SIFT: BOUNDARY-FREE ONLINE ADAPTATION VIA SEMANTIC ISOLATION AND FUNCTIONAL TESTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Boundary-free online continual learning aims to adapt large language models to a stream of evolving data without access to task identities or boundaries. Without this supervision, the learner must decide from the stream alone when incoming data can safely reuse existing adaptation capacity and when it requires separate capacity. Sharing one set of adaptation parameters across the stream causes interference and forgetting, whereas allocating separate parameters typically relies on externally provided task boundaries and can increase parameter counts unnecessarily. We present SIFT (**S**emantic **I**solation and **F**unctional **T**esting), a framework for boundary-free online adaptation that keeps the language model backbone frozen and organizes the stream into mature semantic modules and provisional modules. Because nearby regions can still require conflicting updates, semantic proximity alone cannot certify that adaptation capacity can be shared, so SIFT defers the decision until functional evidence is available. It routes examples to mature modules by cosine distance, isolates uncertain ones into provisional modules that inherit a parent prompt but adapt independently, and consolidates a provisional module only when a radius-based compatibility check and a sentinel-based functional validation agree. We conduct extensive experiments using four instruction-tuned backbones from the Llama and Qwen families, spanning 0.6B to 7B parameters. SIFT achieves up to 21.96% relative improvement in final accuracy over the strongest baseline while substantially reducing forgetting.

1 INTRODUCTION

Large language models increasingly serve applications receiving information from changing environments, including interactive assistants, document processing systems, and specialized agents. As these systems operate over time, their input distributions evolve and new data can require capabilities that were not represented during initial training. Updating the model online is therefore important for maintaining useful performance without repeatedly performing expensive offline retraining. Many traditional continual learning(CL) formulations (De Lange et al., 2021) assume that task identities or reliable task boundaries are available, allowing the learner to switch task-specific components or apply task-specific update rules when the data distribution changes. Although this supervision simplifies adaptation, it is often unavailable in open-ended applications. Figure 2.(A) contrasts the task identities and boundary cues available in traditional CL with their absence in Boundary-Free CL. Boundary-free online CL removes this assumption: the learner processes a potentially non-stationary data stream sequentially, without access to task identities or task boundaries (Aljundi et al., 2019; Koh et al., 2023). Although boundary-free online CL has received substantial attention in computer vision (Aljundi et al., 2019; Wei et al., 2025), it remains comparatively underexplored in large language models despite recent progress (Han et al., 2026).

This setting creates a difficult stability and plasticity tradeoff for large language models. The learner may observe each example only once, cannot rely on future data, and may have no access to past training examples. Online adaptation requires frequent model updates, making full-parameter fine-tuning computationally expensive for large language models. Parameter-efficient adaptation offers a less costly alternative by training only a small prompt (Lester et al., 2021) or a low-rank adapter (Hu et al., 2021) while keeping the backbone fixed. Yet updating one shared adaptation state changes the parameters used by all previously encountered data, which can cause interference and forgetting. Maintaining a separate state for every emerging data source can protect earlier knowledge,

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.

but it requires reliable change detection and can allocate capacity whenever a new source appears, even when it could have safely reused an existing state. The central challenge is therefore to decide when an incoming example should share adaptation capacity with prior data and when it should be isolated. Many existing CL methods protect prior knowledge by estimating parameter importance and penalizing changes to parameters deemed important for previous tasks (Kirkpatrick et al., 2017; Zenke et al., 2017). In their task-based formulations, known task boundaries determine when importance estimates are consolidated and regularization constraints are updated. These boundaries are unavailable in our boundary-free online setting, so the same consolidation rules cannot be directly applied to the incoming stream. This motivates adaptation mechanisms that preserve prior knowledge without relying on externally specified task transitions.

We propose SIFT (Semantic Isolation and Functional Testing), a framework for boundary-free online adaptation that combines semantic organization with sentinel-based functional validation. The framework maintains a collection of mature adaptation modules and provisional branches for uncertain cases. Source representations provide an efficient first decision for routing incoming examples to semantically compatible modules. When the routing decision is uncertain, a provisional branch can adapt in isolation without immediately changing the mature module that serves earlier data. After accumulating sufficient evidence, the branch is resolved using both semantic overlap and a separate functional check of its effect on a small sentinel set from the parent module. Only a branch that is compatible in representation space and does not degrade the sentinel performance is merged with the parent, while other branches remain separate. This design allows the learner to reuse adaptation capacity when reuse is safe and to preserve prior knowledge when semantic similarity is misleading, all without task id, task boundaries, or replay.

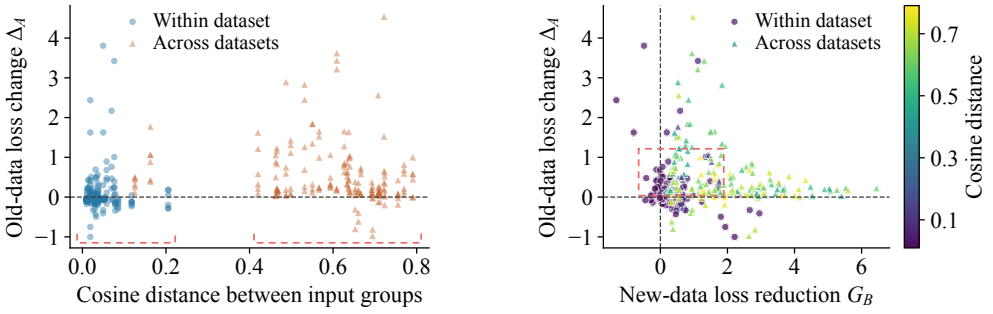
We evaluate SIFT using multiple backbones with parameter counts ranging from 0.6B to 7B. Across model families and parameter scales, SIFT achieves competitive Stage Acc and Final Acc while reducing forgetting relative to shared adaptation baselines. These results support the effectiveness of combining adaptive semantic routing with functional validation to balance adaptation and knowledge retention in boundary-free online CL. Our contributions are as follows:

1. We introduce SIFT, which combines adaptive semantic routing and isolated provisional adaptation with semantic compatibility checks and sentinel-based functional validation. This design mitigates the stability–plasticity tradeoff in online CL by guiding module reuse and isolation without task identities or task boundaries.
2. We show empirically that negative transfer is not confined to different tasks: adaptation to new data can degrade previously acquired knowledge even within the same task. This finding highlights a limitation of using task identity to guide parameter sharing and motivates a more fine-grained assessment of adaptation compatibility.
3. We systematically evaluate SIFT across Llama and Qwen models spanning 0.6B–7B parameters, assessing adaptation and knowledge retention in boundary-free online CL. The results show that SIFT achieves up to 21.96% relative improvement in Final Acc over the strongest baseline while substantially reducing forgetting.

2 Related Work

2.1 Continual Learning

Continual learning studies how a model can learn from a sequence of changing data distributions while retaining previously acquired knowledge. Conventional continual learning methods (Wang et al., 2024) commonly address this problem through rehearsal, regularization, or parameter isolation. Replay-based methods (Rebuffi et al., 2017; Shin et al., 2017; Rolnick et al., 2019) reuse past data to mitigate forgetting. For example, Gradient Episodic Memory (GEM) (Lopez-Paz & Ranzato, 2017) uses stored examples to constrain gradient updates. Regularization-based methods preserve parameters or parameter importance associated with previous knowledge, including elastic weight consolidation and synaptic intelligence (Kirkpatrick et al., 2017; Zenke et al., 2017). Parameter-isolation methods (Zhou et al., 2024) instead allocate separate components for different tasks or data sources. These strategies can reduce catastrophic forgetting, but they often rely on task structure, task boundaries, or gradient updates on stored historical data, requirements that cannot be met in our task setting. By contrast, boundary-free online continual learning (Aljundi et al., 2019) removes the need for task identities and predefined task boundaries. Examples arrive sequentially,



(a) Semantic distance and interference (b) Adaptation gain and interference

Figure 1: Cosine distance between groups is a coarse but insufficient guide to whether an update can be shared. Each point is one update: a prompt is adapted on a source input group and then updated on a target input group, and marker color separates within-dataset from across-dataset pairs. (a) Groups from the same dataset are generally closer in semantic space. (b) Updates that reduce the new-data loss G_B can still raise Δ_A, and these cases (red box) occur within as well as across datasets.

and the learner must infer distribution changes and determine how to update its state directly from the incoming stream. Recent work has explored loss dynamics, parameter importance, prompt-pool inference, and information-geometric signals to guide boundary-free adaptation (Han et al., 2026; Wang et al., 2022). Much of the work on boundary-free online continual learning has focused on computer vision (Lee et al., 2020; Jin et al., 2021; Wei et al., 2025), while its application to language models remains comparatively underexplored. To help address this gap, we investigate how large language models can adapt in this setting while mitigating interference both within and across tasks.

2.2 Knowledge Sharing and Parameter Isolation

Knowledge sharing allows related objectives to benefit from common representations, while parameter isolation provides separate capacity for objectives that may interfere. Multi-task learning (Fan et al., 2022; Ye & Xu, 2023) studies this balance through architectures that combine shared and task-specific components. Cross-stitch networks (Misra et al., 2016) learn how to combine activations from separate task networks, allowing the degree of sharing to vary across layers. Multi-gate Mixture-of-Experts (MMoE) (Ma et al., 2018) shares expert networks across tasks while using a separate gating network for each task to control their contributions. Progressive Layered Extraction (Tang et al., 2020) explicitly separates shared and task-specific experts and progressively routes information between them to mitigate negative transfer. The effectiveness of sharing also depends on how shared parameters are updated. PCGrad (Yu et al., 2020) addresses this issue by projecting conflicting task gradients to reduce destructive interference during joint optimization. These methods facilitate knowledge sharing and mitigate interference through architectural design or gradient adjustment. However, they organize sharing and interference control around explicitly defined tasks, typically assuming offline access to data from all tasks at once. Our work considers knowledge sharing and isolation across tasks while also accounting for opportunities for sharing and potential interference among examples within the same task, in an online continual learning setting where examples arrive sequentially rather than being jointly available.

3 Preserving Prior Knowledge Beyond Semantic Similarity

Reusing adaptation parameters can enable knowledge sharing, but updates on incoming data may compromise performance on previously learned data. Allocating separate parameters avoids direct interference but can prevent beneficial sharing and increase capacity requirements. We examine whether task identity, which prior boundary-aware continual learning methods use to decide when parameters can be shared, and semantic similarity, which is computable from the stream, provide reliable guidance for deciding when adaptation can be shared without degrading prior performance.

We perform an offline pairwise diagnostic on six task datasets. For each dataset, we sample 256 examples and divide them into four groups of 64 based on the similarity of their source representations (Eq.(4)). Each group is randomly split into 32 training and 32 validation examples. In each trial, we fit a soft prompt (Lester et al., 2021) to the training examples of group A and then update it on those of group B. Let V_A and V_B denote held-out validation sets for the two groups, used only to measure

the effect of the update. With P_A and P_{AB} denoting the prompt states before and after that update, we measure

$$\Delta_A = \mathcal{L}(P_{AB}, V_A) - \mathcal{L}(P_A, V_A), \qquad G_B = \mathcal{L}(P_A, V_B) - \mathcal{L}(P_{AB}, V_B). \tag{1}$$

The two quantities carry opposite desired signs: $G_B > 0$ indicates that the update improves target performance, whereas $\Delta_A > 0$ indicates that it degrades source performance.

We compute each group centroid as the normalized mean of its training examples' source representations. Figure 1(a) shows that cosine distance between the centroids of groups A and B provides a useful coarse signal. Same dataset pairs cluster at small distances, whereas different dataset pairs are generally farther apart. Figure 1(b) shows why this signal is insufficient for safe adaptation. In the Llama-3.2-1B-Instruct diagnostic, $\Delta_A > 0$ occurs in 35.6% of same dataset and 92.2% of different dataset trials. The red box in the right panel highlights adaptation cases that interfere with prior task performance. Such cases occur in both different dataset and same dataset pairs. This observation suggests that distinguishing different tasks is not sufficient. We must also avoid data that negatively affect model performance within the same task.

Both must be computed online, where the learner sees neither task boundaries nor the group labels used in this diagnostic. The next section introduces SIFT, which builds its semantic regions incrementally and defers the share-or-isolate decision to a functional check on retained examples from the region that would be overwritten in a boundary-free continual learning setting.

4 METHOD

4.1 PROBLEM SETTING AND LEARNING OBJECTIVE

In the boundary-free online continual learning setting, a learner receives a stream of text pairs $\mathcal{S} = \{(x_t, y_t)\}_{t=1}^{T}$, where x_t and y_t denote the source and target text sequences, respectively. Each pair (x_t, y_t) is drawn from the current distribution $\mathcal{D}_t$, which may shift at unknown times. Examples arrive sequentially, and the learner receives neither task identities nor task boundaries. Let ϕ_t denote the adaptation state after processing example t, that is, the parameters the learner carries across the stream, with ϕ_0 the initial state. And let f_{ϕ_t} be the predictor it induces. An online update rule $\mathcal{A}$ produces $\phi_t = \mathcal{A}(\phi_{t-1}, x_t, y_t)$ from the current example and from this carried-over state, without future examples. In SIFT, ϕ_t is instantiated by the soft prompts and region statistics of Section 4.2.

The goal is to learn from the current distribution while retaining performance on previously encountered distributions. Weighting each arrival equally, the risk of state ϕ on the distributions encountered by time t is

$$\mathcal{R}_t(\phi) = \frac{1}{t} \sum_{s=1}^{t} \mathbb{E}_{(x,y)\sim\mathcal{D}_s} \left[\ell(f_\phi(x), y) \right], \tag{2}$$

where ℓ is the prediction loss. A general online continual learning objective minimizes this risk throughout the stream

$$\min_{\mathcal{A}} \mathbb{E}_{\mathcal{S}} \left[\frac{1}{T} \sum_{t=1}^{T} \mathcal{R}_t(\phi_t) \right], \qquad \phi_t = \mathcal{A}(\phi_{t-1}, x_t, y_t), \tag{3}$$

We minimize over online update rules $\mathcal{A}$ satisfying the sequential information constraint above. The expectation is over stream realizations. The term $s = t$ measures adaptation to the current distribution. Terms $s < t$ measure earlier knowledge retention. These population risks define the learning objective. The learner observes stream examples without direct access to distributions $\mathcal{D}_s$.

4.2 SIFT OVERVIEW

SIFT keeps the backbone parameters θ frozen and places all online adaptation in a small collection of semantic modules, each pairing a soft prompt with the source-space region it serves. Semantic isolation allows provisional modules to adapt without modifying their mature parents, while functional testing checks whether geometrically proposed replacements preserve performance on the corresponding parents' sentinel sets, the examples retained for each module (Section 4.4). Each stream example is used for gradient learning only when it arrives. Sentinel examples are used only to validate proposed module fusions, never for additional gradient updates. We represent the core state of each semantic module as $C_k = (P_k, \mu_k, n_k)$, where $P_k \in \mathbb{R}^{L_p \times d}$ is a trainable soft prompt consisting of L_p token embeddings, d is the backbone hidden dimension, $\mu_k \in \mathbb{R}^d$ is a normalized

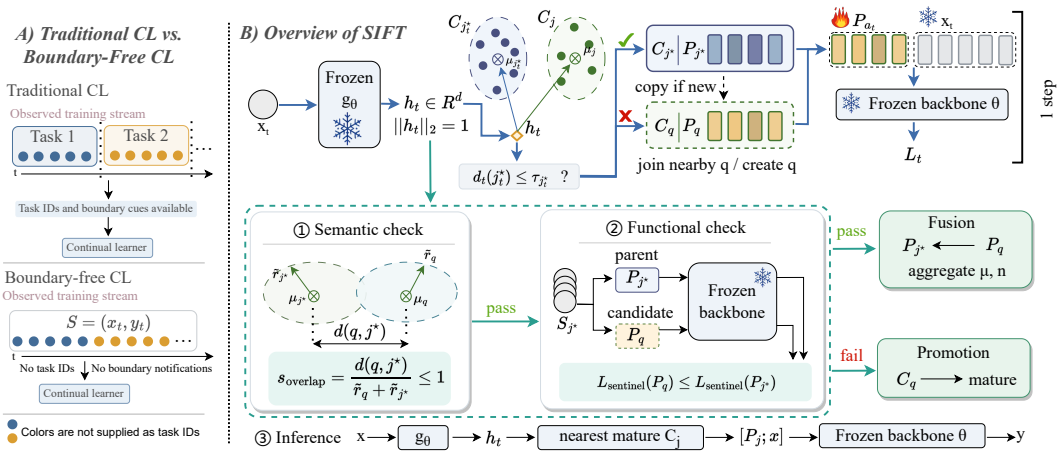


Figure 2: (A) Traditional CL uses task identities and boundary cues, while Boundary-Free CL observes a task-free stream without boundary notifications. (B) Overview of SIFTwith source-only routing, isolated prompt updates, semantic and functional checks, and mature-prompt inference.

centroid in the source representation space, and n_k is the module's sample count. SIFT maintains two kinds of modules. A mature module C_j represents an established semantic region. A provisional module C_q adapts independently with its own prompt P_q and centroid μ_q, pending a decision to merge it into its parent or retain it as a separate mature module.

Figure 2.(B) depicts the lifecycle of one stream example. As shown in the upper part of Figure 2(B), SIFT first extracts the normalized source representation h_t for each incoming example (x_t, y_t). The example is routed to an accepted mature module whenever possible, using h_t for the similarity test. Otherwise, it joins a nearby provisional module or, failing that, SIFT creates a new one. A new provisional module inherits the prompt of its nearest mature parent, but its copied prompt is subsequently optimized in isolation. Exactly one selected prompt receives one answer token gradient update on (x_t, y_t), after which h_t is incorporated into the selected module's online statistics.

The copied prompt P_q is then updated independently of its parent, so a provisional module can adapt without changing the mature module that serves earlier data. Once the module has received enough examples, it is confirmed and becomes a candidate for resolution. During the initial seed phase, confirmed candidates are promoted directly to establish a small set of mature semantic modules before fusion is considered. After this initial set is established, candidate resolution applies two checks. As illustrated in Figure 2.①, the semantic check compares the candidate and parent using their centroids and semantic radii. As shown in Figure 2.②, the functional check compares both prompts on the parent's sentinel set. Fusion is accepted only if both the semantic check and the functional check pass. Otherwise, the already trained candidate is promoted to a separate mature module if capacity permits; in this work, we set the maximum number of mature modules to 384. Promotion requires no additional training. The following sections detail routing and adaptation (Section 4.3) and candidate resolution with module fusion (Section 4.4).

4.3 Mature Module Routing and Online Adaptation

Source representation. For each source sequence x_t, we run the frozen backbone without a soft prompt or the target sequence and extract the hidden state of the last valid source token from the penultimate decoder layer. Let $g_\theta(x_t) \in \mathbb{R}^d$ denote this state, where d is the backbone hidden dimension. The routing representation is detached and normalized as

$$h_t = \frac{g_\theta(x_t)}{\|g_\theta(x_t)\|_2}, \qquad h_t \in \mathbb{R}^d, \quad \|h_t\|_2 = 1. \tag{4}$$

Semantic routing. Each semantic module C_k is characterized by a normalized centroid μ_k and a soft prompt P_k. Each module also tracks its sample count n_k and statistics of sample-to-centroid distances to guide routing and fusion. For the first example assigned to C_k, we initialize $\mu_k = h_t$ and $n_k = 1$. Subsequent assignments update the centroid and count after prompt optimization:

$$\mu_k \leftarrow \frac{n_k \mu_k + h_t}{\|n_k \mu_k + h_t\|_2}, \qquad n_k \leftarrow n_k + 1, \tag{5}$$

where the right-hand side uses the centroid and sample count before the update. We write $\mathcal{M}_t = \{k \mid C_k \text{ is mature}\}$ and $\mathcal{Q}_t = \{k \mid C_k \text{ is provisional}\}$ for the index sets just before routing example t. When $\mathcal{M}_t = \emptyset$, the first example initializes a mature seed module C_0. Otherwise, for any module index $c \in \mathcal{M}_t \cup \mathcal{Q}_t$, define the cosine distance

$$d_t(c) = 1 - \mu_c^\top h_t. \tag{6}$$

Let $j_t^\star = \arg\min_{j \in \mathcal{M}_t} d_t(j)$ be the nearest mature module. Even the nearest module may be semantically distant, so an acceptance threshold prevents assigning an unfamiliar example to an existing module solely because it is the closest. Each mature module tracks the online mean $\bar{d}_j$ and standard deviation σ_j of its observed distances. Because these statistics are unreliable with few observations, a mature module uses the fixed threshold $\tau_{\min}$ until it has accumulated K_{warm} distance observations. Once this warm-up count is reached, it uses

$$\tau_j = \text{clip}\big(\bar{d}_j + \beta\sigma_j, \tau_{\min}, \tau_{\max}\big)\,. \tag{7}$$

Here clip restricts the threshold to $[\tau_{\min}, \tau_{\max}]$. The threshold adapts to the module's observed spread, with β controlling how much variation it allows. Let a_t be the selected module. If $d_t(j_t^\star) \leq \tau_{j_t^\star}$, then $a_t = j_t^\star$ and the example revisits that mature module.

If the mature module test fails and $\mathcal{Q}_t \neq \emptyset$, we choose the nearest provisional module

$$q_t^\star = \arg\min_{q \in \mathcal{Q}_t} d_t(q). \tag{8}$$

If $d_t(q_t^\star) \leq \tau_{\min}$, then $a_t = q_t^\star$ and the example joins $C_{q_t^\star}$. If no provisional module passes this test, including when $\mathcal{Q}_t$ is empty, we create C_q and set $a_t = q$. For this provisional module, let $j^\star = j_t^\star$ denote its mature parent selected at creation, as in Figure 2. We initialize $P_q \leftarrow P_{j^\star}$, and subsequent optimization of C_q does not modify the parent module $P_{j^\star}$.

Updating one selected prompt. Each module a owns a trainable soft prompt $P_a \in \mathbb{R}^{L_p \times d}$ consisting of L_p virtual tokens. With the shared backbone frozen, we update only the selected prompt P_{a_t} using the mean negative log likelihood of the answer tokens

$$\mathcal{L}_t = -\frac{1}{|\mathcal{I}_t|} \sum_{v \in \mathcal{I}_t} \log p_\theta\left(y_{t,v} \mid x_t, y_{t,<v}, P_{a_t}\right), \tag{9}$$

where $\mathcal{I}_t$ indexes answer token positions. The soft prompt is prepended to the source embeddings. Source tokens and soft prompt positions are excluded from the loss. We evaluate $\mathcal{L}_t$ before the update and take one optimizer step using only the current example. The centroids, distance statistics, and routing decisions receive no gradients. Updating a provisional prompt P_q leaves its parent prompt $P_{j^\star}$ unchanged.

We then incorporate h_t into the selected module's centroid and sample count. For a mature revisit, we also update its distance statistics using the distance measured before the centroid update. For a provisional module, we retain h_t for the subsequent fusion check.

4.4 Candidate Resolution and Module Fusion

Confirmation and seed phase. A provisional module C_q is confirmed once it has received at least K_{conf} examples, meaning $n_q \geq K_{\text{conf}}$. It then becomes a candidate for fusion or promotion.

To establish the initial semantic regions, SIFT begins with a seed phase within the same online stream. The first example initializes the mature module C_0. Confirmed candidates are then promoted directly, without fusion checks, until the number of mature modules reaches K_{seed}. After this phase, the geometry and sentinel checks described below determine whether a candidate should merge into its parent or be promoted to a separate mature module.

Semantic check. We refer to this radius-based compatibility test as the semantic check, since it assesses compatibility through the geometry of the modules' semantic representations. We use a semantic radius to summarize the spread of each module in cosine distance. For a mature module j, we approximate the root mean square radius using its online distance moments

$$r_j = \sqrt{\sigma_j^2 + \bar{d}_j^2}. \tag{10}$$

For a candidate q, let $\mathcal{H}_q$ denote the list of normalized representations assigned to it, with $n_q = |\mathcal{H}_q|$. Its raw radius is the root mean square distance to its current centroid.

$$r_q = \sqrt{\frac{1}{n_q} \sum_{h \in \mathcal{H}_q} \left(1 - \mu_q^\top h\right)^2}. \tag{11}$$

The mature radius summarizes historical distances as its centroid evolved, whereas the candidate radius is computed directly from its retained representations.

To reduce sensitivity to small sample counts, we shrink both radii toward statistics across mature modules. Let r_{global} be the lower median of positive mature module radii and n_{global} the lower median of mature module sample counts. For $z \in \{q, j^\star\}$, let r_z and n_z denote its raw radius and sample count, respectively. The shrunk radius is

$$\widetilde{r}_z = \frac{n_z r_z + n_{\text{global}} r_{\text{global}}}{n_z + n_{\text{global}}}. \tag{12}$$

Small modules receive more weight from the global radius, while larger modules rely more on their own estimated spread. For candidate q and fixed parent $j^\star$, the center distance and overlap score are

$$d(q, j^\star) = 1 - \mu_q^\top \mu_{j^\star}, \qquad s_{\text{overlap}} = \frac{d(q, j^\star)}{\widetilde{r}_q + \widetilde{r}_{j^\star} + \varepsilon}. \tag{13}$$

The geometry check passes when the radius sum is positive and $s_{\text{overlap}} \leq 1$. As illustrated in Figure 2.①, this asks whether the centroids are close relative to the modules' estimated spread. Here $\varepsilon > 0$ is a small numerical constant. A zero radius sum fails the geometry check.

Functional check. As discussed in Section 3, semantic proximity does not guarantee that merging the corresponding prompt modules will preserve their performance. We therefore introduce a functional check to evaluate the effect of merging on model performance. SIFT therefore retains a small sentinel set of past input and target pairs for each mature module. These examples are used only to compare the parent and candidate prompts, with no gradient updates.

For a geometry eligible candidate q with fixed parent $j^\star$, let $S_{j^\star} = \{s_1, \ldots, s_m\}$ denote its parent's m sentinel examples. Following Figure 2.②, we write $L_{\text{sentinel}}(P)$ for the mean answer token loss of prompt P on this fixed set

$$L_{\text{sentinel}}(P) = \frac{1}{m} \sum_{i=1}^{m} \ell_{\text{ans}}(s_i, P). \tag{14}$$

Here $\ell_{\text{ans}}(s_i, P)$ is the mean answer token loss of example s_i evaluated with prompt P, as in Eq.(9). The functional check passes if the candidate prompt's mean loss is no larger than the parent's.

Fusion or promotion. After the seed phase, a candidate with a nonempty parent sentinel set and finite validation losses is fused into its parent exactly when

$$\text{accept fusion} \iff \begin{cases} \text{geometry check passes,} \\ L_{\text{sentinel}}(P_q) \leq L_{\text{sentinel}}(P_{j^\star}). \end{cases} \tag{15}$$

When fusion is accepted, the parent takes the already trained candidate prompt and aggregates the two semantic regions. Using the counts before fusion, the state transfer is

$$P_{j^\star} \leftarrow P_q, \qquad \mu_{j^\star} \leftarrow \frac{n_{j^\star} \mu_{j^\star} + n_q \mu_q}{\left\| n_{j^\star} \mu_{j^\star} + n_q \mu_q \right\|_2}, \qquad n_{j^\star} \leftarrow n_{j^\star} + n_q. \tag{16}$$

The parent also incorporates the candidate's distance observations and validation records, and the provisional module is removed. If fusion is rejected, the already trained (μ_q, P_q) is retained as a separate mature module. Neither fusion nor promotion performs an additional training step. When fusion is rejected but mature module capacity is exhausted, C_q remains provisional. Resolution is checked again after a later example is assigned to it. Direct fusion between mature modules is never performed. Appendix E gives a counterexample to safe sharing based only on semantic similarity and a conditional bound on the historical loss change caused by a single fusion.

Table 1: Main results on the boundary-free continual learning benchmark. We report stage accuracy (Stage Acc), final accuracy (Final Acc), and backward transfer (BWT) for three backbones. All scores are scaled to $[0, 100]$. Higher Stage Acc and Final Acc are better, and higher (less negative) BWT indicates less forgetting. Bold numbers indicate the best result for each metric and backbone.

	Llama 3.2 1B			**Llama 3.2 3B**			**Qwen3 0.6B**		
Method	Stage Acc	Final Acc	BWT	Stage Acc	Final Acc	BWT	Stage Acc	Final Acc	BWT
Prompt Tuning (Lester et al., 2021)	19.05	17.20	-22.41	28.72	21.76	-34.81	24.31	18.37	-29.41
L2P (Wang et al., 2022)	25.18	24.67	-23.12	29.91	20.31	-36.80	24.70	18.57	-28.75
MAS (Aljundi et al., 2019)	21.84	21.06	-7.69	29.38	25.27	-12.54	23.92	26.51	**-0.05**
Online LoRA (Wei et al., 2025)	19.25	18.61	-21.81	20.28	25.45	-11.05	19.55	19.10	-18.08
GRID (Tiwari et al., 2025)	24.52	17.96	-31.85	27.32	21.44	-34.00	24.04	17.04	-31.36
FiUni (Han et al., 2026)	23.57	26.36	-5.97	35.27	34.15	-9.72	30.79	28.75	-10.83
F2OCL (Wang et al., 2025)	37.47	34.98	-8.12	**46.01**	45.76	-5.70	38.80	37.61	-4.54
strLoRA (Che et al., 2026)	37.48	34.40	-15.19	45.54	41.51	-11.74	36.17	30.36	-19.76
SIFT (Ours)	**42.12**	**42.66**	**-1.73**	45.80	**45.91**	**-3.95**	**40.16**	**39.48**	-4.63

Inference. As shown in Figure 2.③, we first encode an input x into its normalized representation h using Eq.(4). We then select the mature module C_j whose centroid is closest to h in cosine distance and retrieve its prompt P_j. We prepend P_j to the input embeddings and use the frozen backbone to generate the answer y autoregressively.

5 Experiments

We evaluate SIFT on six datasets using Llama and Qwen backbones from 0.6B to 7B parameters in a single-pass, boundary-free stream. Setup details are in Appendix B.

5.1 Main Results

Main Experimental Results. Table 1 shows that SIFT achieves the highest Final Acc on all three backbones and leads seven of the nine comparisons across metrics and backbones. Its Final Acc gains over F2OCL, the strongest baseline in Final Acc, are 7.68%, 0.15%, and 1.87% on Llama-3.2-1B-Instruct, Llama-3.2-3B-Instruct, and Qwen3-0.6B, respectively.

On Llama-3.2-1B-Instruct, SIFT also improves Stage Acc by 4.65% and raises BWT from -8.12 to -1.73. These gains suggest that adapting provisional prompts independently helps the model learn from new data while preserving previously learned behavior. New provisional branches inherit previously learned prompts but adapt independently of their parents, allowing new behavior to develop without directly overwriting the parent parameters. In contrast, Prompt Tuning and Online LoRA update a single shared parameter set on every arriving example, so each update directly overwrites the state that encodes earlier behavior. This is reflected in their BWT of -22.41 and -21.81 on this backbone, the most severe forgetting among all methods except L2P. On Llama-3.2-3B-Instruct, SIFT and strLoRA achieve similar Stage Acc (45.80 vs. 45.54), yet SIFT obtains 4.40% higher Final Acc and 7.79% higher BWT. strLoRA achieves competitive Stage Acc across backbones but forgets substantially, with BWT between -11.74 and -19.76, since it decides where to share updates from semantic similarity alone. Against F2OCL, SIFT also improves BWT by 1.75% at nearly matched Stage Acc. This retention advantage is consistent with the motivation for functional testing, since semantic proximity alone cannot establish whether a candidate preserves earlier behavior. After the seed phase, consolidation requires both semantic overlap and no increase in loss on the parent's sentinel set. Candidates that fail these checks remain separate, subject to capacity. This mechanism checks update compatibility using earlier examples without additional gradient training on them. On Qwen3-0.6B, SIFT improves Stage Acc and Final Acc over F2OCL by 1.36% and 1.87%, respectively, with only 0.09% lower BWT. Its advantage here is higher accuracy at comparable forgetting. This pattern fits the complementary roles of selective reuse and isolation, as compatible examples update existing modules while provisional branches support separate adaptation. These results suggest that allowing provisional modules to adapt independently and checking their compatibility with earlier knowledge before merging helps the model learn from new data while retaining previously learned behavior.

Performance on a Larger Backbone. Table 2 extends the comparison to Llama-2-7B-Chat, where SIFT achieves the best reported result in all three metrics. Its Stage Acc of 39.94 exceeds strLoRA, the strongest baseline on this metric, by 0.38%, while its Final Acc of 41.43 exceeds F2OCL by 2.67%. Compared with F2OCL, SIFT also improves BWT by 2.84%, from -2.59 to $+0.25$.

Although strLoRA achieves a similar Stage Acc (39.56), its lower Final Acc (34.80) and more negative BWT (-13.39) indicate greater degradation of previously learned performance. These results suggest that SIFT maintains its balance between adaptation and retention at a larger model scale.

5.2 Ablation Study

Table 3 compares five ablated variants with the full SIFT model on Llama-3.2-1B-Instruct. *W/o geometry check* removes the semantic compatibility test before fusion while retaining routing and sentinel validation. *W/o functional validation* permits geometrically compatible candidates to fuse without checking their loss on the parent's sentinel examples. *W/o fusion* trains provisional modules and makes them separate mature modules once confirmed, without merging them into their parents. *W/o provisional modules* directly creates mature modules from individual examples without inheriting parent prompts, bypasses provisional confirmation and fusion, and updates the nearest mature module when capacity is exhausted. *W/o parent initialization* initializes provisional prompts afresh instead of inheriting the parent prompt. We follow the same evaluation setup as the main experiments.

Table 2: Larger-backbone comparison on Llama-2-7B-Chat. Higher is better for all three metrics. Bold numbers indicate the best result.

Method	Stage Acc	Final Acc	BWT
Prompt Tuning (Lester et al., 2021)	27.09	23.29	-24.62
L2P (Wang et al., 2022)	26.98	20.78	-28.40
Online LoRA (Wei et al., 2025)	20.78	15.64	-24.50
GRID (Tiwari et al., 2025)	27.14	25.44	-21.92
FiUni (Han et al., 2026)	17.04	22.11	+0.03
F2OCL (Wang et al., 2025)	39.12	38.76	-2.59
strLoRA (Che et al., 2026)	39.56	34.80	-13.39
SIFT (Ours)	**39.94**	**41.43**	**+0.25**

Table 3: Ablation results on Llama-3.2-1B-Instruct. All runs include all six evaluation stages. Higher is better for all three metrics.

Variant	Stage Acc	Final Acc	BWT
w/o geometry check	40.64	38.85	-3.47
w/o functional validation	39.79	38.59	-3.95
w/o fusion	42.00	40.63	-3.46
w/o provisional modules	30.62	28.52	**-0.25**
w/o parent initialization	38.72	37.70	-3.21
SIFT (Ours)	**42.12**	**42.66**	-1.73

The full SIFT model achieves the highest Final Acc of 42.66 and a BWT of -1.73. Retaining functional validation without geometry slightly outperforms retaining geometry alone across all metrics, suggesting that checking earlier behavior provides value beyond semantic compatibility. The full model's stronger final performance favors combining both checks. Without provisional modules, Final Acc is lowest at 28.52. This variant creates mature modules from individual examples without provisional confirmation or parent prompt inheritance, suggesting that establishing modules with sufficient evidence and prior knowledge is important for effective adaptation. Its BWT of -0.25 also shows that limited average forgetting alone does not imply strong learning performance. Without parent initialization, Final Acc is 37.70, suggesting that inherited knowledge helps adaptation in a single pass stream. Without fusion, confirmed modules remain separate and achieve the highest Stage Acc among the ablated variants at 42.00. However, their Final Acc is 40.63 and BWT is -3.46, compared with 42.66 and -1.73 for the full model. This pattern suggests that keeping modules separate maintains comparable Stage Acc, while fusion guided by both compatibility checks improves final performance and reduces forgetting.

6 Conclusion

We introduced SIFT, which enables boundary-free online adaptation of large language models through semantic isolation and functional testing. SIFT routes incoming examples to semantic prompt modules, adapts provisional branches independently of their parents, and combines semantic overlap checks with sentinel-based validation to decide whether branches should be fused or retained separately. Experiments across four Llama and Qwen backbones show that SIFT improves final average performance and reduces forgetting compared with shared adaptation baselines. This work advances research on boundary-free online continual learning for LLMs, with SIFT achieving up to 21.96% relative improvement in Final Acc over the strongest baseline. It offers a broader perspective on interference beyond task boundaries by highlighting negative transfer within a task.

7 AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks with required disclosure; all such tasks, including data generation, methodology or experiment design, method implementation, and result interpretation, were carried out by the authors, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure, namely to aid and polish the writing, including drafting assistance and editing to improve readability, grammar, and clarity. We have reviewed all AI-assisted work; all AI-generated text was carefully checked, verified, and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11246–11255. IEEE, 2019.

Chang Che, Ziqi Wang, Hui Ma, Cheems Wang, and Zenglin Shi. Strlora: Towards streaming continual visual instruction tuning for mllms. *arXiv preprint arXiv:2605.16353*, 2026.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, Zhangyang Wang, et al. M^3vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. *Advances in Neural Information Processing Systems*, 35:28441–28457, 2022.

Annette Rios Gonzales, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. A new dataset and efficient baselines for document-level text simplification in german. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pp. 152–161, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. Meetingbank: A benchmark dataset for meeting summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16409–16423, 2023.

Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. *Advances in Neural Information Processing Systems*, 34: 29193–29205, 2021.

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

Hyunseo Koh, Minhyuk Seo, Jihwan Bang, Hwanjun Song, Deokki Hong, Seulki Park, Jung-Woo Ha, and Jonghyun Choi. Online boundary-free continual learning by scheduled data prior. In *The Eleventh International Conference on Learning Representations*, 2023.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. *arXiv preprint arXiv:2001.00689*, 2020.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, et al. Codexglue: A machine learning benchmark dataset for code understanding and generation. *arXiv preprint arXiv:2102.04664*, 2021.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1930–1939, 2018.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3505–3523, 2022.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3994–4003, 2016.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5533–5542. IEEE, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In *Proceedings of the 14th ACM conference on recommender systems*, pp. 269–278, 2020.

Anushka Tiwari, Sayantan Pal, Rohini K Srihari, and Kaiyi Ji. Grid: Scaling task-agnostic inference in continual prompt tuning. *arXiv preprint arXiv:2507.14725*, 2025.

Aopeng Wang, Ke Deng, Yongli Ren, and Jun Luo. Rehearsal-free and task-free online continual learning with contrastive prompt. *arXiv preprint arXiv:2510.00467*, 2025.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383, 2024.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 139–149. IEEE, 2022.

Xiwen Wei, Guihong Li, and Radu Marculescu. Online-lora: Task-free online continual learning via low rank adaptation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6634–6645. IEEE, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Hanrong Ye and Dan Xu. Taskexpert: Dynamically assembling multi-task representations with memorial mixture-of-experts. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21771–21780. IEEE, 2023.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. Pmlr, 2017.

Chenye Zhao, Yingjie Li, and Cornelia Caragea. C-stance: A large dataset for chinese zero-shot stance detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13369–13385, 2023.

Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23554–23564. IEEE, 2024.

A PSEUDOCODE

A module $C_k = (P_k, \mu_k, n_k)$ has a soft prompt $P_k \in \mathbb{R}^{L_p \times d}$, normalized centroid μ_k, and sample count n_k. A mature module stores its sentinel set $\mathcal{S}_k$; a provisional module stores the representations $\mathcal{H}_k$ and validation records $\mathcal{R}_k$ collected before resolution, and has a fixed parent $j(q)$. Each mature module also maintains online distance statistics with mean $\bar{d}_j$ and standard deviation σ_j. $\mathrm{UpdateDistanceStats}(j, \delta)$ incorporates one distance δ into the online mean $\bar{d}_j$ and standard deviation σ_j; $\mathrm{ResetDistanceStats}(j)$ clears the distance count, mean, and variance while preserving the sample count n_j. Retain_B and $\mathrm{IsoStep}$ denote the budgeted retention rule and the isolated single-prompt step, respectively. Algorithm 1 summarizes the online procedure.

B EXPERIMENTAL SETUP

Benchmark. We evaluate in a boundary-free continual learning setting for LLMs on a heterogeneous benchmark composed of six NLP datasets: MeetingBank (Hu et al., 2023), Py150 (Lu et al., 2021), NumGLUE-cm (Mishra et al., 2022), NumGLUE-ds (Mishra et al., 2022), 20Minuten (Gonzales et al., 2021), and C-STANCE (Zhao et al., 2023). The collection spans meeting summarization, code completion, mathematical reasoning, German text simplification, and Chinese stance classification. Our experiments cover four instruction-tuned LLM backbones from the Llama and Qwen families, spanning 0.6B to 7B parameters: Llama-3.2-1B-Instruct (Grattafiori et al., 2024), Llama-3.2-3B-Instruct, Qwen3-0.6B (Yang et al., 2025), and Llama-2-7B-Chat. During training, examples from all datasets are presented through a single non-stationary stream and processed one at a time, with each example seen only once. The learner receives no dataset identities, task boundaries, or per-task sample quotas. Dataset-level performance is computed only for offline evaluation and never used to guide training decisions.

Evaluation protocol. After training on each stream segment, the model is evaluated on the test sets of all segments seen so far. Each dataset is scored with its standard metric. We use ROUGE L for MeetingBank, similarity for Py150, accuracy for NumGLUE-cm, NumGLUE-ds, and C-STANCE, and SARI for 20Minuten. All scores are scaled to $[0, 100]$ before aggregation. At each evaluation checkpoint r, we measure the model's performance on the datasets observed so far. Let $a_{r,i}$ denote the score on dataset i at checkpoint r, and let R denote the total number of evaluation checkpoints. We report three aggregate metrics. Stage accuracy is Stage Acc $= \frac{1}{R}\sum_{r=1}^{R}\frac{1}{r}\sum_{i=1}^{r} a_{r,i}$, final accuracy is Final Acc $= \frac{1}{R}\sum_{i=1}^{R} a_{R,i}$, and backward transfer is $\mathrm{BWT} = \frac{1}{R-1}\sum_{i=1}^{R-1}(a_{R,i} - a_{i,i})$.

Implementation details. We evaluate four instruction-tuned backbones spanning 0.6B to 7B parameters, namely Llama-3.2-1B-Instruct (Grattafiori et al., 2024), Llama-3.2-3B-Instruct, Qwen3-0.6B (Yang et al., 2025), and Llama-2-7B-Chat. All backbones remain frozen, and only lightweight adaptation parameters are trained. All methods process the stream with a batch size of 1 and use AdamW at a default constant learning rate of 2×10^{-4}, with source and answer length budgets of 1,024 and 512 tokens, respectively. Unless otherwise specified, we use the same SIFT configuration across all four backbones. AdamW is configured with $(\beta_1, \beta_2) = (0.9, 0.999)$, a numerical stability constant of 10^{-8}, and zero weight decay. SIFT processes the stream once and performs one optimizer step per arriving example, without gradient accumulation or learning-rate warmup. We clip the gradient ℓ_2 norm to 5.0 and disable dropout. Each module owns a soft prefix of $L_p = 100$ virtual tokens, and we reserve 512 prompt slots to accommodate at most $K_{\mathrm{m}} = 384$ mature modules and $K_{\mathrm{q}} = 128$ concurrent provisional modules. Only the selected module's prefix is prepended to the input and updated on the current example. The initial prompt embeddings are sampled from $\mathcal{N}(0, 0.02^2)$, matching the backbone's initialization standard deviation. Provisional modules are confirmed after receiving $K_{\mathrm{conf}} = 4$ examples, and the seed phase ends when $K_{\mathrm{seed}} = 3$ mature modules have been established. Semantic routing uses threshold bounds of $\tau_{\min} = 0.16$ and $\tau_{\max} = 0.35$, with a standard-deviation multiplier of $\beta = 4.0$. The overlap score in Eq. (13) uses $\varepsilon = 10^{-12}$ and passes the semantic check when it is at most 1 and the radius sum is positive. For functional validation, we allocate a global budget of $B = 384$ sentinel examples as evenly as possible across mature modules and retain examples within each module using a deterministic rule controlled by a random seed.

Algorithm 1 SIFT: boundary-free online prompt continual learning.

Require: stream $\{(x_t, y_t)\}_{t=1}^{T}$; frozen backbone θ and source encoder g_θ; initial prompt $P_0 \in \mathbb{R}^{L_p \times d}$; capacities $(K_{\mathrm{m}}, K_{\mathrm{q}})$; counts $(K_{\mathrm{conf}}, K_{\mathrm{seed}}, K_{\mathrm{warm}})$; routing parameters $(\tau_{\min}, \tau_{\max}, \beta)$; sentinel budget B; numerical constant $\varepsilon > 0$.

$\mathcal{M} \leftarrow \emptyset$; $\mathcal{Q} \leftarrow \emptyset$
for $t = 1, \ldots, T$ **do**
  $h_t \leftarrow$ normalized source representation of Eq.(4); $d_t(c) \leftarrow$ cosine distance of Eq.(6) for all $c \in \mathcal{M} \cup \mathcal{Q}$
  if $\mathcal{M} = \emptyset$ **then**
    create mature seed module C_0 with $n_0 = 0$; $\mathcal{M} \leftarrow \{0\}$; $\mathrm{ResetDistanceStats}(0)$; $a_t \leftarrow 0$
  else
    $j_t^\star \leftarrow \arg\min_{j \in \mathcal{M}} d_t(j)$
    $\tau_j \leftarrow$ the adaptive threshold of Eq.(7), for $j \in \mathcal{M}$
    if $d_t(j_t^\star) \le \tau_{j_t^\star}$ **then** ▷ revisit mature module
      $a_t \leftarrow j_t^\star$
    else if $\mathcal{Q} \neq \emptyset$ and, with $q_t^\star$ the nearest provisional module of Eq.(8), $d_t(q_t^\star) \le \tau_{\min}$ **then** ▷ join provisional module
      $a_t \leftarrow q_t^\star$
    else ▷ create provisional module
      if $|\mathcal{Q}| = K_{\mathrm{q}}$, remove the least recently used provisional module
      create C_q; $j(q) \leftarrow j_t^\star$; $P_q \leftarrow P_{j_t^\star}$; $a_t \leftarrow q$
    end if
  end if
  $\mathcal{L}_t \leftarrow$ the answer-only loss of Eq.(9); $P_{a_t} \leftarrow \mathrm{IsoStep}(P_{a_t}, \nabla_{P_{a_t}} \mathcal{L}_t)$ ▷ isolated prompt update
  $(\mu_{a_t}, n_{a_t}) \leftarrow \begin{cases} (h_t, 1), & n_{a_t} = 0, \\ \big((n_{a_t}\mu_{a_t} + h_t)/\|n_{a_t}\mu_{a_t} + h_t\|_2,\ n_{a_t} + 1\big), & n_{a_t} > 0. \end{cases}$ ▷ centroid update
  if $a_t \in \mathcal{M}$ **then**
    if this is a mature revisit, apply $\mathrm{UpdateDistanceStats}(a_t, d_t(a_t))$
    $\mathcal{S}_{a_t} \leftarrow \mathrm{Retain}_B(\mathcal{S}_{a_t} \cup \{(x_t, y_t)\})$ ▷ retain mature record
  else
    $\mathcal{H}_{a_t} \leftarrow \mathcal{H}_{a_t} \cup \{h_t\}$; $\mathcal{R}_{a_t} \leftarrow \mathcal{R}_{a_t} \cup \{(x_t, y_t)\}$ ▷ record for resolution
  end if
  if $a_t \in \mathcal{Q}$ and $n_{a_t} \ge K_{\mathrm{conf}}$ **then** ▷ selected provisional has enough samples
    $q \leftarrow a_t$
    if $|\mathcal{M}| < K_{\mathrm{seed}}$ **then** ▷ seed phase: promote directly
      promote q with $\mathcal{S}_q \leftarrow \mathrm{Retain}_B(\mathcal{R}_q)$; $\mathrm{ResetDistanceStats}(q)$
    else
      with $j \leftarrow j(q)$: $r_j \leftarrow$ the mature radius of Eq.(10); $r_q \leftarrow$ the candidate radius of Eq.(11); $\widetilde{r}_z \leftarrow$ the shrunk radius of Eq.(12)
      $s \leftarrow$ the overlap score of Eq.(13); $G \leftarrow [\widetilde{r}_q + \widetilde{r}_j > 0 \text{ and } s \le 1]$; $V \leftarrow [\mathcal{S}_j \neq \emptyset \text{ and } L_{\mathrm{sentinel}}(P_q) \le L_{\mathrm{sentinel}}(P_j)]$, with L_{sentinel} as in Eq.(14)
      if $G \wedge V$ **then** ▷ fusion acceptance condition of Eq.(15)
        apply the fusion state transfer of Eq.(16); $\mathcal{S}_j \leftarrow \mathrm{Retain}_B(\mathcal{S}_j \cup \mathcal{R}_q)$
        for $h \in \mathcal{H}_q$ **do**
          $\mathrm{UpdateDistanceStats}(j, \delta)$ with δ from Eq.(6) ▷ distance to fused centroid
        end for
        $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{q\}$
      else if $|\mathcal{M}| < K_{\mathrm{m}}$ **then** ▷ promote as a new mature module
        promote q with $\mathcal{S}_q \leftarrow \mathrm{Retain}_B(\mathcal{R}_q)$; $\mathrm{ResetDistanceStats}(q)$
      else ▷ keep provisional and retry later
      end if
    end if
  end if
end for
Inference: $h \leftarrow$ the normalized source representation of Eq.(4); $j^\star \leftarrow$ the mature module nearest to h under the cosine distance of Eq.(6); $\hat{y} \sim p_\theta(\cdot \mid x, P_{j^\star})$

C Visualization of Continual Learning Performance

Figure 3 shows how performance on each dataset changes during sequential learning. On Llama-3.2-1B-Instruct, Prompt Tuning and strLoRA obtain MeetingBank scores of 35.4 and 41.7 imme-

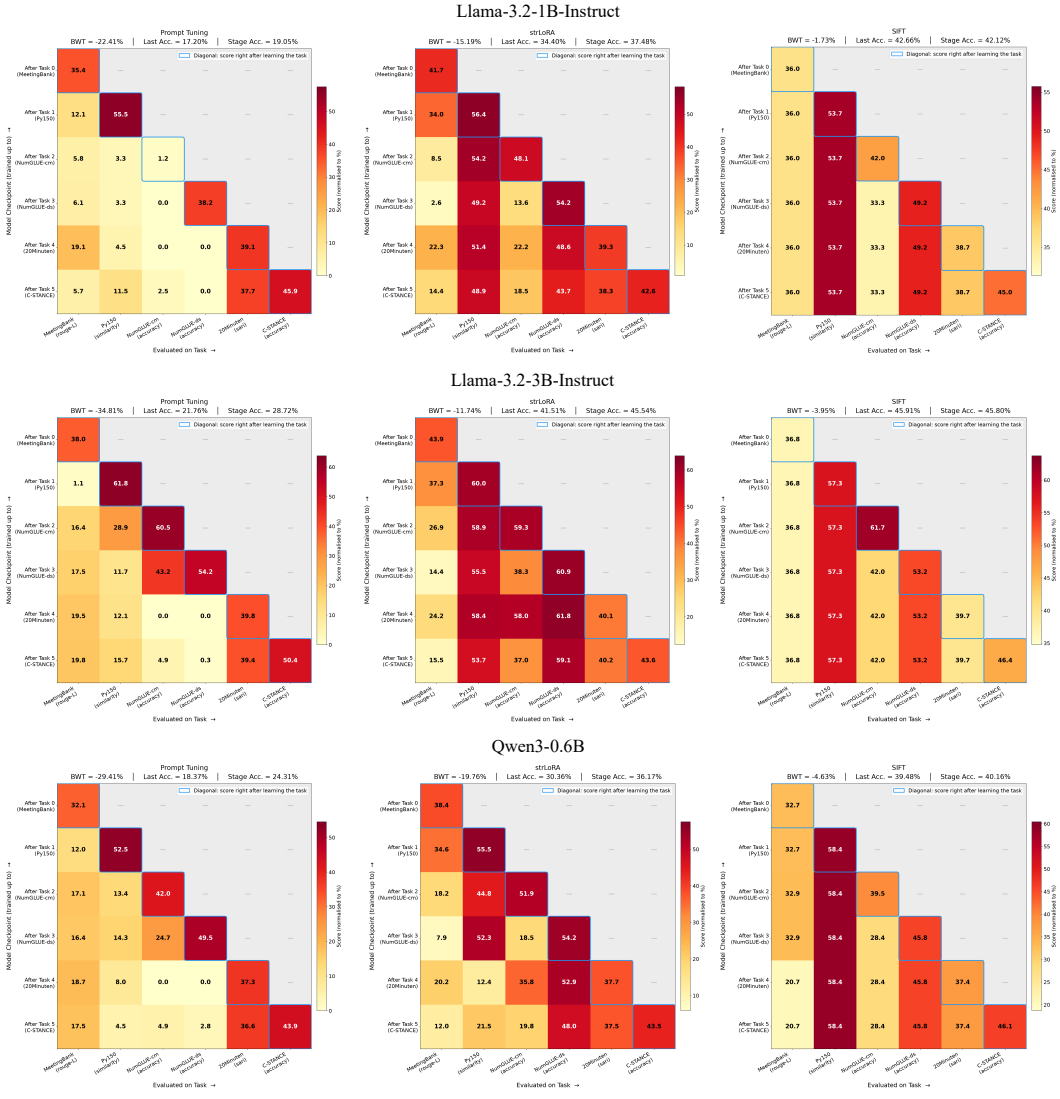


Figure 3: Performance trajectories of Prompt Tuning, strLoRA, and SIFT on three backbones. Within each heatmap, rows indicate successive training checkpoints and columns indicate evaluation datasets.

Table 4: Sensitivity to the confirmation count K_{conf} on Llama-3.2-1B-Instruct, with $K_{\text{seed}} = 3$ and $K_{\text{warm}} = 16$. The reference configuration is shown in bold.

K_{conf}	Stage Acc	Final Acc	BWT	MeetingBank	Py150	NumGLUE-cm	NumGLUE-ds	20Minuten	C-STANCE
1	29.43	21.10	$+0.00$	36.29	50.76	0.00	0.00	37.48	2.05
2	41.04	39.18	-4.69	36.72	53.90	20.99	48.92	38.89	35.65
4	**42.12**	**42.66**	-1.73	**36.01**	**53.74**	**33.33**	**49.23**	**38.68**	**45.00**
8	37.38	35.24	-2.72	34.94	55.70	24.69	13.54	39.07	43.50
16	38.25	37.15	-1.23	34.53	56.32	16.05	41.85	38.82	35.35

Table 5: Sensitivity to the seed phase count K_{seed} on Llama-3.2-1B-Instruct, with $K_{\text{conf}} = 4$ and $K_{\text{warm}} = 16$. The reference configuration is shown in bold.

K_{seed}	Stage Acc	Final Acc	BWT	MeetingBank	Py150	NumGLUE-cm	NumGLUE-ds	20Minuten	C-STANCE
1	39.98	39.69	-5.93	32.94	56.18	18.52	46.15	38.85	45.50
2	40.74	40.49	-1.98	34.32	55.96	25.93	49.85	38.88	38.00
3	**42.12**	**42.66**	-1.73	**36.01**	**53.74**	**33.33**	**49.23**	**38.68**	**45.00**
4	41.82	42.24	-1.48	34.86	55.76	32.10	49.23	38.83	42.65
8	33.72	31.38	$+1.48$	35.57	54.72	9.88	14.15	39.12	34.85

diately after learning the segment, but these fall to 5.7 and 14.4 by the final stage. In comparison, SIFT maintains 36.7 throughout the stream and preserves its Py150 score at 49.7 after learning that segment. Its NumGLUE-cm score further increases from 34.6 to 39.5 after the NumGLUE-ds segment and remains at that level, yielding positive backward transfer. On Llama-3.2-3B-Instruct and Qwen3-0.6B, SIFT likewise preserves Py150 scores at 57.3 and 58.4, while strLoRA declines from 60.0 to 53.7 and from 55.5 to 21.5, respectively. SIFT achieves higher final average scores than both baselines across all three backbones, reflecting stronger retention of previously learned knowledge as the stream progresses.

D Additional Ablation Studies

We further evaluate the sensitivity of SIFT's lifecycle controls on Llama-3.2-1B-Instruct. We vary one count in each table. The other two counts remain fixed at the reference configuration $(K_{\text{conf}}, K_{\text{seed}}, K_{\text{warm}}) = (4, 3, 16)$. The stream order, optimizer, prompt budget, evaluation checkpoints, and random seed remain fixed. These experiments use the same reference training configuration as the main results, varying only the hyperparameter under study.

The reference configuration gives the highest Final Acc in all three sweeps. Nearby settings also perform well. For K_{conf}, a value of 1 provides too little evidence before confirmation. Large values delay module consolidation. The best value in this sweep is $K_{\text{conf}} = 4$. For K_{seed}, values from 2 to 4 give similar results. Final Acc drops when $K_{\text{seed}} = 8$. This suggests that a long seed phase delays the use of the geometry and functional checks. The warmup results vary less than the other sweeps. Both short and long warmup periods perform below the reference. A moderate warmup helps collect reliable distance statistics. It also avoids delaying adaptation.

E Theoretical Analysis

We establish two results that motivate and clarify functional testing. The first shows that identical source representations can conceal incompatible target distributions. The second bounds the historical prediction loss change caused by one module fusion.

E.1 Semantic Similarity Does Not Guarantee Functional Compatibility

Proposition E.1 (Semantic similarity is insufficient)**.** *There exist data distributions A and B with identical source-representation distributions and a positive common semantic radius, and a probabilistic predictor whose adaptation strictly decreases expected one-token negative log likelihood on B while strictly increasing it on A.*

Table 6: Sensitivity to the warmup count K_{warm} on Llama-3.2-1B-Instruct, with $K_{\text{conf}} = 4$ and $K_{\text{seed}} = 3$. The reference configuration is shown in bold.

K_{warm}	Stage Acc	Final Acc	BWT	MeetingBank	Py150	NumGLUE-cm	NumGLUE-ds	20Minuten	C-STANCE
4	40.99	40.15	−2.22	33.94	55.79	30.86	46.77	38.79	34.75
8	40.62	39.72	−1.48	35.72	56.03	24.69	46.46	39.11	36.30
16	**42.12**	**42.66**	−1.73	**36.01**	**53.74**	**33.33**	**49.23**	**38.68**	**45.00**
32	40.31	40.53	−4.20	34.05	51.14	28.40	46.46	38.93	44.20
64	41.58	41.91	−1.98	35.60	54.38	30.86	47.08	39.19	44.35

Proof. Let A and B both assign probability $1/2$ to each of two inputs x_1, x_2. Choose unit representations $h(x_1) = e_1$ and $h(x_2) = e_2$, where e_1, e_2 are orthogonal. Both distributions therefore have the same normalized mean and root mean square cosine radius:

$$\mu = \frac{e_1 + e_2}{\sqrt{2}}, \qquad r = \sqrt{\mathbb{E}[(1 - \mu^\top h(x))^2]} = 1 - \frac{1}{\sqrt{2}} > 0. \tag{17}$$

Their center distance is zero, so source geometry cannot distinguish them.

Fix $0 < \pi < 1/2$. Under A, let $y \sim \text{Bernoulli}(\pi)$ independently of x; under B, let $y \sim \text{Bernoulli}(1 - \pi)$ independently of x. Consider a predictor whose adaptation parameter $p \in (0, 1)$ specifies $\Pr_p(y = 1 \mid x) = p$. Its expected one-token losses are

$$\begin{aligned} L_A(p) &= -\pi \log p - (1 - \pi) \log(1 - p), \\ L_B(p) &= -(1 - \pi) \log p - \pi \log(1 - p). \end{aligned} \tag{18}$$

Take the old parameter $p_A = \pi$ and the new parameter $p_B = 1 - \pi$. Direct substitution gives

$$L_A(p_B) - L_A(p_A) = L_B(p_A) - L_B(p_B) = (1 - 2\pi) \log \frac{1 - \pi}{\pi} > 0. \tag{19}$$

Thus the change improves B and harms A, despite their identical source geometry. All losses are finite because $p_A, p_B \in (0, 1)$. □

This counterexample rules out a universal implication from semantic similarity to safe parameter sharing without assumptions linking source representations to target distributions. It motivates checking target-dependent behavior, but does not assert that a particular frozen backbone realizes this predictor family or that SIFT resolves every such distribution shift.

E.2 A Conditional Risk Bound for One Fusion

Setup. Fix the learner state after a candidate q has been trained and immediately before it is fused into its parent j, as in Eq.(16). Superscripts $-$ and $+$ denote the states before and after this fusion. The mature index set $\mathcal{M}$ stays fixed, and q is excluded from inference in both states. Among mature prompts and centroids, only those of the parent change: $P_j^+ = P_q$, while $P_k^+ = P_k^-$ for $k \neq j$. The backbone and all other mature centroids remain fixed. Let $a^-(x)$ and $a^+(x)$ be the mature-only nearest-centroid inference routers, with the same deterministic rule for breaking ties.

Let D be any fixed historical distribution over $z = (x, y)$. Write $\ell(z, P)$ for the mean answer-token loss of the frozen backbone with prompt P, as in Eq.(9). Define

$$\begin{aligned} R^- &= \mathbb{E}_{z \sim D} \ell(z, P^-_{a^-(x)}), \quad R^+ = \mathbb{E}_{z \sim D} \ell(z, P^+_{a^+(x)}), \\ w &= \Pr_D(a^-(x) = j), \qquad \rho = \Pr_D(a^+(x) \neq a^-(x)). \end{aligned} \tag{20}$$

Here w is the historical probability mass served by the parent before fusion, and ρ is the probability that fusion changes the selected module.

For $w > 0$, let $D_j = D(\cdot \mid a^-(x) = j)$. Let $S_j = (z_1, \ldots, z_m)$, with $m \geq 1$, be the parent's sentinel records before candidate records are incorporated. The population and sentinel loss changes under prompt replacement are

$$\begin{aligned} \Delta &= \mathbb{E}_{z \sim D_j} \left[\ell(z, P_q) - \ell(z, P_j^-) \right], \\ \widehat{\Delta} &= \frac{1}{m} \sum_{i=1}^{m} \left[\ell(z_i, P_q) - \ell(z_i, P_j^-) \right]. \end{aligned} \tag{21}$$

The functional check in Eq.(15) is exactly $\widehat{\Delta} \leq 0$.

Assumptions. Assume $0 \leq \ell(z, P) \leq M$ for some finite $M > 0$, for every relevant prompt and example. For the answer-token negative log likelihood, this requires an additional condition; a uniform lower bound e^{-M} on the relevant target-token probabilities suffices. For $w > 0$, also assume a number $\eta \geq 0$ satisfies

$$\Delta \leq \widehat{\Delta} + \eta. \tag{22}$$

The quantity η allows the sentinel estimate to underestimate the historical loss increase. This is an explicit estimation assumption, not a claim that the retained records are an independent validation sample. No sampling or independence assumption is needed for the conditional bound below.

Proposition E.2 (Risk of a single fusion)**.** *Under the setup and assumptions above, if $w > 0$, then*

$$R^+ - R^- \leq w\widehat{\Delta} + w\eta + M\rho. \tag{23}$$

Consequently, a fusion passing the functional check has risk increase at most $w\eta + M\rho$. If $w = 0$, then $R^+ - R^- \leq M\rho$ without defining D_j or requiring the sentinel estimation assumption.

Proof. Introduce the intermediate risk

$$\widetilde{R} = \mathbb{E}_{z\sim D}\ell(z, P^+_{a^-(x)}),$$

which uses the new prompts with the old router. Only the parent prompt changes, so for $w > 0$,

$$\widetilde{R} - R^- = \mathbb{E}_{z\sim D}\left[\mathbf{1}\{a^-(x) = j\}\big(\ell(z, P_q) - \ell(z, P_j^-)\big)\right] = w\Delta. \tag{24}$$

The predictors defining R^+ and $\widetilde{R}$ agree whenever $a^+(x) = a^-(x)$. Otherwise, their losses differ by at most M. Hence

$$R^+ - \widetilde{R} \leq M \Pr_D(a^+(x) \neq a^-(x)) = M\rho. \tag{25}$$

Adding the two differences and using $\Delta \leq \widehat{\Delta} + \eta$ proves Eq.(23). The functional check gives $\widehat{\Delta} \leq 0$, yielding the stated consequence. When $w = 0$, the parent is selected with probability zero under the old router, so $\widetilde{R} = R^-$ and the routing bound alone proves the result. □