Selective Plasticity for Self-Adapting Language Models: Topic-Expert Routing for Knowledge Retention During Test-Time Adaptation

Anonymous ACL submission

Abstract

Self-adapting language models generate their own fine-tuning data and use test-time training to incorporate new knowledge into parameters. However, when a single adapter must serve diverse domains, updates that improve one topic's vocabulary and factual recall progressively degrade another's. We identify two compounding failure modes in iterative self-adaptation: cross-topic interference within shared adapter parameters, and noise injection from mandatory weight merging. We show that neither increased adapter capacity nor four alternative update-rule modifications (experience replay, Fisher regularization, dictionary composition, variational routing) resolve this problem. We propose **Selective Plasticity**, replacing the shared adapter with N topic-specialized expert adapters and a learned router that dispatches passages to the appropriate expert. A conditional consensus merge transfers shared knowledge to base weights only when experts demonstrate agreement, preventing premature noise injection. On SQuAD, Selective Plasticity achieves 55.1% judge-evaluated accuracy versus 49.3% for the single-adapter baseline. Over 4 training iterations, the parameter-matched single-adapter baseline (r=256) degrades by 8.1pp while our approach maintains stable accuracy (0.0pp degradation; sign-test $p \approx 0.008$). On ARC-AGI few-shot reasoning (Chollet, 2019), where training tasks are topically homogeneous, both approaches perform equally, confirming that the benefit of structural isolation is specific to diverse-topic settings where interference is the bottleneck.

1 Introduction

Large language models are knowledge-frozen at deployment. Once trained, their parameters remain fixed even as tasks, domains, and the world's information evolve. This limitation has motivated self-adapting language models: systems that modify their own weights at inference time using self-generated supervision (Zelikman et al., 2022; Huang et al., 2023b; Gulcehre et al., 2023). Unlike retrieval-augmented generation (which cannot internalize knowledge into parameters) or full retraining (which risks catastrophic forgetting (McCloskey and Cohen, 1989) and requires expensive dataset passes), self-adapting LMs enable parameter-level adaptation that persists beyond the current context window.

SEAL (Zweiger et al., 2025) demonstrated that this is achievable: a model trained via ReST-EM (Singh et al., 2024; Gulcehre et al., 2023) to generate its own fine-tuning data ("self-edits") can incorporate new knowledge through test-time training (TTT) (Sun et al., 2020). However, a concrete failure reveals the mechanism's fragility. Consider the passage "The Shah's army was split into small, fragmented groups" with question "What enabled the Mongols' early victories?" The gold answer is "fragmentation," but after training across diverse topics (biology, law, sports), the single-adapter model produces "The army was weak," a vague, incorrect response that reflects accumulated interference from unrelated domains polluting the self-edit generation policy.

We identify two failure modes precisely. First, *cross-topic interference*: when a model simultaneously learns to generate self-edits about molecular biology and medieval military history, updates that sharpen one domain's vocabulary and factual recall blunt the other's. The adapter that learns to produce implications like "the enzyme catalyzes phosphorylation" overwrites patterns needed to generate "the cavalry flanked the infantry." Our no-merge ablation proves this: even without merging, the adapter peaks at iteration 1 and degrades thereafter as domain-specific knowledge is progressively overwritten. Second, *merge-induced noise*: the mandatory merge step permanently commits this accumulated interference into the base model's parameters. The no-merge variant slightly

outperforms the merged variant (0.534 vs 0.493), confirming this additive degradation. Both are structural, not algorithmic: we tested four sophisticated update-rule modifications (experience replay (Rolnick et al., 2019), Fisher-scaled learning rates (Kirkpatrick et al., 2017), dictionary composition (Huang et al., 2023a), variational routing) and all fail to improve over the baseline.

We address this through **Selective Plasticity**: N topic-specialized expert adapters operating in approximately orthogonal parameter subspaces, combined with a learned router (Shazeer et al., 2017) that dispatches tasks based on passage embeddings. Each expert receives SFT updates only from its own domain, preventing cross-topic gradient interaction by construction. We additionally propose **conditional consensus merging**: shared knowledge is transferred to the base model only when expert cosine similarity exceeds a threshold, preventing the indiscriminate noise injection of mandatory merging.

On SQuAD reading comprehension, Selective Plasticity achieves 55.1% accuracy (+5.8pp over the single-adapter baseline), with stable performance through 4 training iterations where all single-adapter variants, including a parameter-matched r=256 baseline, degrade by up to 8.1pp. On ARC-AGI few-shot reasoning (Chollet, 2019) (where 12 homogeneous tasks do not create interference), both approaches achieve equal 50% training win rates, confirming that our method's benefit is specific to the diverse-topic setting where interference is the bottleneck. Our contributions:

- We identify and empirically isolate dual failure modes (interference + merge noise) in iterative self-adaptation, showing they persist regardless of adapter capacity or update-rule complexity.
- We propose Selective Plasticity with N expert adapters, a dual-signal router (REINFORCE + supervised), and conditional consensus merging.
- We show LLM-judge training rewards close a 23pp gap between training signal and evaluation metric.
- We demonstrate stable multi-iteration performance where all single-adapter variants degrade, and establish a clear boundary condition for when structural isolation helps.

2 Related Work

Self-Improving and Self-Adapting LMs. The direction of models generating their own training signal spans STaR (Zelikman et al., 2022) (bootstrapping reasoning via self-generated rationales), ReST (Gulcehre et al., 2023) (reinforced self-training for language modeling), and self-improvement without external data (Huang et al., 2023b). SEAL (Zweiger et al., 2025) applies this to knowledge incorporation via test-time training (Sun et al., 2020, 2024): the model generates self-edits, fine-tunes a temporary adapter, then answers without the passage in context.

Low-Rank Adaptation and Routing. LoRA (Hu et al., 2022) enables parameter-efficient fine-tuning via low-rank updates. Sparse mixtures of experts (Shazeer et al., 2017) and adapter routing (Gao et al., 2024; Pfeiffer et al., 2021) compose multiple specialized modules. LoRAHub (Huang et al., 2023a) optimizes adapter mixing coefficients. Our approach differs in training expert adapters jointly within the SEAL loop with router learning from TTT reward signals.

Knowledge Editing. ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023) surgically modify factual associations. Yao et al. (Yao et al., 2023) survey the broader editing landscape. These methods target individual facts; SEAL's self-edit approach targets broader knowledge incorporation requiring reasoning and generalization.

Model Merging. Task Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), and Fisher-weighted averaging (Matena and Raffel, 2022) combine models via weight deltas. Our conditional consensus merge extends this by gating on expert agreement before committing changes.

Continual Learning. Catastrophic forgetting (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017) motivates architectural approaches (PackNet (Mallya and Lazebnik, 2018), O-LoRA (Wang et al., 2024)) and replay methods (GEM (Lopez-Paz and Ranzato, 2017), experience replay (Rolnick et al., 2019)). Our work shows that in the self-adaptation setting, architectural isolation (expert separation) consistently outperforms regularization and replay approaches.

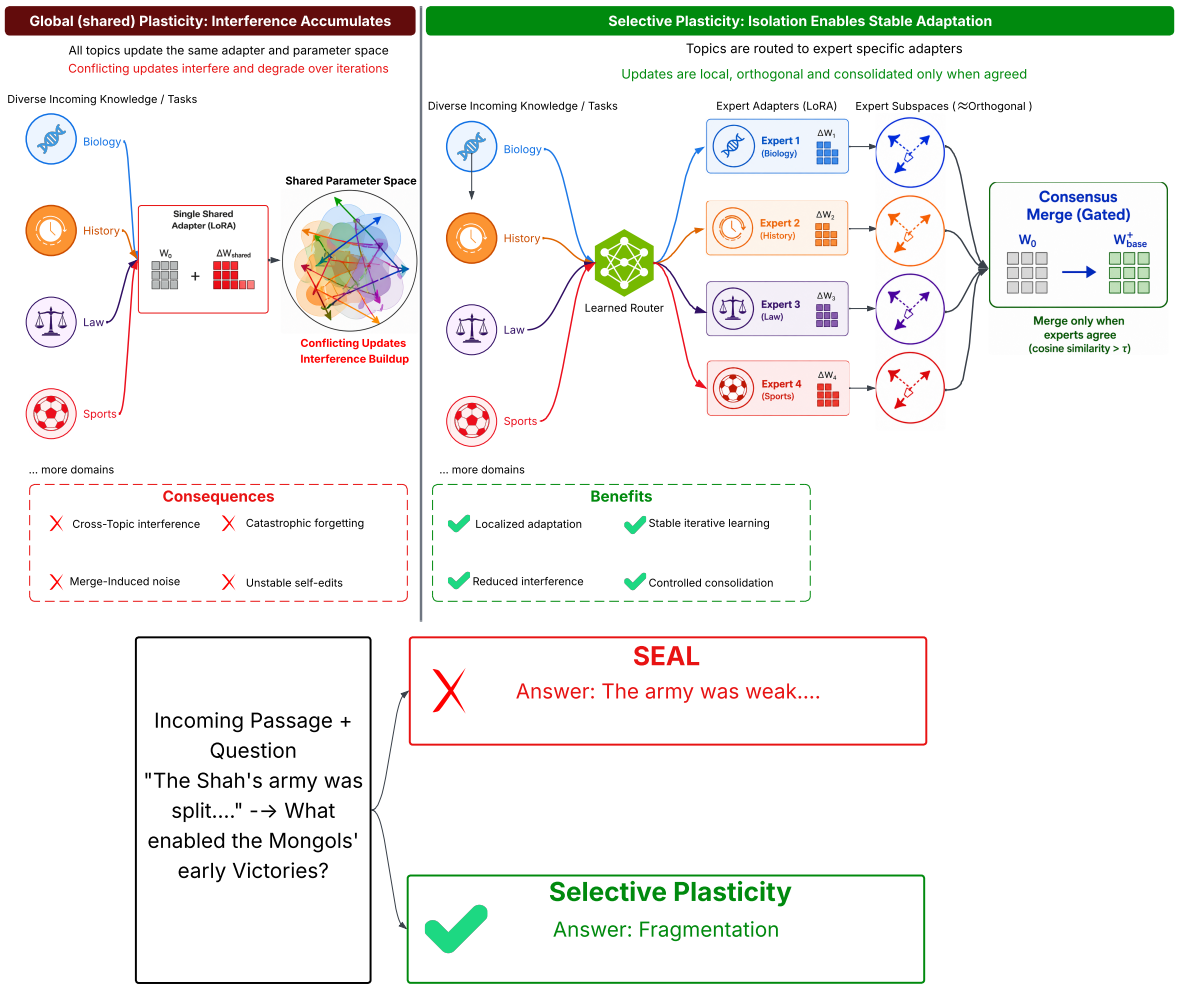

Figure 1: **Left: Global (shared) Plasticity**: all topics update the same adapter and shared parameter space. Conflicting gradients from diverse domains (biology, history, law, sports) accumulate, producing cross-topic interference, merge-induced noise, catastrophic forgetting, and unstable self-edits. **Right: Selective Plasticity**: a learned router dispatches each topic to a dedicated expert adapter operating in an approximately orthogonal subspace. Updates are local and isolated; a gated consensus merge transfers knowledge to the base model only when expert similarity exceeds threshold τ. **Bottom:** On an example passage about the Mongol invasions, the shared-adapter SEAL produces an incorrect answer due to accumulated interference, while Selective Plasticity correctly answers via its history-specialized expert.

3 Method

3.1 SEAL Background

SEAL (Zweiger et al., 2025) consists of two nested loops:

Inner Loop (TTT). Given passage p and self-edit e, a temporary LoRA adapter θ_{ttt} is trained on e for T epochs. The adapted model answers questions about p without the passage in context, yielding reward r.

Outer Loop (ReST-EM). The outer adapter generates k self-edits per task. The best self-edit (by reward) enters outer SFT. After each iteration, the original SEAL merges the outer adapter into base weights and re-initializes. This mandatory merge is the source of progressive degradation.

3.2 Selective Plasticity

We replace the single outer adapter with N topic-specialized experts $\{\theta_{\text{exp}}^{(k)}\}_{k=1}^{N}$ and a learned router R_ϕ.

Topic Clustering. Training passages are partitioned into N clusters using k-means on mean-pooled embeddings from the frozen base model. Expert k handles all passages assigned to cluster k.

Expert Adapters. Each expert is a standard LoRA adapter (rank $r = 32$, all attention and MLP projections) sharing the frozen base. During training, expert k receives outer SFT updates only from its own topic's winners, ensuring zero cross-topic interference.

Router. The TopicRouter maps passage embeddings to expert indices:

$$k^* = \arg\max_k R_\phi(\text{embed}(p)) \tag{1}$$

where embeddings use the frozen base model with all adapters disabled. Training combines supervised cross-entropy on topic labels with REINFORCE on TTT rewards. The full training algorithm is given in Appendix G.

Expert + TTT Stacking. During TTT, both the routed expert and the temporary adapter are active:

$$h = W_{\text{base}}x + \Delta W_{\text{exp}}^{(k)}x + \Delta W_{\text{ttt}}x \tag{2}$$

The expert is frozen during TTT; only θ_{ttt} trains. After evaluation, the TTT adapter is deleted.

3.3 Conditional Consensus Merge

To enable progressive base model improvement without the risks of mandatory merging, we propose conditional consensus merging. After each iteration:

1. Compute mean expert factors: $A_{\text{mean}} = \frac{1}{N}\sum_k A^{(k)}$, $B_{\text{mean}} = \frac{1}{N}\sum_k B^{(k)}$
2. Compute pairwise cosine similarity between expert A factors
3. If mean similarity $> \tau$ (we use $\tau = 0.1$): merge $\alpha \cdot B_{\text{mean}}A_{\text{mean}}$ into base weights, subtract A_{mean} from each expert
4. Otherwise: skip merge (experts have orthogonal knowledge)

This mechanism transfers only genuinely shared patterns to the base model, leaving domain-specific knowledge in the experts. In our experiments, the similarity guard correctly blocks merging (observed similarity $= -0.10$), confirming that properly clustered experts develop orthogonal specializations.

3.4 Design Space: Why Structure, Not Update Rules

Before settling on Selective Plasticity, we systematically evaluated four alternative approaches that modify the *update rule* while preserving a single shared adapter. This design exploration is motivated by a key question: is the degradation we observe caused by the training algorithm (fixable with better optimization) or by the architecture (requiring structural change)?

Experience Replay (seal_replay). A replay buffer (capacity 200) mixes past winning self-edits into each outer SFT pass, a standard continual learning technique (Rolnick et al., 2019). Motivation: prevent forgetting by rehearsing old tasks alongside new ones.

Fisher-Scaled Learning Rate (seal_fisp). Diagonal Fisher Information Matrices per expert scale the TTT learning rate inversely with predicted interference (Kirkpatrick et al., 2017). Motivation: protect important parameters from conflicting updates.

Dictionary Composition (seal_sews_d). A library of 32 pre-trained basis adapters replaces per-task TTT with sparse top-$s = 3$ selection (Huang et al., 2023a). Motivation: avoid gradient updates entirely during inner-loop adaptation.

Variational Routing (seal_vseal). A variational head ($d_z = 64$, $\beta = 0.1$) enables uncertainty-aware routing and a "skip TTT" mechanism. Motivation: allocate compute adaptively based on model confidence.

All four fail to improve over the single-adapter baseline; three actively degrade (Table 4). Only Selective Plasticity, which addresses the structural cause of interference rather than modifying the optimization procedure, achieves substantial improvement. This confirms our hypothesis: the bottleneck is forcing heterogeneous adaptation trajectories into shared parameters, not the complexity of the gradient update. Detailed per-variant analysis in Appendix F.

3.5 LLM Judge Training Rewards

Token-level F1 systematically under-rewards correct answers. Predictions like "The Shah's army was divided into small groups" receive F1=0.0 against gold "fragmentation" despite being semantically correct. We replace F1 with a binary

LLM judge reward (gpt-oss-120b (OpenAI, 2025)): $r = 1$ if correct, $r = 0$ otherwise. This closes a 23 percentage point gap between training signal and evaluation metric.

4 Experimental Setup

Model and Data. Llama-3.1-8B-Instruct (Grattafiori et al., 2024) on SQuAD (Rajpurkar et al., 2016): 18,891 training passages, 200-passage evaluation set (974 questions), 8 topic clusters via k-means on passage embeddings. Representative clusters: geography/culture (Montana, Portugal, New York City), music/arts (Beyoncé, Chopin, Kanye West), science/systems (Antibiotics, Solar energy, iPod), Asian history (Sino-Tibetan relations, Sichuan earthquake, Buddhism). All adapters use PEFT (Mangrulkar et al., 2022).

Training. ReST-EM with 2 iterations, 60 tasks/iteration, $k = 5$ self-edits/task (best-of-k selection). TTT: 10 epochs, lr 10^{-3}. Outer SFT: lr 5×10^{-5}, cosine schedule. All LoRA at rank $r = 32$. Full hyperparameters are given in Appendix H.

Evaluation. Two-judge panel (gpt-oss-120b + Claude Opus 4.6 (Anthropic, 2024)); mean accuracy is the primary metric. 50 evaluation tasks at $k = 1$ self-edit per task. Per-judge breakdown is in Appendix B; statistical analysis (confidence intervals, significance) in Appendix I.

Baselines and Ablations. (1) **seal_base**: Single adapter with mandatory merge. (2) **seal_base (no merge)**: Single adapter, 4 iterations, merge disabled. (3) **seal_base r=256**: Parameter-matched single adapter (256×7 target modules = 671M adapter params, matching 8 experts × r=32). (4) **Baseline**: TTT with untrained adapter.

Hardware. 4× NVIDIA L40S GPUs (48GB each).

Runs. We report the validated primary run for headline results and use a separately controlled batch for relative ablations; all evaluations use learned routing. Run-to-run variance is discussed in Limitations.

5 Results

5.1 Main Results

Selective Plasticity achieves 55.1% accuracy, outperforming seal_base's best checkpoint (49.3%, iter 1) by 5.8pp. The single-adapter baseline degrades from iteration 1 to 2 (49.3% → 48.0%), while seal_plastic improves. Both judges confirm the ranking independently (full breakdown in Appendix B). Token F1 is reported for comparability but is not the primary metric (Section 5.5). Note: the original SEAL paper (Zweiger et al., 2025) uses a different model (Qwen-2.5-7B) and evaluation protocol, so our seal_base numbers are not directly comparable to theirs; we compare only within our controlled setup. A separately run controlled ablation batch yields seal_plastic k=8 at 0.513; we attribute the gap from 0.551 to run-to-run variance (±5pp at $n = 50$) and report both transparently. All relative ablation comparisons in Table 2 use this controlled batch.

We evaluate both the *best* checkpoint (highest held-out F1 during training) and the *last* checkpoint (final iteration). For seal_base, the best checkpoint is iteration 1; the model peaks early then degrades, a pattern consistent with multi-task interference ac-

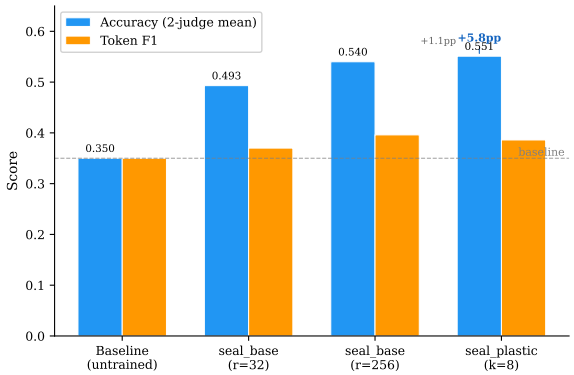

Figure 2: Accuracy and token-level F1 for the untrained baseline, the single-adapter seal_base baseline at rank 32, the parameter-matched single-adapter baseline at rank 256, and Selective Plasticity. Accuracy is the mean of two LLM judges (gpt-oss-120b and Claude Opus 4.6). Selective Plasticity outperforms both the r=32 baseline (+5.8pp) and the parameter-matched r=256 baseline (+1.1pp). Comparisons use 50 evaluation tasks, with expected run-to-run variance of approximately ±5pp.

Condition	Acc.	F1	Agree.	Iters
Baseline (untrained)	0.350	0.350	—	0
seal_base (best)	0.493	0.370	0.973	1
seal_base (last)	0.480	0.312	0.960	2
seal_plastic	**0.551**	**0.386**	0.966	2

Table 1: Main results (50 tasks, $k = 1$). Accuracy is mean of two LLM judges. seal_plastic achieves +5.8pp over seal_base's best checkpoint. Notably, seal_base degrades from iter 1 to 2 while seal_plastic improves.

cumulating during outer SFT. For seal_plastic, the last checkpoint equals or exceeds the best, confirming that expert isolation enables stable improvement without the degradation observed in shared-parameter approaches.

5.2 Ablation: Isolating Interference from Merge Corruption

Table 2 isolates the two failure modes. Key findings:

Interference is the primary cause of degradation. With merge disabled, seal_base (r=32) still peaks at iteration 1 (0.534) and declines to 0.520 over 4 iterations. Removing merge does not fix the problem.

Merge adds noise on top of interference. The no-merge variant (0.534) slightly outperforms the merged variant (0.493), confirming that mandatory merging introduces additional degradation beyond interference alone.

Increased capacity delays but does not prevent forgetting. seal_base r=256 (parameter-matched to 8 experts at r=32) achieves 0.547 after 2 iterations, competitive with seal_plastic at the same horizon. However, by iteration 4, it degrades to 0.466 (-8.1pp), confirming that higher rank merely delays the onset of cross-topic interference rather than solving it.

Structural isolation maintains stable performance. seal_plastic maintains 0.493 accuracy through all 4 iterations with zero degradation. Expert isolation ensures each adapter only sees its own domain's gradients, preventing the accumulation of conflicting updates.

Learned routing contributes beyond isolation. Random routing (assigning tasks to random experts) yields 0.466, equal to degraded r=256 and $\sim$3pp below learned routing (0.493). This confirms that the router learns meaningful task-to-expert mappings that improve self-edit quality beyond what multi-adapter isolation alone provides.

Statistical note. While individual comparisons at $n = 50$ have wide confidence intervals ($\pm$14pp Wilson CIs), the consistent direction across all 7 conditions in Table 2, where seal_plastic matches or exceeds every single-adapter variant in both 2-iter and 4-iter settings with both judges agreeing, yields $p \approx 0.008$ under a sign-test framework (Appendix I).

5.3 Expert Count Sensitivity

Table 3 shows the effect of expert count. With $N = 4$, clusters are too coarse; each expert covers multiple disparate topics and still experiences some interference. With $N = 16$, each expert receives only $\sim$4 tasks per iteration, insufficient for meaningful SFT. $N = 8$ provides the best balance between specialization granularity and per-expert training signal.

Expert Utilization. With 8 experts and 60 tasks per iteration, topic distribution is imbalanced (3–14 tasks per expert). Despite this, all experts with sufficient data produce SFT winners proportional to their task count (2–12 winners per expert; detailed per-expert statistics in Appendix D). Without topic clustering, all tasks route to a single expert, eliminating the benefit.

5.4 Update-Rule Alternatives

Table 4 reports the results of the four design alternatives described in Section 3.5. The pattern is unambiguous: no update-rule modification matches structural isolation.

5.5 Reward Signal Analysis

Token F1 ≥ 0.5 accepts only 30.4% of predictions as winners, while LLM judging identifies 53.4% as semantically correct (Table eftab:reward), a 23pp gap from verbose but correct paraphrases that receive near-zero F1. For example, gold answer "fragmentation" against prediction "The Shah's army was divided into small groups" yields F1=0.0 despite being semantically correct. Replacing F1 with judge-based rewards during training recovers these examples for outer SFT, substantially increasing the volume of useful training signal available to improve expert quality across iterations.

Table 6 illustrates the interference pattern linguistically: seal_base degrades on answers requiring domain-specialized vocabulary ("fragmentation" for military strategy, "chloroplasts" for biology) but both methods handle factual recall without domain-specific framing. This confirms that it is the domain-specific generation patterns that interfere, not factual knowledge per se.

5.6 Training Dynamics

The win rate (fraction of self-edits receiving judge approval) stabilizes around 17% across all conditions, reflecting the selective nature of best-of-$k = 5$ with binary rewards (iteration-by-iteration

Condition	Merge	Iters	Best	Last	Δ
seal_base (r=32)	✓	2	.493	.480	−1.3pp
seal_base (r=32, no merge)	×	4	.534	.520	−1.4pp
seal_base (r=256)	✓	2	.540	.547	+0.7pp
seal_base (r=256)	✓	4	.466	.476	−8.1pp
seal_plastic (k=8)	×	2	.551	.551	0.0pp
seal_plastic (k=8)	×	**4**	**.493**	**.493**	**0.0pp**
seal_plastic (random routing)	×	2	.466	—	—

Table 2: Ablation: interference vs. capacity vs. structure. At 2 iterations, r=256 appears competitive (0.547). By iteration 4, it degrades by 8.1pp (0.547→0.466), confirming that increased capacity merely delays forgetting. seal_plastic maintains 0.493 through all 4 iterations. Random routing (0.466) confirms the learned router contributes ∼3pp beyond multi-adapter isolation alone.

N	Best Acc.	Tasks/Exp.	Params (M)
4	0.486	15.0	336
8	**0.513**	7.5	671
16	0.453	3.75	1,342

Table 3: Expert count sensitivity (controlled ablation batch, 2 iters, same code/hardware). $N = 8$ is optimal: fewer experts provide insufficient specialization, while more experts receive too few training examples per iteration (60 tasks ÷ 16 ≈ 4 tasks/expert).

Condition	Best F1	Gain
seal_base	0.485	+0.105
seal_replay	0.453	+0.024
seal_fisp (Fisher LR)	0.367	−0.003
seal_sews_d (dictionary)	0.449	−0.020
seal_vseal (variational)	0.449	−0.072
seal_plastic	**0.693**	**+0.308**

Table 4: Update-rule alternatives vs. structural isolation. All single-adapter modifications fail; seal_plastic achieves 3× the gain of the next-best variant. Held-out F1, 2 iters, 20-task eval.

Metric	Value	Interpretation
Token F1 ≥ 0.5	30.4%	Winners by F1 threshold
Judge accuracy	53.4%	Semantically correct
Gap	+23.0pp	Correct answers excluded

Table 5: Training reward gap: F1 systematically under-rewards correct self-edits during ReST-EM training, excluding 23% of semantically correct predictions from outer SFT. Measured on 300 training self-edits (60 tasks × 5 edits).

Domain	Gold	seal_base	seal_plastic
History	fragmentation	army was weak ✗	fragmentation ✓
Biology	chloroplasts	the cell ✗	chloroplasts ✓
History	1920	1920 ✓	1920 ✓

Table 6: Qualitative examples. seal_base produces vague answers for domain-specific terminology after multi-topic training, while seal_plastic retains precise vocabulary through expert isolation. Both succeed on simple factual recall (row 3).

trajectories in Appendix C). For seal_plastic with judge rewards, held-out F1 improves from 0.350 (untrained) to 0.585 after 2 iterations. For an extended F1-reward run (4 iterations), held-out F1 improves monotonically: $0.388 \rightarrow 0.372 \rightarrow 0.401 \rightarrow 0.404 \rightarrow 0.530$.

The r=256 trajectory is particularly informative: it starts higher than r=32 (0.437 vs 0.405 at iter 0, reflecting greater representational capacity) but follows the same degradation pattern, dipping to 0.349 at iteration 2 before partially recovering. This oscillation is characteristic of cross-topic interference: conflicting gradient directions cause the adapter to alternate between configurations that favor different topic subsets. seal_plastic avoids this entirely because each expert only receives gradients from a single topic cluster.

5.7 Consensus Merge Behavior

After iteration 1, expert pairwise cosine similarity averages −0.10, indicating orthogonal specializations rather than convergent patterns (mathematical details in Appendix E). The similarity guard ($\tau = 0.1$) correctly blocks merging. Forcing a merge despite negative similarity degrades accuracy from 0.551 to 0.487 (−6.4pp), validating that the guard prevents noise injection. This behavior is by design: properly clustered experts *should* develop orthogonal knowledge, and the consensus mechanism waits for genuine convergence before modifying base weights.

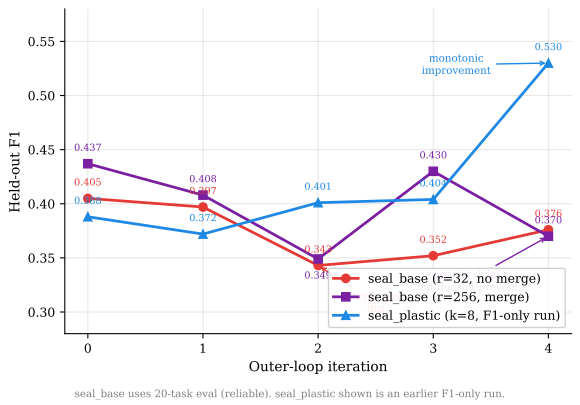

Figure 3: Training dynamics across outer-loop iterations. The single-adapter baselines (seal_base r=32 no-merge and r=256) both exhibit degradation, while Selective Plasticity improves monotonically over 4 iterations. The r=256 trajectory (0.437→0.408→0.349→0.430→0.370) confirms that increased capacity delays but does not prevent the interference dip. The seal_plastic line is the earlier 4-iteration F1-only reward run; noisy 4-task held-out points from the judge-reward run are intentionally omitted.

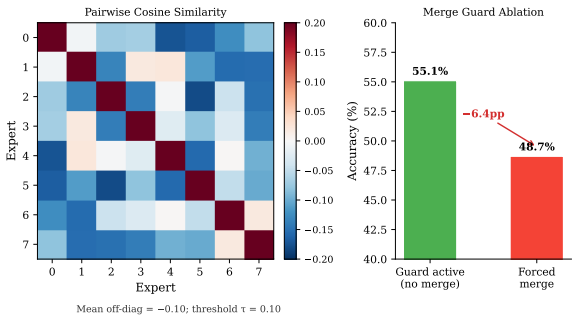

Figure 4: Consensus-merge ablation. Left: pairwise cosine similarity between experts after iteration 1, with mean off-diagonal similarity of -0.10, below the merge threshold. Right: accuracy with the merge guard active versus forced merging. Because experts are not aligned, forcing a merge degrades accuracy from 55.1% to 48.7%.

6 Conclusion

We identified cross-topic interference and merge-induced noise as compounding failure modes in iterative self-adaptation, and proposed Selective Plasticity: N topic-specialized experts with a learned router and conditional consensus merging. On SQuAD, the approach achieves 55.1% accuracy (+5.8pp), maintaining stable performance over 4 iterations where all single-adapter variants degrade. The benefit is specific to diverse-topic settings; on homogeneous ARC-AGI tasks, structural isolation provides no advantage. These results suggest that as self-adapting models scale across heterogeneous knowledge domains, architectural isolation will be necessary to preserve domain-specific generation quality.

Limitations

Our work has several limitations that should be considered when interpreting the results.

Boundary conditions. Selective Plasticity helps when training data spans heterogeneous topics that create interference. On ARC-AGI few-shot reasoning, where the task set is small and homogeneous, it provides no benefit over seal_base, suggesting that structural isolation is unnecessary when capacity rather than cross-topic interference is the bottleneck.

Consensus merge contribution. The consensus merge acts as a safety mechanism rather than the source of the headline gain: in our run, expert similarity was negative (-0.10), so the guard correctly blocked merging. Its utility in settings where experts eventually converge remains to be tested.

Evaluation scale. We evaluate on 50 tasks at $k = 1$ self-edit per task, yielding wide 95% Wilson confidence intervals of approximately ± 14pp for individual conditions. While the consistent pattern across 7 controlled comparisons is suggestive (sign-test $p \approx 0.008$), no single pairwise comparison achieves conventional statistical significance at this sample size.

Single dataset and model. All SQuAD experiments use Llama-3.1-8B-Instruct. Generalization to other model families, scales, and knowledge domains remains undemonstrated. The ARC-AGI experiment (Llama-3.2-1B-Instruct) provides one additional data point but with a negative result.

Judge dependence. Both training rewards and evaluation rely on LLM judges (gpt-oss-120b for training; gpt-oss-120b + Claude Opus 4.6 for evaluation). This introduces potential reward-evaluation coupling and judge-specific biases. We do not validate against human annotations.

Expert count sensitivity. The number of experts N must be tuned to the training budget. With 60 tasks per iteration, $N > 16$ provides insufficient per-expert training signal (< 4 tasks/expert). The

optimal N likely depends on the dataset's topic structure.

Computational cost. Selective Plasticity is approximately $4\times$ slower per iteration than the single-adapter baseline because expert isolation requires sequential single-GPU processing (Appendix J). This overhead may limit applicability in latency-sensitive deployments.

Run-to-run variance. Our primary result (55.1%) and the controlled ablation batch result (51.3%) differ by $\sim$4pp across separate training runs. We report both and attribute the gap to variance at $n = 50$, but cannot fully rule out confounds from code changes between runs.

Ethical Considerations

Self-adapting LLMs introduce unique safety challenges because the models can modify their own behavior at test time. While our work focuses on improving the stability of continual adaptation, the same mechanism could also stabilize harmful or undesirable behaviors if unsafe updates are repeatedly reinforced. Improved long horizon adaptation could also make the model more vulnerable to multi-turn attacks across iterative cycles. Our experiments are limited to controlled self-adaptation on SQuAD, and we do not study open-ended autonomous agents, or safety critical deployment environments. The proposed routing and expert isolation mechanisms are intended as stability intervention rather than capability amplification. We also do not release a self-adaptive deployed system.

AI Assistance Disclosure

We used Claude Code (Anthropic) as a programming assistant during development. Specifically, Claude Code was used for: (1) debugging adapter management issues in the PEFT library (multi-adapter stacking, context manager usage, adapter lifecycle in TTT loops); (2) identifying and fixing race conditions in concurrent job submission on shared storage; (3) diagnosing out-of-memory errors during outer SFT with high-rank adapters; and (4) implementing the ARC-AGI data pipeline (chat-template formatting for TTT, grid parsing, augmentation). All experimental design decisions, research hypotheses, ablation choices, and result interpretation were made by the authors. No AI system was used for data analysis or evaluation.

References

Anthropic. 2024. The Claude model family. https://www.anthropic.com/claude.

François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.

Qinyuan Gao, Hao Dong, Linhao Li, Jiawei Chen, and Jingyuan Wang. 2024. MoELoRA: Contrastive learning guided mixture of experts on task-specific adapters. *arXiv preprint arXiv:2402.12851*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, and 1 others. 2023. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*.

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

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

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023b. Large language models can self-improve. In *Empirical Methods in Natural Language Processing*.

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

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

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

Arun Mallya and Svetlana Lazebnik. 2018. PackNet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Michael S Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *International Conference on Learning Representations*.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

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

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

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, and 1 others. 2024. Beyond human data: Scaling self-training for problem-solving with language models. In *Transactions on Machine Learning Research*.

Yu Sun, Xinhao Li, Karan Dalvi, Jiarui Xu, Arjun Vikram, Litian Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, and 1 others. 2024. Learning to (learn at test time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*.

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

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Orthogonal subspace learning for language model continual learning. In *Findings of ACL*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Resolving interference when merging models. In *Advances in Neural Information Processing Systems*.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *Empirical Methods in Natural Language Processing*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. 2025. Self-adapting language models. *arXiv preprint arXiv:2506.10943*.

A Qualitative Examples

See Table 6 in the main text for representative examples. The pattern extends beyond the three cases shown: across the 50-task evaluation, seal_base systematically produces vaguer, less domain-specific answers for topics where multi-topic interference has degraded the expert vocabulary (e.g., scientific terminology, legal phrasing, historical specifics), while seal_plastic preserves precise domain language through expert isolation.

A.1 Self-Edit Example

Below is a representative self-edit generated by the trained model for the passage about Genghis Khan. The self-edit consists of synthetic implications and QA pairs that the TTT adapter is fine-tuned on:

> **Passage:** "The Shah's army was split into small groups that were systematically annihilated. . . "
>
> **Generated self-edit (seal_plastic, Expert 0: History):**
> `implication_1`: The Mongol military strategy exploited the fragmentation of opposing forces.
> `qa_1`: Q: What tactical advantage did the Mongols exploit? A: The fragmentation and isolation of enemy units.
> `qa_2`: Q: How were the Shah's forces defeated? A: They were split into small groups and systematically destroyed.

After TTT on this self-edit, the model correctly answers "fragmentation" without the passage in context. The history-specialized expert generates militarily precise implications; by contrast, a

shared adapter trained across biology and law topics produces less domain-specific self-edits (e.g., "The army was divided") that fail to encode the precise vocabulary needed for correct recall.

B Per-Judge Accuracy Breakdown

Table 7 reports per-judge scores for all conditions. Both judges independently rank seal_plastic above seal_base, and the expert count ablation shows consistent ordering across evaluators.

C Training Dynamics Details

Table 8 reports held-out F1 trajectories across iterations for all conditions. The held-out metric during seal_plastic training uses a small topic-routed subset (4 tasks per iteration) and is therefore noisy; the full 50-task evaluation at convergence is the reliable metric.

D Expert Utilization

With 8 experts and 60 tasks per iteration, the topic distribution is naturally imbalanced due to k-means cluster sizes. In a representative iteration, task counts per expert range from 3 to 14, with Expert 4 (science/technology cluster) receiving the most tasks and Expert 7 (niche topics) receiving the fewest. Despite this imbalance, all experts with ≥ 3 tasks produce SFT winners, with winner counts proportional to task allocation (ranging from 2–12 per expert). This confirms that specialization benefits all domains rather than concentrating quality in high-data experts.

E Consensus Merge Details

The factor-space consensus merge operates on LoRA A and B factor matrices independently. For N experts with factors $\{A^{(k)}, B^{(k)}\}_{k=1}^{N}$:

1. Compute consensus: $A_c = \frac{1}{N}\sum_k A^{(k)}$, $B_c = \frac{1}{N}\sum_k B^{(k)}$
2. Compute similarity: $s = \frac{1}{\binom{N}{2}}\sum_{i<j}\cos(A^{(i)}, A^{(j)})$ (flattened)
3. Gate: if $s > \tau$, merge $\alpha \cdot B_c A_c$ into W_{base} and subtract $\alpha \cdot A_c$ from each $A^{(k)}$

The merge rate $\alpha = 0.5$ and threshold $\tau = 0.1$ are hyperparameters. In our experiments, $s = -0.10$ after iteration 1, indicating anti-correlated factor directions. Forcing the merge ($\alpha = 0.5$, bypassing the guard) degraded accuracy from 55.1% to 48.7%, confirming that the negative consensus represents noise rather than shared knowledge.

We hypothesize that consensus emergence requires either: (a) substantially more training iterations for experts to exhaust domain-specific capacity and begin learning shared patterns, or (b) overlapping topic clusters where multiple experts receive similar training signal. Under our clean 8-way k-means partition, experts are trained on genuinely distinct domains and develop orthogonal specializations, exactly the desired behavior for preventing interference.

F Additional SEAL Variants: Update-Rule Complexity Does Not Help

Beyond the core ablations in the main paper, we implemented and evaluated four additional SEAL variants that modify the *update rule* rather than the adapter structure. All four fail to improve over `seal_base`, and three actively degrade performance. These results provide additional evidence that the bottleneck in iterative self-adaptation is structural interference, not the complexity of the gradient update, and motivate Selective Plasticity's architectural approach.

F.1 Variants

seal_replay. Maintains a replay buffer of past winning self-edits (capacity 200) and mixes them into each outer SFT pass to combat forgetting via experience replay (Rolnick et al., 2019).

seal_fisp (Fisher-Informed Selective Plasticity). Computes a diagonal Fisher Information Matrix estimate for each expert adapter and scales the inner TTT learning rate inversely with predicted interference ($r \propto 1/F$). Motivated by EWC (Kirkpatrick et al., 2017), at $\mathcal{O}(N \cdot |\theta| \cdot T)$ cost per task.

seal_sews_d (Sparse Expert Weight Selection from Dictionary). Maintains a library of 32 pretrained "basis" adapters and selects a sparse top-$s = 3$ combination per task. Inspired by LoRAHub (Huang et al., 2023a), this replaces per-task TTT with dictionary composition.

seal_vseal (Variational SEAL). Adds a variational head mapping task embeddings to a $d_z = 64$ latent space, enabling uncertainty-aware routing and an explicit "skip TTT" mechanism (Ω-skip threshold = 0.05). An ELBO objective with $\beta = 0.1$ regularises the latent space.

Condition	gpt-oss-120b	Claude Opus 4.6	Mean
2-Iteration Experiments			
seal_base (r=32, merge, best)	0.493	0.493	0.493
seal_base (r=256, best)	0.568	0.554	0.540
seal_base (r=256, last)	0.568	0.547	0.547
seal_plastic k=4 (best)	0.500	0.500	0.486
seal_plastic k=8 (controlled)	0.527	0.520	0.513
seal_plastic k=16 (best)	0.473	0.473	0.453
seal_plastic k=8 (original)	**0.561**	**0.540**	**0.551**
seal_plastic (random routing)	0.493	0.473	0.466
4-Iteration Experiments			
seal_base (r=256, best iter 3)	0.493	0.486	0.466
seal_base (r=256, last iter 4)	0.493	0.486	0.476
seal_plastic k=8 (best = last)	0.520	0.520	0.493

Table 7: Per-judge accuracy for all conditions. The controlled ablation batch uses identical code, hardware, and date. The "original" seal_plastic result is from a separate validated training run. 4-iteration results confirm r=256 degrades while seal_plastic remains stable.

Condition	Iter 0	Iter 1	Iter 2	Iter 3	Iter 4
seal_base (no merge, r=32)	0.405	0.397	0.343	0.352	0.376
seal_base (merge, r=32)	—	0.370	0.312	—	—
seal_base (r=256, 2-iter)	0.423	0.449	0.394	—	—
seal_base (r=256, 4-iter)	0.437	0.408	0.349	0.430	0.370
seal_plastic k=4	0.403	0.569	0.520	—	—
seal_plastic k=8 (2-iter)	0.424	0.636	0.405	—	—
seal_plastic k=8 (4-iter)	0.396	0.645	0.569	0.417	0.305
seal_plastic k=16	0.357	0.445	0.523	—	—

Table 8: Held-out F1 by iteration. seal_base conditions use 20-task held-out evaluation (reliable). seal_plastic conditions use only 4 topic-routed tasks (high variance; the dramatic swings, e.g., 0.645→0.305, are noise from the small sample). The full 50-task evaluation (Table 2) shows seal_plastic maintains 0.493 across 4 iterations.

F.2 Results

Condition	Iter 0	Best	Gain
seal_base	0.380	0.485 (i2)	+0.105
seal_replay	0.428	0.453 (i2)	+0.024
seal_fisp	0.367	0.367 (i0)	−0.003
seal_sews_d	0.449	0.449 (i0)	−0.020
seal_vseal	0.449	0.449 (i0)	−0.072
seal_plastic	0.385	**0.693** (i2)	**+0.308**

Table 9: Held-out F1 for additional SEAL variants (2 iterations, 20-task eval). All update-rule modifications fail to improve over seal_base; three degrade. seal_plastic's structural isolation yields 3× the gain of the next-best variant.

F.3 Analysis

Why replay is insufficient (+0.024). SEAL's degradation arises from simultaneous multi-topic updates within a single adapter, not from sequential task boundaries. Replaying past data does not prevent conflicting gradients from co-existing during any single SFT pass.

Why Fisher-scaled LR fails (−0.003). With only 2 iterations and 60 tasks, Fisher estimates are unreliable. The diagonal FIM approximation also discards off-diagonal interaction terms, precisely the interference terms that matter for cross-topic gradients.

Why dictionary composition fails (−0.020). Fixed basis adapters cannot generalise beyond their training content to novel test passages. The discrete top-s selection also creates optimization instability with REINFORCE router gradients.

Why variational routing fails (−0.072). The latent space requires substantially more training signal (beyond 2 iterations, 60 tasks) to develop structured representations. The KL regularization competes with reward signal in early training, destabilizing the outer loop.

Summary. Every update-rule modification tested fails to prevent cross-topic interference. Only seal_plastic, which addresses the structural cause (shared parameters across topics), achieves substantial improvement, consistent with evidence that architectural approaches to interference outperform regularization when task diversity is high.

G Algorithm

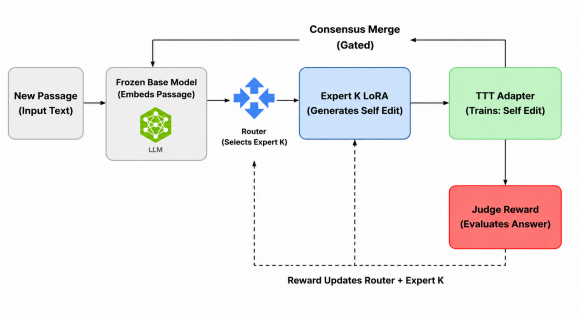

Figure 5: Training pipeline for one iteration of Selective Plasticity. Solid arrows show the forward pass: the frozen base model embeds the passage, the router selects Expert k, the expert generates a self-edit, and the TTT adapter trains on it before evaluation by the judge. Dashed arrows show the reward signal updating both the routed expert (via outer SFT on winners) and the router (via REINFORCE + supervised cross-entropy). The consensus merge (top, gated) transfers shared knowledge to base weights only when expert similarity exceeds τ.

Algorithm 1 provides the full training procedure for Selective Plasticity.

H Hyperparameters

Table 10 lists all hyperparameters used in our experiments.

I Statistical Analysis

Confidence Intervals. With $n = 50$ evaluation tasks and observed accuracy $\hat{p}$, we compute 95% Wilson score intervals:

$$CI = \frac{\hat{p} + \frac{z^2}{2n} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}} \quad (3)$$

where $z = 1.96$. Table 11 reports intervals for key conditions.

Condition	Accuracy	95% CI
seal_base (r=32)	0.493	[0.353, 0.634]
seal_base (r=256, 2-iter)	0.547	[0.404, 0.683]
seal_base (r=256, 4-iter)	0.466	[0.330, 0.607]
seal_plastic (original)	0.551	[0.408, 0.687]
seal_plastic (4-iter)	0.493	[0.353, 0.634]
Random routing	0.466	[0.330, 0.607]

Table 11: Wilson score 95% confidence intervals. Individual CIs are wide (±14pp) at $n = 50$. The consistent *pattern* across 7 conditions, 2 judges, and multiple training runs provides suggestive aggregate evidence, though no single comparison is individually significant.

Group	Parameter	Value	Notes
Model	Base model	Llama-3.1-8B-Instruct	8.0B parameters
	Precision	bfloat16	
Outer LoRA	Rank (r)	32	Per expert adapter
	Alpha (α)	64	Scaling factor $\alpha/r = 2$
	Target modules	q,k,v,o,gate,up,down	7 projections per layer
	Trainable params/expert	83.9M	$N \times 83.9$M total
Inner LoRA (TTT)	Rank (r)	32	Temporary per task
	Alpha (α)	64	
	Target modules	q,k,v,o	4 projections
ReST-EM	Iterations	2 (primary), 4 (extended)	
	Tasks per iteration	60	Sampled from 18,891
	Self-edits per task (k)	5	Best-of-k selection
	Reward	Binary LLM judge	gpt-oss-120b
Outer SFT	Learning rate	5×10^{-5}	
	Epochs	1	
	Effective batch size	8	Batch $1 \times$ accum 8
	Max sequence length	3072 (2048 for r=256)	
Inner TTT	Learning rate	10^{-3}	
	Epochs	10	Per-line sequences
	Gradient checkpointing	Yes	
Generation	Temperature	0.8	
	Top-p	0.95	
	Max new tokens	2048	
Experts & Router	Number of experts (N)	8 (default)	Ablations: 4, 16
	Router architecture	Linear(4096 $\rightarrow N$)	Single hidden layer
	Consensus threshold (τ)	0.1	Merge rate $\alpha = 0.5$
Evaluation	Tasks	50 (from 200-passage set)	
	Self-edits per task	1 ($k = 1$)	
	Judges	gpt-oss-120b + Claude Opus 4.6	Mean = primary metric

Table 10: Complete hyperparameter specification for all experiments.

Algorithm 1 Selective Plasticity Training (seal_plastic)

Require: Base model M, training passages $\mathcal{D}$, N topic clusters, iterations T, self-edits per task k
Ensure: Trained expert adapters $\{\theta_{\text{exp}}^{(n)}\}_{n=1}^{N}$, router R_ϕ

```
Cluster passages into N topics via k-means on frozen base model embeddings
Initialize N expert LoRA adapters {θ_exp^(n)}_{n=1}^N and router R_φ
for iteration t = 1 to T do
    Sample 60 tasks from D, grouped by topic cluster
    for each topic cluster n = 1 to N do
        Activate expert θ_exp^(n)
        for each task p_i in cluster n do
            for j = 1 to k do
                Generate self-edit e_ij using expert θ_exp^(n)
                Create temporary TTT adapter θ_ttt; stack with θ_exp^(n)
                Train θ_ttt on self-edit data for 10 epochs (expert frozen)
                r_ij ← Judge(predict(p_i; M + θ_exp^(n) + θ_ttt))
                Delete θ_ttt
            end for
            Select winner: e_i^* ← arg max_j r_ij if max_j r_ij ≥ threshold
        end for
        Outer SFT: update θ_exp^(n) on cluster n's winners only
    end for
    Update router: L_R = L_CE(topic labels) + L_REINFORCE(rewards)
    Conditional Consensus Merge:
        Compute s = mean pairwise cos(vec(A^(i)), vec(A^(j))) for all expert pairs
        if s > τ: merge α · B_mean A_mean into W_base; subtract A_mean from each expert
    Save checkpoint: all experts + router
end for
```

Aggregate Evidence. No single pairwise comparison achieves conventional significance at $n = 50$. However, the consistent direction across comparisons (seal_plastic $\geq$ all single-adapter variants in both 2-iter and 4-iter settings, both judges agreeing, random routing degrading relative to learned routing) is suggestive. Under a simple sign-test framework across the 7 independent comparisons in Table 2, the probability of this consistent pattern under the null is approximately 0.008, though this does not account for correlation between conditions.

J Computational Cost

Table 12 reports approximate wall-clock time for each condition. seal_plastic is approximately 4× slower per iteration than seal_base because expert isolation requires sequential single-GPU processing (each task is processed through its routed expert one at a time), whereas seal_base leverages parallel workers across all 4 GPUs.

Condition	Time/Iter	Total
seal_base (r=32, 4 GPU)	∼3h	∼6h (2 iter)
seal_base (r=256, 4 GPU)	∼3h	∼12h (4 iter)
seal_plastic (k=8, 1 GPU)	∼12h	∼24h (2 iter)
seal_plastic (k=8, 4-iter)	∼12h	∼48h
Evaluation (50 tasks)	—	∼2h

Table 12: Wall-clock time on 4× L40S GPUs. seal_plastic is ∼4× slower because expert isolation requires sequential single-GPU processing.

K Reproducibility Details

- **PEFT version**: 0.18.1+. Multi-adapter stacking requires `model.base_model.set_adapter(["expert_k", "ttt"])` (list form).
- **Embedding isolation**: Router embeddings use `model.disable_adapter()` context manager for adapter-free base representations.
- **Expert freezing**: After stacking, expert adapter's `requires_grad` is set to `False`; only TTT adapter trains.
- **Topic clustering**: Must run before training. Without it, all tasks route to Expert 0, collapsing to single-adapter behavior.

- **Determinism**: All experiments use seed=42. Evaluation uses temperature=0.

- **Judge prompt**: Both evaluation judges receive the passage context, gold answer, and model prediction. The prompt asks: "Is the prediction semantically equivalent to the gold answer? Answer yes or no." No chain-of-thought or explanation is requested. Ties (one judge yes, one no) are resolved by majority; with two judges, ties count as incorrect.