint host	joint host	two hosts	live	45.53 ±1.4	78.14 ±0.4
C branch on clean s_T	joint host	s_T	—	match	44.03 ±1.8	69.92 ±4.3
ORA_{pre}	s_t	s_T	serial	ctrl	44.42 ±2.0	78.37 ±0.6
ORA_{ckpt}, K=250	$s_{\kappa(t)}$	s_T	serial	ctrl	43.63 ±1.6	77.30 ±1.2
ORA_{post}	s_{t+1}	s_T	serial	ctrl	43.48 ±0.9	76.55 ±0.7
P: reader progress	s_{σ_j}	s_T	concurrent	live	43.11 ±1.2	77.88 ±0.8
H: latest version	$\geq 52\%$ on s_T	s_T	concurrent	live	39.12 ±1.1	67.30 ±2.0
Two-stage (tuned)	s_T only	s_T	after	ctrl	39.75 ±2.1	71.88 ±0.4
Two-stage, P's order	s_T only	s_T	after	match	38.43 ±1.5	64.01 ±1.6
Two-stage, P's order (tuned)	s_T only	s_T	after	match	38.80 ±1.3	70.13 ±0.6
Frozen branch moved to s_T	s_0	s_T	—	ctrl	37.70 ±3.1	64.42 ±11.1

Studies: ctrl, serial policy and deployment-host controls (seeds 15300–15309); live, concurrent confirmation; match, matched terminal-only and transfer follow-up (both 15400–15409; Table 10). The ctrl two-stage row uses an independent order; match rows use P's order.

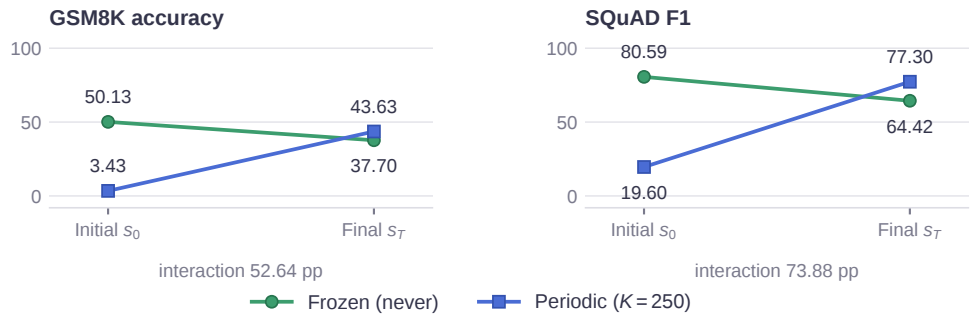

Figure 7: **Deployment host changes the policy ranking.** Trained branch weights are held fixed and only the host they are attached to changes; means over 10 retained seeds, one protected block. The frozen (never-published) branch wins on its training host s_0; the periodic K=250 branch wins on the continuing host's final s_T; the branchless initial host scores $3.21/37.42$. Interaction: $(K250@s_T - K250@s_0) - (\text{never}@s_T - \text{never}@s_0)$. Paired contrasts are in Tables 34 and 35.

U Policy Landscape and Deployment-Host Control

V Exact-Match and Output-Length Diagnostics

These post-outcome diagnostics use all 5,002 SQuAD holdout predictions from each retained branch, with training seed as the unit. They check whether the F1 comparison also appears in exact match (EM), and describe output length. They do not establish robustness to a different prompting template or identify the cause of an error. The machine-readable diagnostics record all per-seed means, empty-answer rates, answer word counts, and prediction digests. The same-recipe and tuned-recipe comparisons share the same P artifacts; they are not independent replications.

Table 66: **SQuAD diagnostics on the same predictions.** Differences are P minus terminal-only, in pp with paired 95% intervals. Lengths are mean generated tokens, including terminal tokens, averaged over seeds; P/terminal. No new significance test is performed.

Comparison	F1 difference	EM difference	Tokens P/terminal
matched same recipe	$+13.87$ $[+12.99, +14.74]$	$+12.37$ $[+11.34, +13.40]$	$5.86/5.97$
matched tuned recipe	$+7.75$ $[+7.24, +8.26]$	$+6.84$ $[+5.81, +7.86]$	$5.86/6.01$
E5 shared	$+13.51$ $[+12.31, +14.71]$	$+11.25$ $[+9.82, +12.67]$	$6.37/6.22$
E5 disjoint	-1.16 $[-1.54, -0.78]$	-2.33 $[-3.28, -1.39]$	$6.28/5.85$