THE DURABLE ASSET IS THE DATA, NOT THE WEIGHTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Organizations increasingly serve fleets of fine-tuned specialists on open-weight language models whose bases improve every few months. Each release reopens the same decision: *freeze* the old specialist, *port* its weights, *refresh* it from retained behavior, or *retrain*. These strategies form a ladder of retained resources—nothing, weights, inputs, labels—yet the decision has never been measured: prior work evaluates ad-hoc model pairs, never a real release sequence. We measure it longitudinally, on a complete fourteen-month release lineage (four consecutive generations, two scales, six tasks spanning classification, structured generation, and tool use), validated on a second family with fully documented training genealogy; hypotheses are fixed before each experimental wave, and every comparison is paired, per-example, and cost-accounted. We find that (1) *specialization is durable*: a frozen specialist can outperform the newest base for over a year, governed by a measurable task–base coupling rather than task class; (2) *weights are fragile*: porting between architecturally identical checkpoints from independent pretraining runs collapses below the no-adapter floor, yet porting onto a short continued-pretraining descendant is nearly free—portability is a distance budget in pretraining tokens, not an architecture or family property; and (3) *data recovers what weights cannot carry*: distilling the old specialist into the new base, using retained inputs only, matches gold-label retraining with zero new annotation. A pre-specified policy built on these facts nearly matches always-retraining at a third of its cost with zero regressions, while the shape-compatibility rule practitioners use is the worst policy we test. All code, records, and cost logs are released; everything runs on one consumer GPU.

1 INTRODUCTION

Two facts about deployed language models are now well established. First, *specialization works*: on narrow, recurrent tasks—intent triage, text-to-SQL, function calling—fine-tuned small models match or beat prompted frontier models at one to two orders of magnitude lower inference cost (Zhao et al., 2024; Bucher & Martini, 2024; Belcak et al., 2025), and multi-adapter serving makes a fleet of specialists cheap to host (Sheng et al., 2024). Second, *base models improve fast*: the capability density of open-weight models doubles on the order of months (Xiao et al., 2025), so the model underneath every specialist is a depreciating substrate.

These facts collide because a fine-tune is bound, both numerically and behaviorally, to the base checkpoint it was trained on. When a better base ships, the organization faces a recurring decision it currently cannot price. Concretely, suppose an intent classifier was fine-tuned as a LoRA adapter on Qwen2, and Qwen2.5 is then released. The options form a *resource ladder*—each rung requires strictly more retained resources than the one before:

- FREEZE — keep serving the Qwen2 specialist unchanged and ignore the release; requires nothing.
- PORT — attach the adapter file trained on Qwen2 to Qwen2.5 instead, with no training at all; requires the old *weights* (and an architecture that accepts them).
- REFRESH — distill the old specialist into the new base: the old model (teacher) relabels the stored task inputs, and a fresh adapter (student) is fine-tuned on Qwen2.5 from those labels; requires the old *inputs*, no gold labels.

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.

- RETRAIN — fine-tune Qwen2.5 on the original gold-labeled dataset, exactly as the first specialist was trained; requires the old *labels* (or new annotation).

We call the quality-and-cost gap between the chosen strategy and the best attainable one the *upgrade tax*. A growing family of methods transfers fine-tuning across bases (Wang et al., 2024; Khan et al., 2025; Farhadzadeh et al., 2025; Li et al., 2025; Lin et al., 2025; Gu et al., 2025), but each is evaluated on its own model pairs, cross-paper comparison is impossible, and the most decision-relevant quantities have never been measured: how long does a specialist's advantage survive real base-model progress, and what does each rung actually recover?

This paper supplies the missing measurement, and the measurement overturns three default beliefs. *Fine-tunes do not become obsolete quickly*: across nine upgrade hops, freezing retains 99–101% of the attainable gain on intent classification, and after fourteen months the frozen specialist still leads the newest base by 28.6 points. *But they are not uniformly safe*: on text-to-SQL the frozen specialist forfeits up to 59% of attainable gain in a single hop, because new-base reasoning penetrates the fine-tune. *And naive weight porting is governed by neither architecture nor family*: between architecturally identical checkpoints from independent pretraining runs, pasting the adapter onto the newer base collapses intent accuracy from 92.8% to 42.9%—below the 60.7% the bare base achieves—yet the same operation onto a continued-pretraining descendant of the same weights retains reference-level accuracy. An ablation on OLMo, whose checkpoints make the training relationship an experimental variable, locates the boundary: retention falls from 0.88–0.99 at a 46B-token continuation to the no-adapter floor at 2.9T tokens, with annealing and souping adding no further damage. Portability is a *distance budget* in continued-pretraining tokens.

Together the results support a thesis that reframes the staleness debate around specialist fleets (Belcak et al., 2025): **the durable asset is the data, not the weights**. The weight artifact expires with its pretraining trajectory, but the specialization itself—latent in retained inputs, gold labels, and the old specialist's behavior—survives every upgrade and can be re-instantiated at parity on any new base. Our contributions:

C1 (Benchmark). A longitudinal, cost-accounted benchmark of fine-tuning value and portability along a complete open-weight release lineage (four generations, fourteen months, two scales, six tasks), with unified metrics, per-query records, bootstrap CIs, and energy logs, all released, on one RTX 4090. **C2 (Quantities).** Formal definitions and first measurements of retention, durability horizons (H_{50}, H_0), task–base coupling β, and recovery cost (§5.1–5.2). **C3 (Continuity criterion).** A twice-replicated dichotomy with a measured boundary: porting fails between independent pretraining runs (eleven task/scale cells, two families), succeeds across *short* continued pretraining (five cells, two families), and decays with continuation distance (§5.3); annotation-free refresh reaches retraining parity everywhere tested (§5.4). **C4 (Decision policy).** A pre-specified policy over the ladder, replayed on 33 measured episodes with gate/score splits: 0.37pp mean regret, zero regressions, one third of the always-retrain budget; the shape-compatibility rule practitioners use is the worst policy tested (§5.6).

2 RELATED WORK

Cross-base transfer of fine-tuning. Trans-LoRA (Wang et al., 2024) first framed base deprecation, proposing near-data-free transfer by distillation; TiTok (Jung & Kim, 2026) and TuneShift-KD (Guan et al., 2026) refine that route. LoRASuite (Li et al., 2025) and PortLLM (Khan et al., 2025) build upgrade-specific parameter mappings; LoRA-X (Farhadzadeh et al., 2025) and Cross-LoRA (Xia et al., 2025) project adapters through aligned subspaces; Trans-PEFT (Gu et al., 2025) immunizes adapters at training time. Each evaluates on self-selected pairs; a recurring assumption is that weights copied unchanged between similar bases are a strong baseline—our measurements on architecturally identical consecutive releases contradict it (§5.3). We propose no new method; we provide the missing common yardstick over a real release sequence, on which two of three hops admit no weight-space method at all.

Task vectors and merging. Task arithmetic treats fine-tuning deltas as portable (Ilharco et al., 2023), with theory tied to shared initialization (Ortiz-Jiménez et al., 2023); Chat Vector (Huang et al., 2024) and fine-tuning transfer (Lin et al., 2025) demonstrate near-free porting *within* a lineage. Our paired

Table 1: Terminology and notation used throughout. All scores lie in $[0, 1]$; all horizons are in months of real release time.

Symbol	Name	Meaning
$\mathcal{B} = (b_0, \ldots, b_G)$	lineage	base checkpoints ordered by public release date
$b_i \to b_j$	hop	one upgrade step within a lineage
$D_t = (D_X, D_Y)$	task data	inputs and gold labels for task t
$\mathcal{A}(b, D)$	recipe	fixed parameter-efficient fine-tuning (PEFT) procedure
$s_{b,t}$	specialist	$\mathcal{A}(b, D_t)$, fine-tuned on base b
$F_t(b)$	adoption floor	score of b with prompting only (zero-/few-shot)
$O_t(b)$	reference	score of the fully retrained specialist $s_{b,t}$
$G(b, t)$	gain	$O_t(b) - F_t(b)$, what fine-tuning buys on b
$R^m_{i\to j}(t)$	retention	fraction of the target's gain recovered by strategy m (Eq. 5)
L^m	shortfall	signed quality gap to the reference, in points
$A_t(\tau)$	advantage	frozen specialist's lead over the floor at calendar time τ
H_{50}, H_0	horizons	first time A_t halves; first time it reaches zero
β_t	coupling	$\Delta O_t / \Delta F_t$: reference gain per point of floor gain
C_{95}	recovery cost	smallest label budget reaching 95% of the gain
$d(b_i, b_j)$	distance	continued-pretraining tokens separating two checkpoints

hops turn shared initialization into an experimental variable: transfer survives a continued-pretraining hop and dies across independent runs of identical architecture, as the tangent-space account predicts.

Model updates and regressions. Replacing a deployed model regresses individual predictions even when aggregate accuracy improves (Yan et al., 2021; Cai et al., 2022); updates also silently break prompt-engineered applications (Chen et al., 2023a; Tripathi et al., 2025). We fold negative flips into the upgrade decision itself (§5.5).

Cost-aware evaluation and specialist economics. Fine-tuned small models beat prompted frontier models on narrow tasks (Zhao et al., 2024; Bucher & Martini, 2024); dollar-denominated evaluation is established (Chen et al., 2023b; Erol et al., 2025; Kapoor et al., 2024); the upgrade tax is the missing depreciation term. To our knowledge, no existing benchmark tracks its models along release time; ours makes the release sequence the primary axis.

3 The Upgrade Decision

Setup. A *lineage* is a sequence of base checkpoints $\mathcal{B} = (b_0, \ldots, b_G)$ ordered by release date (Table 1). For task t with data $D_t = (D_X, D_Y)$ and metric $S_t(\cdot) \in [0, 1]$, a fixed parameter-efficient procedure $\mathcal{A}$ produces the specialist $s_{b,t} = \mathcal{A}(b, D_t)$. The *adoption floor* $F_t(b)$ is the base's prompting-only score; the *reference* $O_t(b) = S_t(s_{b,t})$ is what full retraining delivers.

The four strategies. On a hop $b_i \to b_j$, a strategy m maps the old specialist and the new base to a served system $m(s_{b_i,t},\, b_j)$:

$$\textsc{Freeze}: \quad m = s_{b_i,t} \qquad \text{(requires nothing)} \tag{1}$$

$$\textsc{Port}: \quad m = b_j \oplus \Delta W_{i,t} \qquad \text{(requires old weights; defined iff shapes match)} \tag{2}$$

$$\textsc{Refresh}: \quad m = \mathcal{A}(b_j,\, \{(x,\, s_{b_i,t}(x)) : x \in D_X\}) \qquad \text{(requires old inputs)} \tag{3}$$

$$\textsc{Retrain}: \quad m = \mathcal{A}(b_j,\, D_t) \qquad \text{(requires old gold labels)} \tag{4}$$

Here $\Delta W_{i,t}$ is the fine-tuning delta of $s_{b_i,t}$; for a LoRA adapter, loading the adapter and adding its merged $\Delta W = BA$ are the same operation, so PORT covers adapter copying and diff-vector porting (Lin et al., 2025) at once. The definition is method-agnostic—any fine-tuning whose artifact is a parameter delta, including the full-fine-tuning task vector (Ilharco et al., 2023), ports the same way—though our measurements instantiate it with LoRA adapters (§6). REFRESH is knowledge distillation (Hinton et al., 2015) in its hard-label, retained-input form: the old specialist is the teacher, and the new base hosts the student. ADOPT ($m = b_j$, prompting only) is the floor—the "no-adapter" score—against which everything is measured. The ladder in Eqs. 1–4 is ordered by retained resources—nothing, weights, inputs, labels—which is why the decision is nontrivial: the

Table 2: The measured lineage, read from each checkpoint's `config.json`. Shape compatibility governs whether weight-space transfer is *defined* at all; exactly one hop qualifies. The continuation release is shape-identical to Qwen2.5 *and* a continued-pretraining descendant of its weights, isolating continuity from compatibility.

Checkpoint	Released	Hidden	Layers	Heads (Q/KV)	FFN	Hop to next
Qwen1.5-7B-Chat	2024-02	4096	32	32/32 (MHA)	11008	incompatible
Qwen2-7B-Instruct	2024-06	3584	28	28/4 (GQA)	18944	**compatible**
Qwen2.5-7B-Instruct	2024-09	3584	28	28/4 (GQA)	18944	incompatible
Qwen3-8B	2025-04	4096	36	32/8 (GQA)	12288	—

Continuation release: Qwen2.5-7B-Instruct-1M (2025-01), shape-identical to Qwen2.5 and a continued-pretraining descendant of its weights. *Small-scale track:* Qwen1.5-1.8B → 2-1.5B → 2.5-1.5B → 3-1.7B; the 2→2.5 hop is likewise shape-identical.

options are not interchangeable methods but what is *available* given what the organization kept. With gold labels in hand, RETRAIN dominates REFRESH (§5.4); the lower rungs exist for organizations that lack them.

Retention and the upgrade tax. For hop $b_i \to b_j$ and strategy m,

$$R^m_{i\to j}(t) \;=\; \frac{S_t(m(s_{b_i,t}, b_j)) - F_t(b_j)}{O_t(b_j) - F_t(b_j)}. \tag{5}$$

$R{=}1$ matches full retraining; $R{=}0$ adds nothing over cheap adoption; $R{<}0$ is worse than not transferring at all; $R{>}1$ beats retraining (both extremes occur in our data). The *upgrade tax* of m is $T^m = (1 - R^m)\,(O_t(b_j) - F_t(b_j))$, paired with m's measured cost. T^m simplifies to the signed shortfall $L^m = O - Q^m$, where Q^m is the strategy's score; we report L^m with paired bootstrap CIs as the primary quantity; the normalized R^m is shown only where the reference gain's lower confidence bound exceeds 2pp. The full tax of an episode is a vector—shortfall, migration GPU-minutes and energy, gold labels, teacher queries—and we keep it one: scalarization needs organization-specific utility weights, so we report Pareto components for the policy (§5.6) to combine.

Durability. The advantage of a generation-i specialist over a generation-$(i{+}g)$ base at calendar time τ is $A_t(\tau) = S_t(s_{b_i,t}) - F_t(b_{i+g})$. We report two horizons: the half-value horizon $H_{50} = \inf\{\tau : A_t(\tau) \le \frac{1}{2}A_t(0)\}$ and the crossover $H_0 = \inf\{\tau : A_t(\tau) \le 0\}$, both interval-censored by the discrete release calendar. They differ by an order of magnitude on some tasks (§5.1); a single "half-life" number would conflate them.

Coupling. We summarize how much a task's ceiling tracks base progress with the elasticity $\beta_t = \Delta O_t / \Delta F_t$ over a fixed hop window: $\beta \approx 0$ means the fine-tune is the ceiling (*data-bound*); β large means base progress penetrates the fine-tune (*base-bound*). §5.2 shows β is a property of a task–scale–release window, not of a task.

4 BENCHMARK DESIGN

Primary lineage. We instantiate the benchmark on the Qwen 7–8B series (Qwen Team, 2024), covering every major generation from February 2024 to April 2025 (Table 2), with a parallel 1.5–1.8B track replicating the whole design at small scale. Architecture churn is the norm: two of three hops admit no weight-space transfer at all. *Applicability is itself an outcome*: transfer papers that evaluate only shape-compatible pairs describe a regime two-thirds of this lineage's real upgrades never entered.

Ground-truth genealogy (OLMo). Qwen's training history is inferred from release notes. To test the continuity criterion where lineage is ground truth, a prospectively specified wave uses OLMo (Groeneveld et al., 2024; Team OLMo et al., 2025), whose checkpoints and training stages are documented. It contributes five edges between architecturally identical checkpoints: (i) OLMo-1 → OLMo-1.7, a verified fresh-pretraining pair; and, holding the source fixed at an OLMo-2 stage-1 checkpoint (∼1T tokens in), targets at (ii) a 46B-token continuation, (iii) the end of stage 1 (∼2.9T

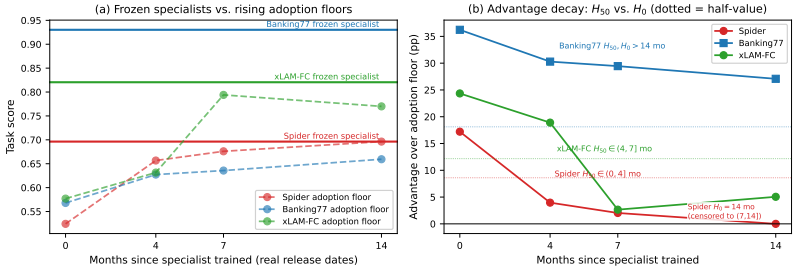

Figure 1: Durability in real release time (n = 3,080 / 1,034 / 2,000). **(a)** Frozen generation-0 specialists vs. each task's rising adoption floor. **(b)** The same data as advantage over the floor, with H_{50} and H_0 marked; horizons are interval-censored by the release calendar.

tokens later, before any annealing), (iv) the post-anneal checkpoint, and (v) the released model-soup merge. Edges (iii)–(v) are distance-matched, separating continuation distance from the release operations. All OLMo hypotheses were fixed before any OLMo run (see the prospective-specification protocol below and App. A).

Tasks. Six tasks, two per class: intent classification (Banking77 (Casanueva et al., 2020), CLINC150 (Larson et al., 2019)), structured generation (Spider text-to-SQL (Yu et al., 2018), FinQA numerical programs (Chen et al., 2021)), and function calling (xLAM-FC (Liu et al., 2024), glaive-FC). Splits are fixed and released; a memorization probe finds no differential contamination across generations (App. G); a leakage audit led us to re-split glaive-FC by template group (App. B).

Protocol. One recipe for all trainings: QLoRA (Dettmers et al., 2023) at rank 16 on all attention/MLP projections, fixed hyperparameters and seed (App. C). Every training logs wall-clock, peak VRAM, and integrated board energy. All comparisons are paired on identical test sets and tested with exact McNemar on discordant pairs; bootstrap 95% CIs (10,000 resamples) accompany every cell. Multi-seed probes bound run-to-run variance ($\sigma \approx 0.12$pp on classification; up to 2.5pp on generative references), which informs how single-hop differences are interpreted. Inventory: 98 trainings, 193 evaluation cells, 148.7 training GPU-hours (47.8 kWh measured) on one RTX 4090; the coverage matrix is sparse by design and released, with empty cells stated rather than implied.

Prospective specification. Hypotheses were fixed in the released runner scripts before the wave that tests them (no external registry, so *prospectively specified* rather than pre-registered). Refuted predictions are reported as refuted: three of our five headline hypotheses were, including the shape-compatibility prediction $R \geq 0.8$ for porting. Appendix A lists every prediction and outcome per wave, including the wave-5 ablation that overturned our own earlier attribution of the OLMo collapse to annealing.

5 Results

5.1 Specialization is durable—when the coupling is low

Across nine upgrade hops, FREEZE retains 99–101% of what retraining would deliver on Banking77: the retrained reference is flat at 92.8–93.2% over four generations while zero-shot climbs from 43.5% to 64.4% (Figure 1a). Fourteen months after training, the frozen specialist still leads the newest base by 28.6pp ($p < 10^{-200}$), so H_{50} and H_0 both lie beyond the observed series. Spider is the systematic outlier: freezing retains only 41–93% per hop (mean 0.64) because the retrain ceiling itself rises every generation (69.6%→75.8%)—new-base SQL competence penetrates the fine-tune. Its advantage decays from +17.2pp to +4.0pp within one release interval ($H_{50} \in (0, 4]$ months, p=0.005) yet takes the full series to cross zero ($H_0 = 14$ months, censored to $(7, 14]$). The two horizons differ by an order of magnitude, which a single "half-life" number would conflate. xLAM-FC sits between, and non-monotonically: Qwen3's zero-shot function calling *drops* 2.4pp from Qwen2.5, so the 2024 specialist's advantage rebounds from +2.6pp to +5.1pp. Aggregate trends like the densing law (Xiao

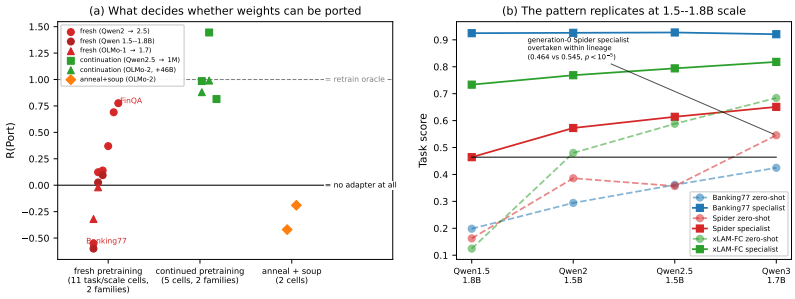

Figure 2: **(a)** Retention of naive porting between architecturally identical checkpoints, two families: at or below the no-adapter floor across independent pretraining runs (eleven cells); 0.82–1.45 across short continued pretraining (five cells). **(b)** The structure replicates at 1.5–1.8B scale.

et al., 2025) do not govern single tasks: "wait for the next base" is not directionally safe without task-level measurement.

Durability is a task–series property, not a task property: on the OLMo release series the frozen Spider specialist holds a 39–58pp advantage across nine months with no half-value crossing, because that series' SQL floors and references barely move. Erosion is driven by the series' floor growth, not by the task.

5.2 What separates the tasks: task–base coupling

Adding a second task per class sorts the suite by what the underlying competence depends on, not by class. With three seeds at every endpoint the claim rests on, retrained references rise on Spider ($+4.8$pp; every Qwen3 seed exceeds every Qwen2 seed) and mildly on xLAM ($+2.3$pp, $p{=}3{\times}10^{-4}$); they are flat within training noise on Banking77, glaive-FC, FinQA, and CLINC150 (an apparent CLINC decline at one seed did not survive three). The coupling elasticity spans a continuum: $\beta = 1.49$ (Spider), 0.17 (xLAM), 0.07 (Banking77), 0.00 (glaive-FC), -0.07 (CLINC150). FinQA's generative surface but data-bound competence (a dataset-specific program DSL) shows the continuum cuts across the discriminative/generative distinction transfer evaluations usually adopt.

The 1.5–1.8B replication reproduces and sharpens the structure: the Banking77 ceiling stays flat while zero-shot climbs 22.6pp; the Spider ceiling rises every generation and the generation-0 specialist is overtaken *within* the lineage (-8.1pp at $g{=}3$, $p{=}2{\times}10^{-6}$)—directly observing the crossover we could only bracket at 7–8B. Moreover xLAM, nearly flat at 7–8B, gains $+8.5$pp at small scale: coupling is a relation between the task and the model class's remaining headroom, so today's base-bound tasks will drift toward data-bound as bases improve—a falsifiable claim the released harness can track.

5.3 Weights are fragile: the continuity-and-distance criterion

Prior work treats naive weight copying between similar bases as a strong baseline; our prospectively specified hypothesis predicted $R \geq 0.8$ on the shape-compatible hop. *Both are wrong.* On the Qwen2→2.5 hop—architecturally identical in every dimension—the copied Banking77 adapter scores .4286: a 50-point collapse, and 17.9 points *below* the .6071 the bare target achieves ($R = -0.551$). Across nine task/scale cells on this hop, seven lie at or below $R{=}0.37$, and porting is worse than FREEZE in every cell.

The continuation hop reverses the verdict: onto Qwen2.5-1M, a continued-pretraining descendant of the same weights, the same adapters retain $R = 0.985/1.445/0.815$, with no detected difference from retraining and non-inferiority at 2pp (App. D).

Ground truth. The OLMo panel (Table 3) reproduces both regimes where genealogy is documented, then locates the boundary. On the verified fresh pair, porting collapses exactly as on Qwen. Holding the source fixed and moving the target along the run's own trajectory: at a 46B-token continuation ($\sim$1% of the run) the ported adapter retains $R = 0.88$–0.99, though reference parity is already

Table 3: OLMo panel: porting the same intent adapters along a fully documented trajectory. Top pair: verified fresh pretraining; below: targets progressively farther along OLMo-2's own run. Exact McNemar vs. floor (↓: below; ≈: indistinguishable).

Target edge	Task	PORT	Floor	Ref	R	p vs floor
OLMo-1 → 1.7 (fresh run)	Banking77	.0266	.2464	.9292	-0.32	$< 10^{-169}$ (↓)
	CLINC150	.2364	.2482	.9151	-0.02	0.063 (≈)
+46B cont.	Banking77	.8179	.0097	.9305	$+0.88$	$< 10^{-300}$
	CLINC150	.8951	.0120	.9078	$+0.99$	$< 10^{-300}$
+2.9T cont. (end stage 1)	Banking77	.1279	.1432	.9289	-0.02	0.074 (≈)
	CLINC150	.1433	.0315	.9204	$+0.13$	$< 10^{-138}$
+anneal (pre-soup)	Banking77	.1276	.3055	.9315	-0.28	$< 10^{-78}$ (↓)
	CLINC150	.1296	.2822	.9122	-0.24	$< 10^{-111}$ (↓)
+soup (released)	Banking77	.1140	.3523	.9256	-0.42	$< 10^{-118}$ (↓)
	CLINC150	.1313	.2575	.9125	-0.19	$< 10^{-78}$ (↓)

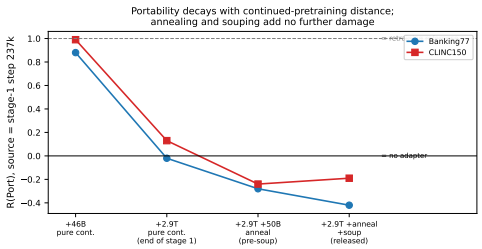

Figure 3: Porting one adapter (trained at an OLMo-2 stage-1 checkpoint) onto four later checkpoints of the same documented run. Retention falls to the no-adapter floor within the pure-continuation segment; annealing and souping add no further damage.

rejected on Banking77 (-11.3pp, $p < 10^{-79}$): even short continuations exact a measurable tax. At 2.9T further tokens of *pure* continuation it is statistically at the floor; the stage-2 anneal and soup merge change the ported score by at most 1.4pp—the more negative R there reflects later checkpoints' higher floors, not damage. (Our wave-4 text attributed the collapse to annealing/souping; the wave-5 ablation overturned that reading.) **Portability is a distance budget in continued-pretraining tokens, not an architecture or family property** (Figure 3)—and since released checkpoints typically sit trillions of tokens past any point where an old adapter was trained, the copyable regime is far narrower than "same family" suggests.

Ruling out artifacts. Four checks: the failure is not a parsing problem (48.1% of collapsed outputs are near-miss label variants, no degenerate generations); not a chat-template mismatch (a full template×base swap changes no score to four decimals); not a 4-bit artifact (bf16 re-run differs by 0.6pp against a 50-point collapse); and not purely a format regression (label-constrained decoding recovers .7505→.8669 but remains 7.4pp below the constrained reference—a real mapping loss persists). The collapse magnitude is rank-dependent (at r=64 the Banking77 loss shrinks to 21pp) but its ordering—port below FREEZE and below retraining—holds at both ranks; full fine-tuning remains unmeasured.

5.4 THE DATA RECOVERS WHAT THE WEIGHTS CANNOT CARRY

REFRESH (Eq. 3)—the remedy the cross-base transfer literature itself proposes (Wang et al., 2024; Jung & Kim, 2026; Guan et al., 2026)—is annotation-free distillation with retained inputs: the old specialist (teacher) relabels them, and a fresh adapter (student) is trained on the new base; in effect, a longitudinal test of hard-label distillation across real release intervals. On the newest hop, refreshed students reach retention $0.997/1.047/0.960$ (Banking77/Spider/xLAM) against gold-label retraining, with no detected difference (McNemar $p \geq 0.58$) and non-inferiority at a 2pp margin; a CLINC150 cell extends the pattern to a second intent-classification task, landing inside its three-seed reference

Algorithm 1 Upgrade policy for hop $b_i \to b_j$ (ε: 1pp classification, 2pp structured; gates on a held-out split). Trying REFRESH before RETRAIN treats annotation as the scarce resource; with reusable gold labels in hand, swap those rungs (§5.4).

estimate $F_t(b_j)$, $O_t(b_j)$, and $S_t(s_{b_i,t})$ on the gate split
if $O_t(b_j) - S_t(s_{b_i,t}) \leq \varepsilon$ **then**
    return FREEZE ▷ new base buys nothing over the old specialist
else if shapes match **and** b_j is a documented *short* continuation of b_i **and** PORT passes the regression gate at ε **then**
    return PORT
else if task inputs D_X retained **and** REFRESH student passes the gate **then**
    return REFRESH ▷ treats gold labels as the scarce resource
else
    return RETRAIN
end if

Table 4: Offline policy replay over 33 measured episodes. Gates read one half of each test set; regret and regressions are scored on the disjoint half, against the best measured action per episode.

Policy	Mean regret	Max regret	GPU-h	Gold-label eps.	Regressions
Always-FREEZE	1.80pp	11.5pp	0	0	0
Always-ADOPT	32.35pp	91.6pp	0	0	33
Always-RETRAIN	0.21pp	2.0pp	47.7	33	1
Port-if-shape	17.11pp	90.8pp	25.8	15	14
Proposed (Alg. 1)	**0.37pp**	6.6pp	**15.6**	**9**	**0**

band. Notably, the Spider teacher is only 73% accurate yet its student matches the gold reference, echoing weak-to-strong generalization (Burns et al., 2024), though our seeds cannot separate that from noise. The resource picture is asymmetric: REFRESH saves *annotation*, not always compute (39–331 GPU-minutes vs. 46–216 for retraining); if gold labels were retained, plain retraining dominates it. Its niche is the regime where no reusable gold set exists: specialists trained from preference or feedback signals rather than a labeled dataset (the behavior is the only artifact), and specialists obtained without their data—third-party adapters, or models inherited from a vendor or a departed team. There, RETRAIN is off the menu at any annotation price short of rebuilding the pipeline, and REFRESH is the only rung between FREEZE and a full rebuild. One caveat: REFRESH addresses label *availability*, not concept shift—if the correct labels changed, the teacher is as stale as the labels. On the taxed task, $C_{95} > 4{,}096$ gold labels while REFRESH reaches 105% of the gain with zero (App. E): where the tax is real, recycling the old specialist's behavior is cheaper than re-annotating.

5.5 WHAT AVERAGES HIDE: BEHAVIORAL REGRESSIONS

From the released per-example records we compute negative/positive flip rates for every strategy pair. Even quality-neutral retraining churns: replacing the Qwen2.5 specialist with a retrained Qwen3 reference negatively flips 1.2% (Banking77) to 4.7% (Spider) of items when the net change is zero or positive; an apparent -0.9pp CLINC decline hides 140 previously-correct items breaking (Yan et al., 2021). Naive porting is catastrophic on this axis (50.4% negatively flipped), while REFRESH matches gold retraining's flip profile—refresh adds no churn beyond retraining itself. Our policy gates on this axis.

5.6 A DECISION POLICY OVER THE LADDER

We test the framework's claim—that genealogy, coupling, and flips suffice to choose well—by offline replay over the 33 task×scale×hop episodes where FREEZE, ADOPT, and the retraining reference are all measured, with Algorithm 1 pre-specified. To avoid scoring the policy on the measurements its gates consult, each test set is split in half by item-id hash: gates read one half, regret is computed on the other. The policy tracks always-retraining (0.37 vs. 0.21pp mean regret) at 33% of its compute and 27% of its gold-label episodes, with zero regressions (Table 4); its worst episode is a gate miscall on a half-sized test set. The instructive failure is *port-if-shape*—the rule practitioners actually use: 17pp

mean regret and 14 regression episodes, because shape compatibility fires exactly where porting is most dangerous. Regret is relative to measured actions only; an unmeasured method could dominate.

The small-evidence regime. Deployment gates are far smaller than our test sets: an organization typically holds n=100–600 items, where the point-estimate gate is unusable (13–18% false opens at $n \leq 300$). Appendix F replaces it with a three-zone equivalence-testing gate (false opens 0.2–0.5% at every n) and shows that labeling 25 *disagreements* beats 100 i.i.d. labels for resolving the upgrade direction.

5.7 Predicting portability before paying for it

Can a probe costing one forward pass per prompt replace the measured port? For the eight base pairs with measured $R(\textsc{Port})$, we run 256 fixed task prompts through both checkpoints. Linear CKA between hidden states rank-correlates with mean retention at Spearman ρ=0.74; next-token top-1 agreement and JSD are weaker (0.38, -0.52). The failure mode is informative: CKA tracks the OLMo distance spectrum monotonically but cannot distinguish the Qwen fresh pair from the Qwen continuation pair. At n=8 this is exploratory: probes carry real signal, but none substitutes for one cheap measured port plus a regression gate.

6 Discussion

The durable asset is the data, not the weights. The three result groups compose into one statement. The weight artifact is bound to its pretraining trajectory: it cannot cross a fresh run and survives only tens of billions of continuation tokens—a budget every major release exceeds. The specialization is durable twice over: frozen, it outperforms cheap adoption for months to years wherever coupling is low; and wherever it must move, it moves through *data*—distilling the old specialist into the new base recovers reference quality with zero new annotation. For the specialist-fleet debate (Belcak et al., 2025), this reframes staleness as a priceable line item and relocates what an organization should treasure: the task data, the regression records, and the refresh pipeline—not the frozen weights.

What default policies get wrong. "Always retrain" wastes 17–422 GPU-minutes per specialist per generation for no measurable accuracy on low-coupling tasks; "never revisit" forfeits 42–59% of attainable gain per hop on high-coupling ones; "just move the weights over" depends on release genealogy and distance, not `config.json`—and is the worst measured policy. The rational policy is per-task and per-release (Alg. 1); the coupling that drives it drifts with scale and time, so must be re-measured—which the released harness automates. Freezing is not free either: it carries security-patch, dependency, and compliance costs we do not price.

Limitations. Specialists are capped at 8B and adapted only with QLoRA at rank 16; full-parameter fine-tuning perturbs weights at much higher effective rank (Biderman et al., 2024) and may transfer differently—the rank-64 replicate shows the collapse *magnitude* is rank-dependent, though its ordering is not. One primary lineage confounds family culture with generational effects; OLMo mitigates but does not remove this. The continuity criterion rests on two fresh-pretraining hops, two short continuations, and one measured distance spectrum: a twice-replicated dichotomy with one measured boundary, not a universal decay rate. Tasks are English, two per class; REFRESH is measured in its input-retaining variant (Wang et al., 2024); the policy replay's regret is relative to measured actions only.

7 Conclusion

We measured the base-model upgrade decision along a complete fourteen-month open-weight release lineage, at two scales, across six tasks, with a documented-genealogy validation family. Specialization is durable where task–base coupling is low and erodes fast where it is high; porting is licensed by pretraining continuity and distance, not architectural compatibility; retained data recovers reference quality on any new base at zero annotation cost; and a pre-specified policy on these facts nearly matches always-retraining at a third of its budget. When the base model ships, the asset to protect is the data, not the weights.

Reproducibility Statement

All experiments run on a single RTX 4090 with pinned software versions (App. C). We release training and evaluation scripts, all 98 adapter checkpoints, per-query prediction records and bootstrap CIs for all 193 evaluation cells, dataset split seeds and manifests (including the glaive-FC leakage audit and re-split, App. B), the relabeled REFRESH corpora, the memorization probe (App. G), the policy-replay and small-n code (App. F), and raw cost logs (wall-clock, peak VRAM, integrated energy) behind every figure. Prospectively specified hypotheses ship verbatim with timestamps (App. A).

Ethics Statement

All datasets are public under their stated licenses; no private or personal data is used. The benchmark informs build-vs-buy decisions with labor implications; we report costs transparently, including 47.8 kWh of measured training energy. Fine-tuning can erode safety alignment (Qi et al., 2024); released adapters are task-scoped and carry no safety claims. The study uses only publicly released checkpoints and documentation; model and task selection criteria were fixed before the first experiment.

AI Use Statement

In this work, we used a large language model (Anthropic's Claude) for tasks in the policy's recommended-disclosure category: editing the manuscript for readability, condensing prose to the page limit, suggesting section structure, and formatting the LaTeX source and references. We did not use generative AI tools for any required-disclosure research subtask: not to propose or refine hypotheses, design the methodology or experiments, implement methods, generate or process data, formulate or prove mathematical claims, or interpret results; the remaining required-disclosure subtasks are not applicable to this work. All experiments, analyses, measured numbers, and claims originate from the authors' released code and study registry, against which all AI-assisted text was reviewed and verified by the authors. We take responsibility for the final content of this work.

References

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic AI. *arXiv preprint arXiv:2506.02153*, 2025.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, et al. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024.

Martin Juan José Bucher and Marco Martini. Fine-tuned 'small' LLMs (still) significantly outperform zero-shot generative AI models in text classification. *arXiv preprint arXiv:2406.08660*, 2024.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Deng Cai, Elman Mansimov, Yi-An Lai, Yixuan Su, Lei Shu, and Yi Zhang. Measuring and reducing model update regression in structured prediction for NLP. In *Advances in Neural Information Processing Systems*, 2022.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020.

Lingjiao Chen, Matei Zaharia, and James Zou. How is ChatGPT's behavior changing over time? *arXiv preprint arXiv:2307.09009*, 2023a.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023b.

Zhiyu Chen, Wenhu Chen, Charese Smiley, et al. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023.

Mehmet Hamza Erol, Batu El, Mirac Suzgun, Mert Yuksekgonul, and James Zou. Cost-of-pass: An economic framework for evaluating language models. *arXiv preprint arXiv:2504.13359*, 2025.

Farzad Farhadzadeh, Debasmit Das, Shubhankar Borse, and Fatih Porikli. LoRA-X: Bridging foundation models with training-free cross-model adaptation. In *International Conference on Learning Representations*, 2025.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

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 (ACL)*, 2025.

Yushi Guan, Jeanine Ohene-Agyei, Daniel Kwan, et al. TuneShift-KD: Knowledge distillation and transfer for fine-tuned models. *arXiv preprint arXiv:2603.24518*, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Shih-Cheng Huang, Pin-Zu Li, Yu-Chi Hsu, Kuang-Ming Chen, Yu Tung Lin, Shih-Kai Hsiao, Richard Tzong-Han Tsai, and Hung-yi Lee. Chat vector: A simple approach to equip LLMs with instruction following and model alignment in new languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

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

Chanjoo Jung and Jaehyung Kim. TiTok: Transfer token-level knowledge via contrastive excess to transplant LoRA. In *International Conference on Learning Representations*, 2026.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *arXiv preprint arXiv:2407.01502*, 2024.

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.

Stefan Larson, Anish Mahendran, Joseph J. Peper, et al. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

Yanan Li, Fanxu Meng, Muhan Zhang, et al. LoRASuite: Efficient LoRA adaptation across large language model upgrades. In *Advances in Neural Information Processing Systems*, 2025.

Pin-Jie Lin, Rishab Balasubramanian, Fengyuan Liu, Nikhil Kandpal, and Tu Vu. Efficient model development through fine-tuning transfer. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

Zuxin Liu, Thai Hoang, Jianguo Zhang, et al. APIGen: Automated pipeline for generating verifiable and diverse function-calling datasets. *arXiv preprint arXiv:2406.18518*, 2024.

Guillermo Ortiz-Jiménez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, 2023.

Xiangyu Qi, Yi Zeng, Tinghao Xie, et al. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

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

Ying Sheng, Shiyi Cao, Dacheng Li, et al. S-LoRA: Serving thousands of concurrent LoRA adapters. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, et al. 2 OLMo 2 furious. In *Conference on Language Modeling (COLM)*, 2025.

Shivani Tripathi, Pushpanjali Nema, Aditya Halder, Shi Qiao, and Alekh Jindal. Prompt migration: Stabilizing GenAI applications with evolving large language models. *arXiv preprint arXiv:2507.05573*, 2025.

Runqian Wang, Soumya Ghosh, David Cox, et al. Trans-LoRA: Towards data-free transferable parameter efficient finetuning. In *Advances in Neural Information Processing Systems*, 2024.

Feifan Xia, Mingyang Liao, Yuyang Fang, Defang Li, Yantong Xie, Weikang Li, Yang Li, Deguo Xia, and Jizhou Huang. Cross-LoRA: A data-free LoRA transfer framework across heterogeneous LLMs. *arXiv preprint arXiv:2508.05232*, 2025.

Chaojun Xiao, Jie Cai, Weilin Zhao, Guoyang Zeng, Xu Han, Zhiyuan Liu, and Maosong Sun. Densing law of LLMs. *Nature Machine Intelligence*, 7:1823–1833, 2025.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Tao Yu, Rui Zhang, Kai Yang, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Justin Zhao, Timothy Wang, Wael Abid, et al. LoRA land: 310 fine-tuned LLMs that rival GPT-4, a technical report. *arXiv preprint arXiv:2405.00732*, 2024.

A PROSPECTIVELY SPECIFIED HYPOTHESES AND OUTCOMES

Hypotheses were fixed in the released runner scripts before the wave that tests them; the scripts are released verbatim with timestamps, but were not deposited with an external immutable registry. We therefore call them *prospectively specified* rather than pre-registered, and mark each hypothesis with the wave boundary it preceded. Outcomes are reported against them, refutations included.

RQ1 (Durability). How large is $A_t(\tau)$, and where are H_{50} and H_0? **H1** (before wave 1): crossover within one generation for reasoning-heavy structured tasks, two or more for discriminative tasks. *Outcome*: directionally supported for the discriminative task ($H_0 > 14$ months, advantage 27.1pp over the 5-shot floor at the last observed release, $p < 10^{-200}$) but refuted in magnitude for text-to-SQL: the crossover took the full observed series ($H_0 = 14$ months, interval-censored to $(7, 14]$), while the half-value horizon was under one release interval ($H_{50} \in (0, 4]$ months; the advantage drops from 17.2pp to 4.0pp at the first hop). The two horizons differ by an order of magnitude, which the original single "half-life" number conflated.

RQ2 (Strategies). What retention at what cost? **H2** (before wave 1): on shape-compatible hops PORT retains $R \geq 0.8$; across shape breaks only REFRESH exceeds $R = 0.5$. *Outcome*: both clauses refuted as stated. Between independent pretraining runs PORT retained -0.60 to 0.78 across nine task/scale cells, below FREEZE in all nine; between checkpoints joined by a short continuation it retained 0.82–1.45. Shape was the wrong variable, and (after wave 5) continuity alone is not sufficient either: portability decays with continued-pretraining distance (§5.3). REFRESH reached 0.96–1.05, far above 0.5, at zero gold labels.

RQ3 (Task class). Does a discriminative-vs-generative retention gap replicate on real release series? **H3** (before wave 1): retention ranks D > S > A. *Outcome*: refuted. Reference scores rise only where the coupling between task competence and base capability is high, and PORT's ordering (A > S > D) inverts the prior report. We describe the coupling with the continuous elasticity $\beta_t = \Delta O_t / \Delta F_t$ rather than a binary base-bound/data-bound label, since the coupling itself shifts with model scale (§5.2).

RQ4 (Prompt asset). Is prompt-side specialization more durable than weight-side? *Outcome* (pilot, App. H only): the prompt asset is roughly 30–80$\times$ smaller in delivered gain on the one task/hop measured, and its optimum flips across generations; the pilot lacks per-example records, so we report it as a bounded observation, not a contribution.

RQ5 (Predictability). Do cheap representational probes predict retention before any migration is paid for? *Outcome* (wave 6, exploratory; §5.7): across the eight measured base pairs, linear CKA over 256 task prompts rank-correlates with mean R(PORT) at Spearman $\rho = 0.74$; next-token top-1 agreement and JSD are weaker (0.38, -0.52). No single probe is decision-grade at $n = 8$: CKA separates the OLMo distance spectrum cleanly but cannot distinguish the Qwen fresh pair from the Qwen continuation pair.

Wave-4 predictions (fixed 2026-07-31, before any OLMo run). **P1**: porting OLMo-1 adapters onto OLMo-1.7 (verified fresh pretraining) collapses, $R < 0.5$ and below FREEZE; **P2**: porting between OLMo-2 stage-1 checkpoints of the same run ($\sim$46B tokens apart) retains $R \approx 1$, indistinguishable from the reference; **P3**: porting across the stage-2 anneal and soup merge is a stress test whose either outcome delimits the criterion. *Outcomes* (§5.3): P1 confirmed ($R = -0.32$ and -0.02, at or below the adoption floor). P2 directionally confirmed ($R = 0.88$ and 0.99, far above the floor) but exact reference parity is rejected on Banking77 (-11.3pp, $p < 10^{-79}$): even 46B continuation tokens exact a small, measurable tax. P3's collapse was confirmed, but its attribution was not resolvable from that wave alone.

Wave-5 predictions (fixed 2026-08-10, before any wave-5 run). **Q1**: if the Qwen port collapse is not a chat-template artifact, swapping source/target templates changes nothing; **Q2**: constrained decoding shrinks but does not eliminate the CLINC generational decline; **Q3**: if the wave-4 P3 collapse is driven by annealing/souping, porting onto the end-of-stage-1 checkpoint (2.9T tokens of *pure* continuation, no anneal, no soup) stays high; if driven by distance, it collapses there already.

Outcomes: Q1 confirmed exactly (template swap changes no score by even one item). Q2 confirmed (the decline shrinks by roughly a third under constrained decoding). Q3 resolved *against* the annealing/souping reading our wave-4 text had favored: porting is already at the floor after 2.9T tokens of pure continuation, and the anneal and soup edges add no further damage. We report the correction rather than the earlier interpretation.

Wave-6 predictions (fixed 2026-08-16). **W1**: on a second family (the OLMo release series), the high-coupling task's half-value horizon is again shorter than one release interval; **W2**: next-token top-1 agreement predicts $R(\text{PORT})$ better than CKA; **W3**: the fresh-hop port collapse at rank 64 matches the rank-16 magnitude. *Outcomes*: W1 refuted, informatively: on OLMo the frozen Spider specialist keeps a 39–58pp advantage across nine months of releases, because that series' floors and reference scores barely move on SQL; durability horizons are properties of the task–series pair, not of the task. W2 refuted—CKA is the stronger rank-predictor ($\rho = 0.74$ vs. 0.38). W3 refuted in magnitude: at r=64 the Banking77 port loses 21pp instead of 50pp (R: $-0.55 \rightarrow +0.29$) and Spider reaches $R = 0.74$, though porting stays below both FREEZE and the reference at both ranks; the collapse magnitude is rank-dependent, its ordering is not.

B LINEAGE, TASK, AND EVALUATOR DETAILS

B.1 LINEAGE DETAILS

Architecture parameters in Table 2 are read directly from each checkpoint's `config.json` and reproduced in the release. Notes: (i) the vocabulary size alternates ($151{,}936 \rightarrow 152{,}064 \rightarrow 152{,}064 \rightarrow 151{,}936$), so embedding-row alignment differs by hop even though the tokenizer family is constant; (ii) Qwen1.5-7B uses full multi-head attention (32 query, 32 key/value heads), making its KV cache $8\times$ larger than Qwen2/2.5's GQA configuration—a deployment-relevant regression that per-parameter comparisons miss, and in our runs the dominant cost factor for long-prompt evaluation on a consumer GPU; (iii) Qwen3-8B introduces an explicit head dimension of 128 and a narrower FFN (12,288) than Qwen2.5 (18,944) despite being wider overall; (iv) Qwen2.5-7B-Instruct-1M (2025-01) matches Qwen2.5-7B-Instruct in every architectural dimension and extends its context window via continued pretraining, making it the continuation release of §5.3; (v) the small-scale track uses Qwen1.5-1.8B-Chat (MHA 16/16, hidden 2048, 24 layers), Qwen2-1.5B-Instruct and Qwen2.5-1.5B-Instruct (identical: GQA 12/2, hidden 1536, 28 layers, FFN 8960), and Qwen3-1.7B (GQA 16/8, hidden 2048, 28 layers).

Table 5: Primary-lineage scores (test sets; greedy decoding; 4-bit inference). "Spec." = specialist fine-tuned on that generation (the retrain reference for hops ending there). Bootstrap 95% CIs for all cells are released; typical half-widths $\pm$0.9pp (Banking77), $\pm$2.8pp (Spider), $\pm$1.6pp (xLAM).

	Qwen1.5-7B		Qwen2-7B		Qwen2.5-7B		Qwen3-8B	
Task	0-shot	Spec.	0-shot	Spec.	0-shot	Spec.	0-shot	Spec.
Banking77	.4354	.9302	.5747	.9276	.6071	.9312	.6438	.9321
(5-shot)	.5679		.6273		.6357		.6594	
Spider	.5242	.6963	.6567	.6992	.6760	.7321	.6963	.7582
xLAM-FC	.5770	.8205	.6315	.8225	.7940	.8480	.7700	.8455

Table 6: Per-hop retention by strategy on the primary 7–8B lineage. FREEZE is measured on all hops; PORT on the shape-compatible 2→2.5 hop; REFRESH on the most recent hop (2.5→3), with cost = teacher relabeling + student training on one RTX 4090.

	R(FREEZE)				R(PORT)	REFRESH	
Task	1.5→2	2→2.5	2.5→3	mean	2→2.5	R	cost (min)
Banking77	1.007	0.989	0.997	0.998	-0.551	0.997	39
Spider	0.932	0.414	0.578	0.641	0.139	1.047	255
xLAM-FC	0.990	0.528	1.033	0.850	0.370	0.960	331

Table 7: Fine-tuned ceilings across the three most recent 7–8B generations. Δ = Qwen3 minus Qwen2; p from the exact McNemar test on the paired test set; ^{3s} = three seeds at the endpoints.

Task	Class	Qwen2	Qwen2.5	Qwen3	Δ (pp)	Evidence
Spider^{3s}	S	.7025	.7321	.7501	$+4.8$	all seeds ordered
xLAM-FC^{3s}	A	.8225	.8473	.8460	$+2.3$	3×10^{-4}
FinQA	S	.5981	.5990	.6103	$+1.2$	0.22
Banking77	D	.9276	.9312	.9321	$+0.5$	0.16
glaive-FC	A	.9640	.9650	.9645	$+0.05$	1.0
CLINC150^{3s}	D	.9313	.9322	.9262	-0.5	within seed noise

Small-scale lineage scores. Zero-shot / specialist by generation. Banking77: .1984/.9247, .2942/.9260, .3614/.9273, .4247/.9208. Spider: .1625/.4642, .3859/.5725, .3569/.6141, .5455/.6509. xLAM-FC: .1245/.7335, .4800/.7685, .5880/.7940, .6840/.8180. Porting on the small 2→2.5 hop: Banking77 .0218 ($R = -0.600$), Spider .3636 ($R = 0.026$), xLAM .6075 ($R = 0.095$).

OLMo validation lineage. From `config.json`: OLMo-1-7B and OLMo-1.7-7B are architecturally identical (hidden 4096, 32 layers, MHA 32/32, FFN 11008, vocabulary 50304); the OLMo-1.7 model card states it was trained from scratch. OLMo-2-7B and its published stage-1 checkpoints at steps 237000 and 248000 share one shape (hidden 4096, 32 layers, MHA 32/32, FFN 11008, vocabulary 100352); the two stage-1 snapshots are ∼46B training tokens apart within the same run, and the released base adds the remaining stage-1 tokens, a stage-2 anneal on a different data mix, and a model-soup merge (Team OLMo et al., 2025). All are base checkpoints; training and evaluation use one fixed plain-text format (system text, blank line, input, "Answer:") with completion-only loss and the intent-task recipe of App. C. Zero-shot floor / specialist reference per checkpoint, Banking77 then CLINC150 (lenient): OLMo-1 .0334/.8795, .1416/.8336; OLMo-1.7 .2464/.9292, .2482/.9151; s1-237k .0045/.9279, .0091/.9062; s1-248k .0097/.9305, .0120/.9078; s1-928646 .1432/.9289, .0315/.9204; stage2-ingredient3 .3055/.9315, .2822/.9122; OLMo-2 main .3523/.9256, .2575/.9125. The near-zero floors of mid-run checkpoints against the substantial floors of the released models reflect instruction-following ability arriving late in pretraining; retention is computed against each target's own floor and reference, so the comparisons are unaffected. Cross-family comparisons of absolute scores are not meaningful (base-model floors vs. Qwen's instruct floors); all OLMo conclusions are within-family.

B.2 Task, prompt, and evaluator details

Six tasks (splits fixed at seed 42 and released): Banking77 (77-way intent; 9,003/1,000/3,080), CLINC150 (151-way intent including out-of-scope; 10,000/1,000/5,500), Spider (text-to-SQL; 6,500/500/1,034 dev; execution accuracy), FinQA (numerical programs; 6,251/500/1,147; program accuracy), xLAM-FC (function calling; 10,000/500/2,000 held-out; call-set match), glaive-FC (calling + refusal; 7,580/505/2,000 group re-split; call-set match). The original three tasks are measured across all four generations and both scales; the second task in each class across the three most recent 7–8B generations.

Full prompt templates, label normalization, the SQL execution comparator (result-multiset equality, 10^{-4} float rounding, 10 s timeout), and the call-set matcher (name equality; type-coerced, order-insensitive argument comparison) are released. Protocol notes: (i) Banking77 cheap adoption carries the 77-label inventory in the system prompt; zero-shot and 5-shot (fixed exemplars, seed 42, five distinct labels) are both reported; a manual audit found zero parsing failures across all baseline conditions—every baseline error is a genuine near-class confusion, so adoption floors are not deflated by format artifacts. (ii) Spider prompts serialize the schema (tables, columns, types, primary/foreign keys); the single-database execution variant's known false-positive risk is noted. (iii) xLAM-FC uses a held-out split of the APIGen-verified corpus (Liu et al., 2024); it measures in-distribution function calling. (iv) CLINC150 label matching is reported under a strict normalization and a lenient morphological one (e.g., mapping *translation* to the gold label *translate* when the match is unique); we use the lenient score as the adoption floor and release both (the Qwen2.5 zero-shot floor is .7173 strict, .7249 lenient, .7207 under constrained label decoding). (v) FinQA programs are normalized before exact match (whitespace, trailing commas, numeric canonicalization); its adoption floors are

volatile across generations (.221/.093/.215) because zero-shot models drift in and out of the dataset's program DSL, which we flag wherever a FinQA ratio depends on the floor.

glaive-FC leakage audit and re-split. An audit of our original random split found that 21.0% of test conversations (420/2,000) also appeared verbatim in training—the corpus reuses templates and tool schemas heavily—so all glaive-FC numbers in this paper use a re-split that groups conversations by (masked user template, called function) and assigns whole groups to train/val/test (7,580/505/2,000 from a 10,085-item exact-deduplicated pool, 5,428 groups). The re-split has zero cross-split leakage at exact, normalized-exact, and group level; the split manifest with SHA-256 hashes and the audit script are released. Scores on the clean split are 0.6–5.4pp lower than on the leaked split; the qualitative conclusion (a flat reference score across generations) is unchanged. glaive-FC retains 25% no-call negatives, so specialists are also graded on refusing to call.

Overlong-completion drops. Training drops examples whose completion is fully truncated at the context limit; counts vary with each model's tokenizer (Spider 138/6,500 for most generations but 154 for Qwen3; xLAM 160 vs. 161; FinQA 6–8, rising to 41/6,251 at the longer length budget; CLINC150 0/10,000; glaive-FC 96/10,000; Banking77 0) and are reported per run in the released logs rather than folded silently into the totals.

Table 8: Negative/positive flip rates between strategy pairs on the fixed test sets (per-example records; NF = old correct, new wrong).

Transition	NF%	PF%	Net (pp)
Banking77: 2.5-specialist → 3-reference	1.23	1.33	+0.10
Spider: 2.5-specialist → 3-reference	4.74	7.35	+2.61
xLAM: 2.5-specialist → 3-reference	3.90	3.65	−0.25
CLINC150: 2-specialist → 3-specialist	2.55	1.65	−0.89
FinQA: 2-specialist → 3-specialist	4.27	5.49	+1.22
glaive-FC: 2-specialist → 3-specialist	0.50	0.40	−0.10
Banking77: source specialist → PORT	50.39	0.49	−49.90
Banking77: reference → REFRESH	1.17	1.07	−0.10
Spider: reference → REFRESH	3.77	4.06	+0.29
xLAM: reference → REFRESH	2.20	1.90	−0.30

Format reliability. Fine-tuning buys parseability as well as accuracy: unparseable xLAM outputs drop from 17–28/2000 (zero-shot) to 5–6/2000 (specialists); ported adapters land in between (15/2000). Banking77 parsing failures are zero in all conditions.

C TRAINING SETUP AND FULL RUN INVENTORY

Recipe. One recipe for all lineage trainings: QLoRA (Dettmers et al., 2023) with NF4 double quantization, bf16 compute; LoRA $r = 16$, $\alpha = 32$, dropout 0.05 on all attention/MLP projections; learning rate 2×10^{-4} cosine, warmup 3%, completion-only loss, seed 42 (43/44 for variance probes). Intent tasks: 3 epochs, length 256, batch 16. Spider/FinQA: 3 epochs, length 1024/1280, micro-batch 4 × accumulation 4. Function calling: 2 epochs, length 1024, same batching (small-scale track: micro-batch 8 × 2). Optimizer `paged_adamw_8bit`; gradient checkpointing on.

Stack and hardware. Single RTX 4090 (24 GB); PyTorch 2.13.0+cu126, `transformers` 5.14.1, `peft` 0.19.1, `bitsandbytes` 0.50.0. Peak VRAM 6.5–21.1 GB across runs.

Inventory. 98 QLoRA trainings (12 lineage specialists at 7–8B, 9 second-task specialists, 12 small-scale specialists, 3 continuation-release references, 4 REFRESH students, 3 ladder budgets, 17 OLMo specialists/references along the documented trajectory and release series, 3 glaive re-split specialists, 4 rank-64 replicates, and 31 seed replicates), 4 relabeling passes, 193 evaluation cells, one memorization probe (4 bases × 3 test sets), one prompt-side pilot. Cost is a three-way split: 148.7 training GPU-hours with 47.8 kWh of measured board energy (board power sampled at 10 s intervals, integrated; the power integral covers training only), 79.4 evaluation GPU-hours, and 4.6

teacher-relabel GPU-hours; $0 API spend. Longest single run: glaive-FC/Qwen3-8B, 422.5 min; largest energy: FinQA/Qwen3-8B, 1,255 Wh; largest peak VRAM: FinQA/Qwen3-8B, 21.1 GB; smallest run: Spider ladder-256, 8.0 min, 49.5 Wh. Evaluation cells take 1.2–31.7 minutes each under normal memory conditions; per-cell generation-token counts are logged.

Operational notes for single-GPU replication. (1) Exceeding physical VRAM triggers silent host-memory spill with 10–40× slowdown; with MHA checkpoints (KV cache 8× GQA's) and ∼1,000-token prompts this is systematic at batch 32 on 24 GB—the OLMo zero-shot floors cost 40.8 of the study's GPU-hours this way before an automatic batch-shrink guard was added; wall-clock is reported as measured. (2) MSYS-based shells exhaust fork resources after roughly a day of continuous subprocess churn; the idempotent runner resumes cleanly. (3) With plain-text prompts, completion-only masking must tokenize prompt and completion separately: BPE merges a trailing-space prompt boundary into the completion's first token, which silently masks that token from the loss—training converges normally while the specialist learns answers with their first token missing (reference self-evaluation collapsed to 5–17%). The wave-4 trainings were quarantined and rerun after this was caught by the reference check; the released runner now concatenates separately tokenized prompt and completion, making the training context a token-exact prefix of the evaluation prompt by construction. One evaluator bug (a type guard in the xLAM call matcher) was fixed and the affected cell rerun; the changelog documents it.

D Non-Inferiority Analyses

Absence of a detected difference is not parity, so alongside McNemar tests we test non-inferiority against pre-set margins (1pp classification, 2pp structured) using paired bootstrap CIs and three-seed reference bands.

Continuation hop (Qwen2.5 → 2.5-1M, PORT). Banking77 is non-inferior at 2pp but borderline at 1pp (port−reference CI $[-1.01, +0.10]$pp; three-seed reference band .9308–.9325). Spider's port (.7292) lies inside the three-seed reference range (.7137/.7340/.7408), so the comparison is bounded by training noise rather than by item noise (the $R = 1.445$ ratio reflects a small denominator: the 1M reference gains only 3.5pp over its floor). xLAM fails the 1pp test outright (CI $[-2.00, 0.00]$). In short: porting onto a short continuation costs at most about a point; porting across a fresh run costs fifty. Incidentally, the 1M release itself buys specialists nothing: frozen 2.5 specialists match or exceed 1M-retrained references on all three tasks (e.g., Spider .7321 vs. .7137, $p = 0.10$).

REFRESH (2.5 → 3). Banking77 is formally non-inferior at 1pp (paired CI $[-0.62, +0.42]$pp; three student seeds span .9302–.9312); Spider and xLAM are non-inferior at 2pp but not at 1pp (CIs $[-1.45, +1.93]$ and $[-1.20, +0.60]$). Spider's student seeds span .7302–.7689 (3.9pp), so its single-run point estimate sitting above the reference (.7611 vs. .7582) is inside training noise.

E Recovery Cost Under a Label Budget

On the taxed task (Spider, Qwen3), gold-label retraining with a budget ladder yields .6847 at 256 labels (nominally below the .6963 zero-shot floor, though not significantly, $p = 0.46$: 256 labels buy nothing), .7205 at 1,024 (39% of attainable gain), .7427 at 4,096 (75%), and .7582 at the full 6,362. The 95%-of-gain budget C_{95} exceeds 4,096 labels, whereas label-free REFRESH reaches 105% using retained inputs only. (An absolute-score criterion $Q_N \geq 0.95\,O$ gives systematically smaller budgets; we use the gain criterion because it is invariant to the floor.) Where the upgrade tax is real, the cheapest way to pay it is to recycle the old specialist's behavior, not to re-annotate.

F The Small-Evidence Regime: Protocol and Operating Curves

Every main-text analysis runs at n=1,034–5,500 paired test items; an organization applying Algorithm 1 to its own specialist typically holds n=100–600, and on a saturated task the two systems disagree on only a handful of them. This appendix characterizes the policy's operating behavior at those evidence sizes by replaying it over the benchmark's own per-example records, and revises the opportunity gate so its verdicts stay calibrated when the evidence cannot carry a point estimate.

Three-zone gate. At small n a point estimate passed through a fixed margin conflates three epistemic states. We replace it with an equivalence-testing (TOST-style) verdict on the paired-difference confidence interval, computed from the discordant counts (n_{01}, n_{10}) with a continuity correction, and a rule-of-three bound $|\delta| \leq 3.69/n$ when the two systems agree on every item. The CI lying entirely above ε establishes the gain and opens the waterfall of Alg. 1; entirely below ε establishes equivalence (freeze as a verdict, with the exclusion bound stated); an interval straddling ε is unresolved, and the policy says so instead of defaulting to freeze. Unresolved verdicts carry a posterior over the true gain, with the benchmark's 193-cell corpus as an empirical-Bayes prior per lineage kind (fresh pretraining: $\mu = +0.82$pp, $\sigma = 2.4$pp over 16 pair-task samples; continuation: $\mu = +0.51$pp, $\sigma = 0.66$pp; seed-to-seed training variance from the multi-seed waves folded into the observation noise), and resolve to COLLECT (label the disagreement set) when the posterior leaves the gain live, or WAIT (hold until the next release) when $P(\text{gain} > \varepsilon_{\text{dec}}) < 0.10$.

Disagreement-first labeling. The exact McNemar test conditions on the discordant pairs, so the decision-relevant annotation complexity is $O(\#\text{disagreements})$, not $O(n)$: an organization can run both systems over unlabeled traffic, collect the items where their predictions differ, and label only those.

Protocol. We take the 21 (task, upgrade-pair) cells of the main study for which both endpoints have full per-example records (15 ground-truth null or negative at full n, i.e. $\delta_{\text{full}} \leq \varepsilon$; 6 ground-truth upgrades), subsample enterprise-sized gate sets ($n \in \{100, \ldots, 1000\}$, 300 replicates per cell per size, fixed seed), and score each verdict against the full-n ground truth. The corpus prior is computed leave-one-pair-out, so no cell is judged under a prior that contains itself. COLLECT is played out, not assumed: up to 50 real disagreement records are drawn and labeled, and the waterfall opens only if the exact sign test settles positive. The comparison rule is the point-estimate gate of §5.6 applied directly at small n.

Table 9: Small-evidence operating curve, replayed on real records. At enterprise gate sizes the point-estimate gate opens the upgrade waterfall on one in six ground-truth-null episodes; the three-zone gate holds the false-open rate near its nominal level at every n, and its equivalence verdicts are almost never wrong. The cost of that conservatism is bounded: mean serving-accuracy regret 0.58pp vs. 0.48pp at $n = 100$ (the point-estimate rule's spurious opens occasionally land on real upgrades; its migration and regression costs are outside this metric).

	false-open (15 null cells)		false-equivalence
n	point-estimate gate	three-zone + COLLECT	(6 upgrade cells)
100	17.9%	1.9%	0.5%
200	15.2%	2.1%	0.2%
300	13.0%	2.0%	0.2%
600	8.2%	1.8%	0.1%
1000	1.8%	1.5%	0.0%

Table 10: Probability that the upgrade direction settles (exact sign test, correct sign) under a fixed labeling budget, averaged over the cells with $|\delta_{\text{full}}| \geq 1$pp. Labeling b random items wastes almost the entire budget on items both systems already get right; labeling b disagreements concentrates it on the only items that carry decision information.

labeling budget b	i.i.d. sampling	disagreement-first
25	0%	37%
50	4%	48%
100	13%	67%

Three regime-level facts follow (Tables 9–10). First, at $n \leq 300$ the point-estimate gate is not a usable instrument: its false-open rate (13–18%) means roughly one in six null episodes triggers a retrain-and-migrate cycle. Second, honesty is cheap: refusing to over-claim costs about a tenth of a percentage point of mean serving accuracy while cutting false opens by an order of magnitude, and the equivalence verdicts that replace "freeze by default" are correct essentially always. Third,

when the verdict is unresolved, the efficient next step is never "collect more i.i.d. labels": at these disagreement rates (1–15%), a budget of 25 disagreement labels outperforms 100 i.i.d. labels by a factor of three on direction resolution. The replay code and per-cell results ship with the released tooling.

G Memorization Probe

For 200 test items per task, each base receives the first 60% of the raw item text (no chat template) and greedily continues 24 tokens; we report the longest-common-prefix ratio against the true continuation and an 8-gram hit rate. Verbatim rates (LCP > 0.5): Banking77 0–1%, Spider 0%, xLAM 3–5% (with 8-gram rates 0.32–0.46 driven by templated JSON tool syntax). Rates are approximately constant across generations, so cross-generation comparisons are not differentially contaminated. This probe detects verbatim memorization only; it cannot rule out paraphrased exposure. Any residual contamination would inflate baselines, making measured expert advantages conservative.

H Prompt-Side Pilot

On the Banking77 2.5→3 hop we compare the weight asset against a prompt asset (best of six system prompts selected on a 200-item dev slice). The prompt asset is small and unstable: dev-selected prompts gain at most $+0.4$pp over the default instruction on the new base (.6474 vs. .6438 zero-shot), porting the old best prompt recovers .6416, and the identity of the best prompt changes across the hop. The weight asset delivers $+27$–29pp over the same floors and survives three generations (§5.1). Both routes pay an upgrade tax; on this task the weight asset is 30–$50\times$ larger in delivered advantage and no more fragile. Scope: one task, one hop, a deliberately small prompt-search budget, dev-slice selection, no per-example records—this bounds the prompt asset from below rather than characterizing prompt engineering in general; agentic harnesses with tool scaffolds may behave differently, an extension the released harness supports.

I Refresh Details

Relabel variant: teacher = the Qwen2.5-7B specialist for each task; teacher inference over retained train+val inputs (5.5 / 112.6 / 152.5 minutes for Banking77 / Spider / xLAM); student recipe identical to App. C. Teacher–gold agreement: 97.45%/92.6% (Banking77 train/val), 88.3%/66.8% (Spider exact-string; semantic agreement is higher since equivalent SQL strings differ), 92.8%/83.4% (xLAM). Refreshed students score .9312 / .7611 / .8425, i.e. retention 0.997 / 1.047 / 0.960 against gold-label retraining (McNemar $p = 0.81$, 0.82, 0.58); the CLINC150 student distilled from the Qwen2.5 specialist scores .9267 on Qwen3, inside the three-seed band of its gold-label references (.9180–.9313). Where porting was applicable and achieved positive retention, refresh delivered $2.6\times$ (xLAM) to $7.5\times$ (Spider) more of the attainable gain; on Banking77 the comparison is degenerate because porting was net harmful. Three-way student–teacher–gold records are released. Zero gold labels, zero API calls.

J Dataset Licenses

Banking77 (CC-BY-4.0); Spider (CC BY-SA 4.0); xLAM-function-calling-60k (CC-BY-4.0, gated access via Hugging Face); CLINC150, FinQA, and glaive-v2 as listed by their official distributions. License texts checked 2026-07; re-verify all before redistribution.