STREAMSELECT: SPARSE FINE-TUNING VIA ONLINE QUANTILE ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse fine-tuning efficiently adapts a pretrained large model to downstream tasks by updating a small number of existing weights. Choosing which weights to update, however, typically requires importance scores over the entire weight matrix, and a static choice can also become outdated as training proceeds. We introduce StreamSelect, which identifies and dynamically updates trainable weights online without full-matrix ranking. Specifically, StreamSelect derives importance scores for a candidate set sampled from the frozen weights and uses an online estimate of an upper-tail quantile to filter high-score candidates. We further design a replacement mechanism that maintains a fixed trainable budget by removing small, stabilized updates according to an online lower-tail quantile. The quantile-based selection and replacement then form a three-stage training framework. Theoretically, we characterize how score and quantile estimation errors affect the quality of selected weights, and bound the risk incurred by resetting a single update under local smoothness. Experiments on commonsense reasoning, mathematical reasoning, and code generation show that StreamSelect achieves the best average performance across all evaluated settings while updating only $< 1\%$ of model weights.

1 INTRODUCTION

Adapting a pretrained large model to downstream tasks becomes costly as the model scale and the number of tasks grow. Parameter-efficient fine-tuning (PEFT) instead freezes most parameters of the pretrained model and learns a compact, task-specific update, reducing the trainable parameters, optimizer state, and checkpoint storage. Existing methods fall into three families (Ding et al., 2023). Addition-based methods insert trainable modules or continuous prompts into the model (Houlsby et al., 2019; Lester et al., 2021). Sparse fine-tuning (FT) methods select a subset of existing weights to update (Guo et al., 2021). Reparameterization-based methods represent the update with fewer parameters, most notably through low-rank factorization (Hu et al., 2022).

Addition-based methods add inference-time cost, since their modules remain attached to the model. Sparse FT and reparameterization-based methods avoid this by merging the update directly into the pretrained weights. However, reparameterization-based methods constrain the update to a structured form, most notably the low-rank adaptation family (Hu et al., 2022). Such structural constraints limit expressivity. Sparse FT does not impose a low-rank constraint and can produce high-rank updates with a small trainable-parameter budget (Bhardwaj et al., 2024). Its expressivity depends instead on the choice of trainable weights, making weight selection the central design problem for this family.

Existing sparse FT methods typically select weights by importance-based scoring: gradient magnitude or Fisher information ranks individual weights (Xu et al., 2021; Nguyen et al., 2024; Bhardwaj et al., 2024), while block-level aggregates provide a coarser but cheaper score (He et al., 2025; Choudhary et al., 2025). Either way, the ranking requires scoring the full matrix and is fixed afterward. Dynamic sparse FT revises the trainable set during training. SpIEL (Ansell et al., 2024) uses full-matrix gradient information to identify new locations, but reduces the accumulation and storage cost by prescreening candidates before collecting their multi-step statistics. Consequently, locations excluded by the initial screening receive no further score accumulation within that selection round, potentially missing weights whose importance becomes clearer over multiple steps.

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.

This raises a natural question: can sparse FT select trainable weights both efficiently and dynamically, without full-matrix ranking? Our answer builds on two observations. First, sparse FT requires an informative trainable set, not the exact globally top-ranked weights. We can therefore estimate an upper-tail quantile from a small sample of weights and use it to filter candidates. Second, importance rankings change during training. In Fig. 1, the Top-10% Jaccard overlap between the initial ranking and the ranking after 100 steps falls to 0.116, despite using identical evaluation batches and dropout masks. Together, these observations suggest that sparse FT should seek informative weights rather than recover an exact global ranking, while revisiting the trainable set as importance evolves.

Based on these observations, we propose *StreamSelect*, which discovers and dynamically updates trainable weights without full-matrix ranking. At each selection round, StreamSelect samples a fixed-size candidate set from the currently unselected locations of each adapted matrix, computes their importance scores, and filters candidates using an online estimate of an upper-tail quantile. A replacement mechanism maintains a fixed trainable budget by replacing small, stabilized updates, identified through an online lower-tail quantile, with newly selected weights. StreamSelect integrates these mechanisms into three stages that construct the trainable set, refine it through dynamic replacement, and fix the resulting set for final optimization.

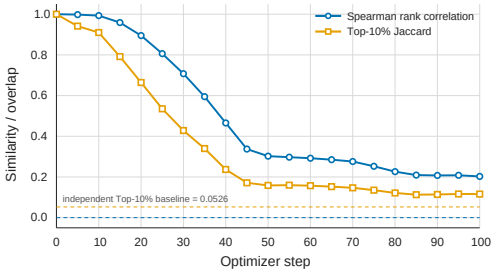

Figure 1: Change in gradient Signal-to-noise Ratio (SNR) rankings during fine-tuning. For each weight, $\mathrm{SNR} = |\mathbb{E}[g]|/(\mathrm{Std}(g))$ across five fixed calibration batches. Curves show Spearman correlation and Top-10% Jaccard overlap with step 0 for 100,000 frozen weights in one *q-projection* of LLaMA2-7B on Math10K.

Our contributions are threefold. 1) We introduce a sparse FT method that replaces full-matrix ranking with online quantile-based selection, and dynamically refines the trainable set through replacement. 2) We bound screening probabilities for sufficiently important and unimportant candidates under score and quantile estimation errors, as well as the risk increase from resetting a single update. 3) Experiments on commonsense reasoning, mathematical reasoning, and code generation show that StreamSelect achieves the best performance across all evaluated settings while updating $< 1\%$ of model weights.

2 Related Work

Sparse Fine-tuning. Sparse fine-tuning adapts pretrained models through sparse updates to existing weights. Diff Pruning (Guo et al., 2021) learns a sparse task-specific difference vector. FISH Mask (Sung et al., 2021) and LT-SFT (Ansell et al., 2022) determine trainable weights from Fisher information and a dense warm-up, respectively. PaFi (Liao et al., 2023) uses pretrained weight magnitudes without task-specific mask learning. Recent methods derive unstructured masks from early gradients or related salience (Song et al., 2024; Nguyen et al., 2024; Bhardwaj et al., 2024). Other methods incorporate pretrained weight scale or pruning-inspired salience (Choudhary et al., 2025; Kang et al., 2026; Ilin et al., 2026). Structured methods select larger parameter groups for more efficient execution. GIFT-SW (Zhelnin et al., 2025) updates salient columns, RoCoFT (Kowsher et al., 2025) trains rows and columns, and S^2FT (Yang et al., 2024) couples attention heads and channels. SMT (He et al., 2025) selects salient submatrices using gradient or activation statistics collected during a short warm-up. Methods that fix the trainable set after initial selection cannot revise that choice as importance changes during training. Task-aware selection also commonly requires full-model statistics or a dense warm-up.

Dynamic Sparse Fine-Tuning. Dynamic sparse fine-tuning revises the trainable set during optimization while preserving a fixed parameter budget. SpIEL (Ansell et al., 2024) alternates sparse optimization with replacement: it removes update coefficients that have largely stabilized and adds weights identified from accumulated gradient information. To reduce the cost of selection from full-matrix gradients, SpIEL first prescreens locations by gradient magnitude and then accumulates statistics only for the retained candidates. Our work targets this efficiency–dynamism tradeoff by replacing repeated full-matrix ranking with online quantile estimation-based selection.

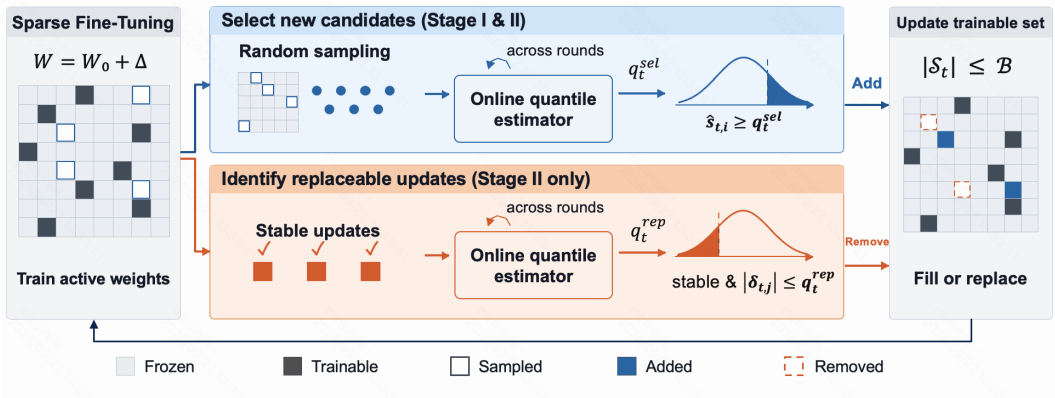

Figure 2: Overview of StreamSelect for one adapted matrix. In Stages I–II, sampled candidate scores update an upper-tail boundary for selecting new locations. In Stage II, historical first-stabilization magnitudes update a lower-tail boundary for removal.

Online Quantile Estimation. Exact quantiles over a growing stream generally require retaining all observations, motivating bounded-memory approximations. Reservoir sampling maintains a uniform sample from a stream of unknown length (Vitter, 1985), with extensions to weighted sampling (Efraimidis & Spirakis, 2006). Randomized order-statistic summaries offer explicit space–accuracy tradeoffs (Manku et al., 1999), while sketches such as t-digest emphasize tail accuracy (Dunning & Ertl, 2019). These methods estimate distributional thresholds without storing the full stream, but their use for dynamically selecting trainable weights remains unexplored in sparse FT.

3 Method

Consider a pretrained weight matrix $W_0 \in \mathbb{R}^{p\times q}$, whose weights are indexed by $\mathcal{I} = [p] \times [q]$, where $[n] = \{1, \ldots, n\}$. We identify each weight location with its index $i \in \mathcal{I}$. We learn a sparse update $\Delta \in \mathbb{R}^{p\times q}$ such that $W = W_0 + \Delta$. Let $\mathcal{S}_t \subseteq \mathcal{I}$ denote the index set of trainable weights after selection round t. At most $\mathcal{B} \leq pq$ coefficients are trainable at any time, so $|\mathcal{S}_t| \leq \mathcal{B}$ and $\Delta_i = 0$ for $i \notin \mathcal{S}_t$, where Δ_i is the update coefficient at location i.

3.1 From Full-Matrix Ranking to Quantile-Based Selection

We first consider how to select weights to add to the trainable set. Given an importance scoring rule, let $s_i \geq 0$ be the score of location $i \in \mathcal{I}$ at the pretrained state W_0, with larger scores indicating greater importance. Full-matrix ranking evaluates all pq scores and selects the $\mathcal{B}$ locations with the largest scores (Bhardwaj et al., 2024; Ansell et al., 2024). Define the empirical score distribution $F(x) = (pq)^{-1} \sum_{i\in\mathcal{I}} \mathbb{1}(s_i \leq x)$ and its quantile function $Q(u; F) = \inf\{x : F(x) \geq u\}$. For clarity, assume that the scores are distinct (ties can be resolved by a fixed ordering of the weights). The exact Top-$\mathcal{B}$ set can then be written as

$$i \in \mathrm{Top}_{\mathcal{B}}(s) \quad \Longleftrightarrow \quad s_i > Q(1 - \mathcal{B}/pq; F)\,. \tag{1}$$

Eq. (1) motivates an alternative criterion for selecting important locations based on a quantile boundary rather than full-matrix scoring and ranking. Exact Top-$\mathcal{B}$ recovery requires evaluating all pq scores, whereas deciding whether a location belongs to the high-score tail requires only a comparison with the upper-tail quantile once that boundary is known. This suggests estimating the boundary and comparing sampled scores against it to identify an approximate set of important locations without scoring and ranking the full matrix.

3.2 Online Quantile-Based Selection and Replacement

We apply the quantile-based criterion proposed in Section 3.1 to two decisions: which weights to add to the trainable set and which existing updates to remove. Selection uses importance scores to identify promising locations. Replacement uses the magnitudes of learned updates, together with a stability condition, to identify updates that can be discarded with a small change to the model parameters.

Upper-tail Quantile for Weight Selection. As training proceeds, importance scores change with the model parameters. We organize selection into *rounds*. In each round, we keep a candidate set fixed for $H \geq 1$ training steps to accumulate gradient statistics and estimate its importance scores before making a selection decision. At selection round t, let $s_{t,i} \geq 0$ be the importance score of location i under the current model. We search among the unselected locations $\mathcal{I}_t = \mathcal{I} \setminus \mathcal{S}_{t-1}$, while continuing to train the selected weights. Define the score distribution over these unselected locations as $F_t(x) = |\mathcal{I}_t|^{-1} \sum_{i \in \mathcal{I}_t} \mathbb{1}(s_{t,i} \leq x)$ and the desired selection boundary as $Q(1 - \rho_t^{\text{sel}}; F_t)$, where $\rho_t^{\text{sel}} \in (0, 1)$ is the target fraction of high-score locations we specify at round t. Here ρ_t^{sel} controls the selectivity of screening, whereas $\mathcal{B}$ limits the number of simultaneously trainable weights. Thus ρ_t^{sel} need not equal $\mathcal{B}/(pq)$, and each round need not recover the exact Top-$\mathcal{B}$ set.

Computing $Q(1 - \rho_t^{\text{sel}}; F_t)$ requires all $|\mathcal{I}_t|$ scores. Instead, we sample a candidate set $\mathcal{C}_t \subseteq \mathcal{I}_t$ of $m \ll pq$ locations. During the round, we train the weights in $\mathcal{S}_{t-1}$ and estimate each candidate's score $\hat{s}_{t,i}$ for $i \in \mathcal{C}_t$. Our main experiments use the gradient signal-to-noise ratio (SNR), $\hat{s}_{t,i} = |\hat{\mu}_{t,i}|/(\hat{\sigma}_{t,i} + \varepsilon)$, where $\hat{\mu}_{t,i}$ and $\hat{\sigma}_{t,i}$ are the empirical mean and standard deviation of the candidate's gradients, and $\varepsilon > 0$ ensures numerical stability. Appendix B.1 defines the alternative scores. To estimate $Q(1 - \rho_t^{\text{sel}}; F_t)$ efficiently, we compute the empirical quantile of candidate scores in each $\mathcal{C}_t$ (the current-round quantile) and smooth these quantiles with an exponential moving average (EMA) (Chen et al., 2000; Liu et al., 2026). To stabilize estimation across rounds with different score scales, we normalize each current-round quantile by the mean candidate score before averaging and restore the current scale afterward. The algorithm OnlineQuantileSel computes the selection boundary q_t^{sel} as

$$z_t = (1 - \beta) z_{t-1} + \beta \frac{Q(1 - \rho_t^{\text{sel}}; \{\hat{s}_{t,i} : i \in \mathcal{C}_t\})}{\varepsilon \vee m^{-1} \sum_{i \in \mathcal{C}_t} \hat{s}_{t,i}}, \qquad q_t^{\text{sel}} = m^{-1} z_t \sum_{i \in \mathcal{C}_t} \hat{s}_{t,i}. \tag{2}$$

Here Q applied to a finite collection denotes its empirical quantile, $a \vee b = \max\{a, b\}$, $\beta \in (0, 1]$ is the averaging rate, and $\varepsilon > 0$ prevents division by zero. At the first round of selection phase, z_t is initialized to the normalized current-round quantile. We select candidates satisfying $\hat{s}_{t,i} \geq q_t^{\text{sel}}$. Repeating this process with new candidate sets allows selection to continue as importance changes, without scoring all frozen locations. Appendix B.2 gives the implementation details.

Lower-tail Quantile for Weight Replacement. As importance changes during training, continued selection allows newly important locations to enter the trainable set. Once the budget is full, this requires freeing capacity by removing existing updates. We remove updates that have stabilized at small magnitudes: resetting them changes the model parameters only slightly, while checking stability avoids immediately removing newly selected weights whose updates start at zero. For $j \in \mathcal{S}_t$, let $\delta_{t,j}$ be its update coefficient and $\delta_{t-1,j}$ its value at the preceding selection round. Adjacent selection rounds are separated by H steps. Starting after H steps since selection, we measure its relative change as $c_{t,j} = |\delta_{t,j} - \delta_{t-1,j}|/(|\delta_{t,j}| + \varepsilon_c)$, where $\varepsilon_c > 0$ prevents division by zero. We consider the update stable when $c_{t,j} \leq \tau$ holds for P consecutive rounds, where $\tau > 0$ is the stability tolerance.

Let $\mathcal{E}_t$ contain the locations that first become stable at round t during their current stay in the trainable set. Unlike selection, replacement estimates a lower-tail quantile over all historical stabilization records to identify the least important stable updates, using magnitude as a proxy for importance. We therefore use reservoir sampling (Vitter, 1985; Efraimidis & Spirakis, 2006; Manku et al., 1999) to retain a small uniform sample of these records and compute its empirical quantile. Each matrix maintains a reservoir $\mathcal{V}_t$ of at most R magnitude–priority records, initialized empty at the start of Stage II. Each new record receives an independent random priority $u_{t,j} \sim \text{Unif}(0, 1)$, and we update

$$\mathcal{V}_t = \text{Keep}_R(\mathcal{V}_{t-1} \uplus \{(|\delta_{t,j}|, u_{t,j}) : j \in \mathcal{E}_t\}), \qquad q_t^{\text{rep}} = Q(\rho^{\text{rep}}; \{x : (x, u) \in \mathcal{V}_t\}). \tag{3}$$

Here Keep_R retains the records with the R largest priorities, or all records if fewer are available; $\uplus$ merges records, and repeated magnitudes are counted separately. Existing records keep their priorities and recorded magnitudes. The level $\rho^{\text{rep}} \in (0, 1)$ specifies the lower-tail quantile. We denote this reservoir update and boundary calculation by OnlineQuantileRep. After a warm-up period, a weight $j \in \mathcal{S}_t$ is ready for replacement only if it remains stable and its current magnitude satisfies $|\delta_{t,j}| \leq q_t^{\text{rep}}$. A smaller ρ^{rep} targets smaller magnitudes and makes replacement more conservative. Appendix B.2 gives the estimator implementation and warm-up requirements.

Algorithm 1 Three-stage StreamSelect training for a weight matrix

Initialize $\mathcal{S} \leftarrow \varnothing$ and the selection estimator
for selection rounds $t = 1, \ldots, T_{\mathrm{f}} - 1$ **do**
  On entering Stage II, reset selection for its new quantile level and initialize replacement
  Sample unselected locations $\mathcal{C}_t$; train $\mathcal{S}$ for H steps and accumulate candidate statistics
  Update $q_t^{\mathrm{sel}} \leftarrow \textsc{OnlineQuantileSel}(1 - \rho_t^{\mathrm{sel}}; \{\hat{s}_{t,i} : i \in \mathcal{C}_t\})$
  if $|\mathcal{S}| < \mathcal{B}$ (Stage I) **then**
    Add candidates satisfying $\hat{s}_{t,i} \geq q_t^{\mathrm{sel}}$ to $\mathcal{S}$, up to the remaining budget
  else {Stage II}
    Collect first-stabilization events $\mathcal{E}_t$; update $q_t^{\mathrm{rep}} \leftarrow \textsc{OnlineQuantileRep}(\rho^{\mathrm{rep}}; \{|\delta_{t,j}| : j \in \mathcal{E}_t\})$
    After warm up, replace stable weights satisfying $|\delta_{t,j}| \leq q_t^{\mathrm{rep}}$ with candidates satisfying $\hat{s}_{t,i} \geq q_t^{\mathrm{sel}}$
  end if
  Discard removed updates; initialize new coefficients and optimizer states to zero
end for
Stage III: fix $\mathcal{S}$ and train its coefficients for the remaining steps

3.3 Three-Stage Training Procedure

We combine the selection and replacement rules in three stages: initialize the trainable set, revise it within the budget, and optimize the final fixed set (Algorithm 1)

Stage I: trainable-set initialization. Starting from $\mathcal{S}_0 = \varnothing$, we add candidates using the upper-tail rule while training selected coefficients. A relatively large ρ_t^{sel} fills the budget efficiently. Once $|\mathcal{S}_t| = \mathcal{B}$, we enter Stage II, reset the selection estimator for the new quantile level, and begin stability tracking and lower-tail estimation.

Stage II: dynamic weight replacement. We continue selection with a smaller ρ_t^{sel}. After warm-up, candidates passing the upper-tail boundary replace stable updates satisfying $|\delta_{t,j}| \leq q_t^{\mathrm{rep}}$. We pair higher-scoring candidates with eligible weights in increasing order of $c_{t,j}$ until either group is exhausted. The magnitude threshold determines eligibility; among eligible updates, we prioritize those changing least, leaving more actively changing updates further opportunity to train. Each pair removes one location and adds one, preserving $|\mathcal{S}_t| = \mathcal{B}$. Removed weights return to their pretrained values; newly added coefficients and optimizer states start at zero in both Stages I and II.

Stage III: final optimization. At round $T_{\mathrm{f}} > 0$, we stop candidate scoring and replacement and train the fixed set for the remaining steps. Optionally, replacement is reduced before this cutoff by retaining a fraction $1 - (\min\{t/T_{\mathrm{f}}, 1\})^{\lambda}$ of valid pairs, where $\lambda > 0$ controls the decay.

4 Theoretical Guarantees

We analyze how reliably the quantile boundary accepts important candidates and rejects unimportant ones, and bound the immediate cost of replacement. For selection, we condition on the training history before round t, which fixes the underlying scores and F_t. Let G_t^{sel} be the CDF of the observed score $\hat{s}_{t,i}$ at a uniformly sampled location in $\mathcal{I}_t$. For replacement, let $x_1, \ldots, x_{n_t}$ be all first-stabilization magnitudes recorded up to round t, including those from locations since removed. For $n_t > 0$, define $G_t^{\mathrm{rep}}(x) = n_t^{-1} \sum_{k=1}^{n_t} \mathbb{1}(x_k \leq x)$.

Assumption 4.1 (Online Quantile Tracking)**.** At each round t, there are $\delta_t, \epsilon_t \geq 0$ and $\eta_t \in [0, 1/2)$ such that the selection bounds below hold jointly with probability at least $1 - \eta_t$, conditional on any fixed location $i \in \mathcal{I}_t$ being sampled. The replacement bound holds with probability at least $1 - \eta_t$:

$$Q(1 - \rho_t^{\mathrm{sel}} - \delta_t; G_t^{\mathrm{sel}}) - \epsilon_t \leq q_t^{\mathrm{sel}} \leq Q(1 - \rho_t^{\mathrm{sel}} + \delta_t; G_t^{\mathrm{sel}}) + \epsilon_t,$$
$$q_t^{\mathrm{rep}} \leq Q(\rho^{\mathrm{rep}} + \delta_t; G_t^{\mathrm{rep}}) + \epsilon_t. \tag{4}$$

The rank error δ_t shifts the quantile level, while the value error ϵ_t allows a numerical deviation from the corresponding quantile. We use common upper bounds for the two estimators. Appendices A.5 and A.6 give sufficient conditions for these bounds. For the normalized EMA, conditional sampling, bounded normalized scores, and local quantile regularity yield a candidate-conditioned error bound

that accounts for candidate sampling, empirical scale estimation, and the retained state; bounded distributional change controls its historical lag. For the reservoir, a fixed event history independent of the random priorities yields a quantile-level error bound that decreases with reservoir capacity.

Assumption 4.2 (Importance-Score Observations)**.** For fixed $\epsilon \geq 0$ and $\delta \in [0, 1)$, every $i \in \mathcal{I}_t$ satisfies $\mathbb{P}(|\hat{s}_{t,i} - s_{t,i}| > \epsilon \mid i \in \mathcal{C}_t) \leq \delta$.

Theorem 4.3. *Suppose Assumptions 4.1–4.2 hold and $\delta_t + \delta < \min\{\rho_t^{\text{sel}}, 1 - \rho_t^{\text{sel}}\}$. For any fixed $i \in \mathcal{I}_t$, condition on $i \in \mathcal{C}_t$: (i) if $s_{t,i} \geq Q(1 - \rho_t^{\text{sel}} + \delta_t + \delta; F_t) + \epsilon_t + 2\epsilon$, then $\mathbb{P}(\hat{s}_{t,i} \geq q_t^{\text{sel}}) \geq 1 - \eta_t - \delta$; (ii) if $s_{t,i} < Q(1 - \rho_t^{\text{sel}} - \delta_t - \delta; F_t) - \epsilon_t - 2\epsilon$, then $\mathbb{P}(\hat{s}_{t,i} \geq q_t^{\text{sel}}) \leq \eta_t + \delta$.*

The theorem shows that candidates sufficiently above the current importance quantile pass screening with high probability, while those sufficiently below it rarely pass. The combined errors shift the reference quantile level by $\delta_t + \delta$ and add a score margin of $\epsilon_t + 2\epsilon$. Larger errors therefore require a greater separation in true importance for reliable screening. As importance distribution changes during training, tracking lag contributes to the boundary error; details are deferred to Appendix A.5.

Replacement frees capacity for new candidates by removing stabilized updates with small magnitudes. The historical distribution G_t^{rep} sets the reference boundary, while current stability and magnitude determine eligibility for removal. We first bound the immediate replacement cost using the realized threshold, then relate this threshold to the historical lower-tail quantile. Let $\mathcal{R}(W) = \mathbb{E}_{Z \sim P_{\mathcal{D}}}[\mathcal{L}(W; Z)]$ denote the population risk, where $P_{\mathcal{D}}$ is the downstream data distribution, Z is a data example, and $\mathcal{L}(W; Z)$ is its loss. All model parameters outside the adapted matrix W are held fixed in this comparison.

Theorem 4.4 (Cost of Weight Replacement)**.** *Let $j \in \mathcal{S}_t$ satisfy $|\delta_{t,j}| \leq q_t^{\text{rep}}$, where $q_t^{\text{rep}} \geq 0$ is the realized replacement threshold. Let e_j be the matrix with entry one at location j and zero elsewhere, and $\nabla_j \mathcal{R}(W_t)$ be the corresponding partial derivative. If $\mathcal{R}$ is C-smooth on the coordinate box between W_0 and $W_t = W_0 + \Delta_t$, then resetting this update to zero gives*

$$\mathcal{R}(W_t - \delta_{t,j} e_j) - \mathcal{R}(W_t) \leq q_t^{\text{rep}} |\nabla_j \mathcal{R}(W_t)| + C(q_t^{\text{rep}})^2/2. \tag{5}$$

Corollary 4.5 (Quantile-Based Replacement Cost)**.** *Under the conditions of Theorem 4.4 and the replacement bound in Assumption 4.1, define $U_t = Q(\rho^{\text{rep}} + \delta_t; G_t^{\text{rep}}) + \epsilon_t$. With probability at least $1 - \eta_t$, $q_t^{\text{rep}} \leq U_t$, and hence*

$$\mathcal{R}(W_t - \delta_{t,j} e_j) - \mathcal{R}(W_t) \leq U_t |\nabla_j \mathcal{R}(W_t)| + C U_t^2/2. \tag{6}$$

Theorem 4.4 bounds the cost of removing a single update through the realized threshold, independently of how that threshold is estimated. For a fixed location, a smaller q_t^{rep} generally gives a tighter bound. Theorem 4.4 thus supports our choice to replace stabilized updates with small accumulated magnitudes. Corollary 4.5 relates this threshold to the historical first-stabilization distribution. The relevant magnitude is the accumulated change from the pretrained weight. Stability does not imply $\nabla_j \mathcal{R}(W_t) = 0$, so the bound retains the first-order term. These bounds concern immediate deletion before subsequent optimization.

5 Experiments

We evaluate StreamSelect on commonsense reasoning, mathematical reasoning, and code generation. We further examine the effects of candidate sampling and dynamic replacement, along with training speed and GPU memory use. Appendix B provides supplementary importance-score and replacement-intensity ablations, along with additional efficiency results.

Baseline Methods. We compare StreamSelect with low-rank methods LoRA (Hu et al., 2022) and DoRA (Liu et al., 2024), and sparse fine-tuning methods SHiRA-SNIP (Bhardwaj et al., 2024), SpIEL (Ansell et al., 2024), and SMT (He et al., 2025). Tables 1–3 report task performance and trainable parameter fractions. Results taken from prior work are marked in the table captions. Our mathematical reasoning and code generation runs use SpIEL-AG, labeled SpIEL in the tables.

Implementation Details. All main experiments use gradient SNR for candidate scoring. We train for three epochs on the reasoning tasks and one epoch on code generation. More details such as task-specific adaptation budgets, selection settings, and training and evaluation protocols are given in Appendix B.3.

Table 1: Accuracy on eight commonsense reasoning benchmarks. †, ‡, and ∗ denote values taken from Xu et al. (2026), He et al. (2025), and Bhardwaj et al. (2024), respectively; unmarked results are obtained with our implementation. Bold and underlined values indicate the best and second-best results within each model block, respectively.

Model	Method	Params.	BoolQ	PIQA	SIQA	HellaS	WinoG	ARC-e	ARC-c	OBQA	Avg.
LLaMA2-7B	LoRA† ($r = 32$)	0.83%	68.9	82.2	78.1	86.9	81.2	79.3	65.4	78.4	77.6
	DoRA‡ ($r = 16$)	0.42%	<u>72.0</u>	83.1	79.9	89.1	83.0	84.5	71.0	81.2	80.5
	SHiRA-SNIP*	1.00%	70.4	81.7	79.0	89.8	80.5	83.3	68.6	81.0	79.3
	SpIEL‡	0.83%	70.5	80.6	**80.8**	85.8	<u>83.4</u>	81.2	65.8	<u>81.8</u>	78.7
	SMT‡	0.84%	<u>72.0</u>	<u>83.8</u>	**80.8**	<u>93.3</u>	82.8	<u>86.7</u>	<u>74.0</u>	81.0	<u>81.8</u>
	StreamSelect	**0.43%**	**73.1**	**85.9**	<u>80.5</u>	**94.1**	**85.2**	**88.3**	**75.1**	**84.2**	**83.3**
LLaMA3-8B	LoRA† ($r = 32$)	0.70%	71.2	85.1	79.3	92.1	82.6	85.2	70.1	81.4	80.9
	DoRA‡ ($r = 32$)	0.71%	74.6	<u>89.3</u>	79.9	95.5	85.6	90.5	80.4	85.8	85.2
	SHiRA-SNIP	0.70%	74.7	88.5	80.2	95.7	85.0	91.0	79.7	86.0	85.1
	SpIEL‡	0.70%	72.1	83.6	80.0	91.8	85.4	91.2	76.8	80.8	82.7
	SMT‡	0.71%	**75.7**	88.4	<u>81.4</u>	<u>96.2</u>	<u>88.2</u>	<u>92.7</u>	<u>83.2</u>	**88.6**	<u>86.8</u>
	StreamSelect	**0.36%**	<u>75.4</u>	**89.4**	**82.1**	**96.6**	**89.3**	**93.9**	**84.0**	<u>87.8</u>	**87.3**

5.1 Commonsense Reasoning

We fine-tune LLaMA2-7B (Touvron et al., 2023) and LLaMA3-8B (Grattafiori et al., 2024) on Commonsense170K (Hu et al., 2023). We evaluate on eight benchmarks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), ARC-Easy and ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). We report accuracy on each task and their average (Avg.).

Table 1 shows that StreamSelect achieves the highest average accuracy on both models: 83.3% on LLaMA2-7B and 87.3% on LLaMA3-8B. It outperforms the second best results SMT by 1.5 and 0.5 percentage points, respectively, with about half as many trainable parameters. StreamSelect leads on seven of eight tasks with LLaMA2-7B and six with LLaMA3-8B, showing that it performs well across different commonsense tasks with a small trainable budget.

5.2 Mathematical Reasoning

We fine-tune LLaMA2-7B and LLaMA3-8B on Math10K, sampled from MetaMathQA (Yu et al., 2024), with 9,419 training and 500 validation examples. Following the four-task arithmetic-reasoning evaluation protocol used by Yao et al. (2026), we evaluate on the official test sets of GSM8K (Cobbe et al., 2021), AQuA (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), and SVAMP (Patel et al., 2021). We report both the unweighted average accuracy across the four benchmarks (Avg.) and the pooled accuracy over all 2,811 evaluation examples (Weighted Avg.). The latter additionally accounts for the substantial differences in test-set size across benchmarks.

Table 2 shows that StreamSelect achieves the highest Avg. and Weighted Avg. on both models. Its average accuracy reaches 56.85% on LLaMA2-7B and 73.91% on LLaMA3-8B, exceeding the second best result SpIEL by 1.47 and 1.78 percentage points, respectively. StreamSelect uses 28.80M trainable parameters compared with 56.10M for SpIEL. Section 5.4 further compares the methods at the same trainable budget.

5.3 Code Generation

We fine-tune LLaMA3-8B-Instruct on Magicoder-Evol-Instruct-110K (Wei et al., 2024). We evaluate on HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and their extended-test versions, HumanEval+ and MBPP+, using EvalPlus (Liu et al., 2023). We report Pass@1 on each benchmark and their average (Avg.).

Table 3 shows that StreamSelect achieves the highest average Pass@1 of 60.19%, exceeding the second best result SHiRA-SNIP by 0.85 percentage points with about one third as many trainable parameters. It ranks first on MBPP and MBPP+ and ties for first on HumanEval+. Together with

Table 2: Accuracy (%) on four mathematical reasoning benchmarks. † denotes results from Yang et al. (2024); unmarked results use our implementation. Bold and underline mark the best and second-best scores per model.

Model	Method	Trainable	GSM8K	AQuA	MAWPS	SVAMP	Avg.	Weighted Avg.
LLaMA2-7B	LoRA†	0.83%	44.00	20.90	85.70	59.20	52.45	50.85
	DoRA†	0.84%	43.80	24.40	**89.10**	59.10	54.10	51.33
	SHiRA-SNIP	0.83%	43.97	**30.71**	82.35	60.30	54.33	51.83
	SpIEL	0.83%	<u>45.64</u>	25.98	<u>89.08</u>	60.80	<u>55.38</u>	<u>52.93</u>
	SMT	0.83%	40.71	<u>30.31</u>	81.93	<u>61.90</u>	53.72	50.80
	StreamSelect	**0.43%**	**48.52**	28.35	88.24	**62.30**	**56.85**	**54.96**
LLaMA3-8B	LoRA†	0.70%	61.60	25.60	90.80	73.80	62.95	65.16
	DoRA†	0.71%	62.70	26.80	**91.20**	74.00	63.68	65.89
	SHiRA-SNIP	0.70%	66.57	44.88	87.82	78.30	69.39	70.58
	SpIEL	0.70%	<u>70.74</u>	47.64	90.34	**79.80**	<u>72.13</u>	<u>73.53</u>
	SMT	0.71%	62.32	<u>48.03</u>	84.87	76.70	67.98	68.05
	StreamSelect	**0.36%**	**72.40**	**53.15**	<u>91.18</u>	<u>78.90</u>	**73.91**	**74.56**

Table 3: Greedy Pass@1 (%) on code generation benchmarks. † denotes results from Jiang et al. (2025); unmarked results follow our evaluation protocol. Trainable fractions use the total model and adapter parameter count. Bold and underline mark the best and second-best scores.

Method	Trainable	HumanEval	HumanEval+	MBPP	MBPP+	Avg.
LoRA†	2.05%	48.80	44.50	<u>67.20</u>	<u>57.40</u>	54.48
DoRA†	2.06%	51.80	46.30	66.90	<u>57.40</u>	55.60
SHiRA-SNIP	2.05%	**60.98**	**55.49**	65.34	55.56	<u>59.34</u>
SpIEL	2.05%	<u>58.54</u>	53.05	65.61	54.76	57.99
SMT	2.05%	57.93	<u>53.66</u>	64.02	55.29	57.72
StreamSelect	**0.67%**	<u>58.54</u>	**55.49**	**67.99**	**58.73**	**60.19**

the reasoning results, these findings show that StreamSelect adapts effectively across different tasks while keeping fewer than 1% of model parameters trainable at any time.

5.4 Ablation Studies and Analysis

We study candidate sampling, dynamic replacement, and training efficiency on LLaMA2-7B with Math10K. StreamSelect retains the main configuration above. All comparisons use $\mathcal{B} = 180$k per matrix and a five-step selection interval.

We first vary the candidate count m over 10k, 20k, and 40k, adjusting the selection quantiles to keep the expected numbers of additions and proposals constant. Figure 3(a) shows that 10k candidates already yield accuracy close to SpIEL, while 40k achieves the highest Avg. and Weighted Avg. among these candidate counts.

To test whether full-matrix ranking improves selection, we also evaluate a full-matrix ranking baseline that retains our training procedure but accumulates five-step SNR statistics at every unselected location and selects the exact top 5k during fill and top 1k during replacement. These counts match the expected selections of StreamSelect-40k during stage I & II. Figure 3(a) shows that full-matrix ranking lowers Avg. by 2.15 points and Weighted Avg. by 2.42 points. This does not mean that larger SNR values indicate less important locations. Rather, SNR is estimated from only five gradient observations per location, and selecting the same number from a much larger population targets a more extreme upper tail of these estimates. For example, 1k selections correspond to the top 2.5% of a 40k candidate set but approximately the top 0.006% of a 4096×4096 matrix. Ranking this more extreme tail may favor estimates inflated by sampling noise, so a larger observed SNR need not reflect a larger underlying SNR.

For comparison with SpIEL, we match the per-matrix trainable budget and five-step selection interval, and use the same training data and evaluation protocol. We also replace only SpIEL's grow score,

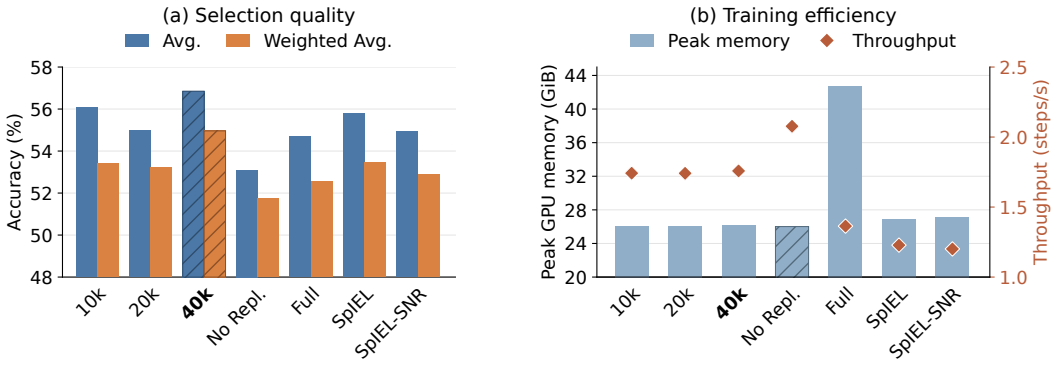

Figure 3: Ablations on LLaMA2-7B Math10K: (a) accuracy; (b) peak allocated memory and training throughput. Full denotes full-matrix ranking. No Repl. reports peak memory during the fixed phase; other peaks cover the full training run. The hatched bar represents best result.

from the absolute mean gradient to SNR, and denote this variant by SpIEL-SNR. StreamSelect exceeds their Avg. by 1.05 and 1.90 points, respectively (Figure 3(a)), indicating that its advantage is not explained by the choice of SNR alone. Both SpIEL variants retain their initial gradient-based prescreening and RigL replacement procedure, so this comparison evaluates the overall selection and replacement design.

We isolate continued selection by comparing the 40k run with a no-replacement variant (No Repl. in Figure 3(a)) that shares its entire fill trajectory but fixes the trainable set immediately afterward. Continuing replacement improves Avg. by 3.75 points and Weighted Avg. by 3.24 points (Figure 3(a)). Since both runs start from the same filled set, the gain supports revising trainable locations as training proceeds, consistent with the changing importance observed in Figure 1.

We measure training throughput including selection work and peak allocated GPU memory per GPU (StepPeak). The full-matrix ranking baseline requires 43,727 MiB, compared with 26,730 MiB for StreamSelect-40k, and reduces throughput from 1.758 to 1.364 steps/s (Figure 3(b)). Maintaining full-matrix gradient statistics therefore adds substantial memory cost without improving accuracy in this comparison, limiting its scalability. SpIEL instead prescreens locations using the first step's absolute gradients and accumulates selection statistics only for the retained candidates. This reduces persistent storage relative to full-matrix accumulation and explains its much lower memory peak. StreamSelect-40k still achieves 43.2% higher throughput and 781 MiB lower StepPeak than SpIEL with aligned cache and allocator settings.

6 Discussion and Limitations

Our focus is efficient selection within a fill-and-replacement framework. We compare seven importance scores but do not further study their task-dependent behavior. Understanding when and why different scores are effective could guide their choice across tasks. The current implementation already achieves efficiency gains, with further room to optimize gradient collection and sparse updates. More selective gradient computation and optimized sparse kernels could reduce runtime and memory use without changing the selection and replacement rules.

7 Conclusion

StreamSelect finds sparse update locations through sampled importance scores and online quantile boundaries. It continues to admit high-score candidates after the budget is full by retiring stable, small-magnitude updates. The theory relates boundary accuracy to admitted-score quality and immediate retirement cost, and experiments demonstrate competitive fine-tuning across reasoning and code generation. Candidate-set-size and identical-fill comparisons support the central result: useful sparse supports can be discovered and revised during training without full-matrix importance ranking.

REFERENCES

Alan Ansell, Edoardo Maria Ponti, Anna Korhonen, and Ivan Vulić. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 1778–1796, 2022.

Alan Ansell, Ivan Vulić, Hannah Sterz, Anna Korhonen, and Edoardo M. Ponti. Scaling sparse fine-tuning to large language models. *arXiv preprint arXiv:2401.16405*, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Kartikeya Bhardwaj, Nilesh Prasad Pandey, Sweta Priyadarshi, Viswanath Ganapathy, Shreya Kadambi, Rafael Esteves, Shubhankar Borse, Paul Whatmough, Risheek Garrepalli, Mart Van Baalen, Harris Teague, and Markus Nagel. Sparse high rank adapters. In *Advances in Neural Information Processing Systems*, volume 37, pp. 13685–13715, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

Fei Chen, Diane Lambert, and José C. Pinheiro. Incremental quantile estimation for massive tracking. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 516–522. ACM, 2000.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Anand Choudhary, Yasser Sulaiman, Lukas Mauch, Ghouthi Boukli Hacene, Fabien Cardinaux, and Antoine Bosselut. GaLLoP: Gradient-based sparse learning on low-magnitude parameters. *arXiv preprint arXiv:2510.19778*, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5:220–235, 2023.

Ted Dunning and Otmar Ertl. Computing extremely accurate quantiles using t-digests. *arXiv preprint arXiv:1902.04023*, 2019.

Pavlos S. Efraimidis and Paul G. Spirakis. Weighted random sampling with a reservoir. *Information Processing Letters*, 97(5):181–185, 2006.

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.

Demi Guo, Alexander M. Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 4884–4896, 2021.

Haoze He, Juncheng Li, Xuan Jiang, and Heather Miller. SMT: Fine-tuning large language models with sparse matrices. In *International Conference on Learning Representations*, pp. 32369–32393, 2025.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, 2019.

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

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5254–5276, 2023.

Ivan Ilin, Philip Zmushko, and Peter Richtarik. Super-tuning: From activation-aware pruning to sparse fine-tuning. *arXiv preprint arXiv:2607.09287*, 2026.

Shuyang Jiang, Yusheng Liao, Ya Zhang, Yanfeng Wang, and Yu Wang. Fine-tuning with reserved majority for noise reduction. In *International Conference on Learning Representations*, pp. 30495–30514, 2025.

Sungmin Kang, Jisoo Kim, Salman Avestimehr, and Sunwoo Lee. GEM: A scale-aware and distribution-sensitive sparse fine-tuning framework for effective downstream adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22509–22517, 2026.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1152–1157, 2016.

Md Kowsher, Tara Esmaeilbeig, Chun-Nam Yu, Chen Chen, Mojtaba Soltanalian, and Niloofar Yousefi. RoCoFT: Efficient finetuning of large language models with row-column updates. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 26659–26678, 2025.

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.

Baohao Liao, Yan Meng, and Christof Monz. Parameter-efficient fine-tuning without introducing new latency. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 4242–4260, 2023.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 158–167, 2017.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21558–21572, 2023.

Rui Liu, Peng Gao, Yu Shen, Ming Lin, and Pratap Tokekar. Adaptive conformal guidance for learning under uncertainty. In *International Conference on Learning Representations*, pp. 6168–6192, 2026.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32100–32121. PMLR, 2024.

Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 251–262, 1999.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018.

Bac Nguyen, Stefan Uhlich, Fabien Cardinaux, Lukas Mauch, Marzieh Edraki, and Aaron Courville. SAFT: Towards out-of-distribution generalization in fine-tuning. In *European Conference on Computer Vision*, pp. 138–154. Springer, 2024.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, 2021.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4463–4473, 2019.

Weixi Song, Zuchao Li, Lefei Zhang, Hai Zhao, and Bo Du. Sparse is enough in fine-tuning pre-trained large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46121–46135, 2024.

Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Jeffrey S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, 1985.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. Magicoder: Empowering code generation with OSS-instruct. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52632–52657, 2024.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9514–9528, 2021.

Yixing Xu, Guanchen Li, Chao Li, Xuanwu Yin, Dong Li, Spandan Tiwari, Ashish Sirasao, and Emad Barsoum. Low kruskal-rank adaptation. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, 2026.

Xinyu Yang, Jixuan Leng, Geyang Guo, Jiawei Zhao, Ryumei Nakada, Linjun Zhang, Huaxiu Yao, and Beidi Chen. S^{2}FT: Efficient, scalable and generalizable LLM fine-tuning by structured sparsity. In *Advances in Neural Information Processing Systems*, volume 37, pp. 59912–59947, 2024.

Kai Yao, Zhenghan Song, Kaixin Wu, Mingjie Zhong, Danzhao Cheng, Zhaorui Tan, Yixin Ji, and Penglei Gao. Gast: Gradient-aligned sparse tuning of large language models with data-layer selection. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4401–4416, 2026.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, pp. 45040–45061, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

Maxim Zhelnin, Viktor Moskvoretskii, Egor Shvetsov, Maria Krylova, Venediktov Egor, Zuev Aleksandr, and Evgeny Burnaev. GIFT-SW: Gaussian noise injected fine-tuning of salient weights for LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 6463–6480, 2025.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for language editing, consistency checks, and improving clarity. We did not use generative AI tools for any of the research tasks requiring mandatory disclosure under the ICLR policy. All AI-assisted edits were reviewed by the authors. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

A PROOFS AND ESTIMATOR ERROR BOUNDS

We first relate a candidate's true importance to its probability of passing an estimated boundary, then bound the immediate retirement perturbation. The replacement theorem is deterministic; its quantile-based corollary and the selection theorem use the estimator errors in (4). The later subsections derive sufficient bounds for those errors, distinguishing the current observed-score law for selection from the historical event distribution for replacement.

For a fixed-round argument, we omit the time index on scores, matrices, sets, and quantile boundaries: $F = F_t$, $G^{\mathrm{sel}} = G_t^{\mathrm{sel}}$, $G^{\mathrm{rep}} = G_t^{\mathrm{rep}}$, and $\rho^{\mathrm{sel}} = \rho_t^{\mathrm{sel}}$. The labels sel and rep retain their meanings from the main text. We keep the indices on $\epsilon_t, \delta_t, \eta_t$ to distinguish estimator errors from score-observation errors ϵ, δ. All quantile levels below are required to lie in $(0, 1)$.

A.1 PROBABILITY SPACE AND QUANTILE CONVENTIONS

Fix a round t and a realization of the training history before its candidate draw. This fixes the current matrix, the unselected set $\mathcal{I}_t$, and the underlying scores used to define F_t. Select a location $i \in \mathcal{I}_t$ without inspecting this round's candidate scores. The event $i \in \mathcal{C}_t$ has positive probability. For each $j \in \mathcal{I}_t$, let $\mathbb{P}_j$ denote probability conditional on this history and on $j \in \mathcal{C}_t$. In particular, the tested location is analyzed under $\mathbb{P}_i$. Other candidate locations, score observations, and any remaining estimator randomness are not fixed. In particular, the boundary and $\hat{s}_{t,i}$ may depend on the same observations.

The index j below ranges over all unselected locations, not just the realized candidates. The population observed-score CDF is the mixture $G^{\mathrm{sel}}(x) = |\mathcal{I}_t|^{-1} \sum_{j \in \mathcal{I}_t} \mathbb{P}_j(\hat{s}_j \le x)$. Thus conditioning on the tested location's inclusion does not replace the reference population by that location's score distribution.

For a CDF F and $v \in (0,1)$, let $a = Q(v;F) = \inf\{x : F(x) \geq v\}$. The limits of a CDF at $-\infty$ and $+\infty$ ensure that a is finite. Monotonicity and right continuity imply

$$x < a \implies F(x) < v, \qquad F(a) \geq v, \qquad F(a-) \leq v. \tag{7}$$

Indeed, $F(x) \geq v$ at a point $x < a$ would contradict the definition of the infimum. For every n, monotonicity gives $F(a + 1/n) \geq v$, and right continuity yields $F(a) \geq v$. Finally, $F(x) < v$ for all $x < a$, so taking the left limit gives $F(a-) \leq v$. These statements allow atoms and repeated scores. No density or strictly increasing CDF is required.

To compare observed and underlying importance, for every $j \in \mathcal{I}_t$ define $\mathcal{O}_j = \{|\hat{s}_j - s_j| \leq \epsilon\}$. Assumption 4.2 gives $\mathbb{P}_j(\mathcal{O}_j^c) \leq \delta$. On $\mathcal{O}_j$, the implication $s_j \leq x - \epsilon \Rightarrow \hat{s}_j \leq x$ holds. Conversely, $\hat{s}_j \leq x$ and $\mathcal{O}_j$ together imply $s_j \leq x + \epsilon$. Since s_j is fixed under the reference history,

$$\begin{aligned} \mathbb{P}_j(\hat{s}_j \leq x) &\geq \mathbb{1}(s_j \leq x - \epsilon) - \mathbb{P}_j(\mathcal{O}_j^c), \\ \mathbb{P}_j(\hat{s}_j \leq x) &\leq \mathbb{1}(s_j \leq x + \epsilon) + \mathbb{P}_j(\mathcal{O}_j^c). \end{aligned} \tag{8}$$

Average the two inequalities over the unselected locations. By the definitions of F and G^{sel},

$$F(x - \epsilon) - \delta \leq G^{\mathrm{sel}}(x) \leq F(x + \epsilon) + \delta \qquad \text{for every } x \in \mathbb{R}. \tag{9}$$

This step only uses the individual error probabilities. It does not require errors at different locations to be independent, unbiased, or independent of the true scores.

These CDF inequalities imply a quantile comparison. For $v - \delta > 0$ and $v + \delta < 1$, we claim

$$Q(v - \delta; F) - \epsilon \leq Q(v; G^{\mathrm{sel}}) \leq Q(v + \delta; F) + \epsilon. \tag{10}$$

For the lower inequality, fix $x < Q(v - \delta; F) - \epsilon$. Then $x + \epsilon < Q(v - \delta; F)$, so (7) gives $F(x + \epsilon) < v - \delta$. The upper CDF bound in (9) yields $G^{\mathrm{sel}}(x) < v$. No point strictly below $Q(v - \delta; F) - \epsilon$ can therefore reach CDF level v, proving the lower quantile bound.

For the upper inequality, set $x = Q(v + \delta; F) + \epsilon$. Then $F(x - \epsilon) \geq v + \delta$ by (7). The lower CDF bound in (9) gives $G^{\mathrm{sel}}(x) \geq v$. Since this x reaches level v, the first crossing cannot be larger. This proves the upper quantile bound. The lower comparison alone only requires $v - \delta > 0$; the upper comparison alone only requires $v + \delta < 1$.

A.2 Proof of Theorem 4.3

Proof. We continue under $\mathbb{P}_i$, with the tested location fixed before the candidate scores are observed. Write $u = 1 - \rho^{\mathrm{sel}}$. The theorem's level condition implies $u - \delta_t - \delta > 0$ and $u + \delta_t + \delta < 1$, so all quantiles in the following argument are well-defined.

Let $\mathcal{E}$ be the event that both selection inequalities in Assumption 4.1 hold. Its candidate-conditioned probability is $\mathbb{P}_i(\mathcal{E}) \geq 1 - \eta_t$. Apply the lower comparison in (10) at $v = u - \delta_t$, and the upper comparison at $v = u + \delta_t$. On $\mathcal{E}$,

$$\begin{aligned} q^{\mathrm{sel}} &\geq Q(u - \delta_t; G^{\mathrm{sel}}) - \epsilon_t \geq Q(u - \delta_t - \delta; F) - \epsilon_t - \epsilon, \\ q^{\mathrm{sel}} &\leq Q(u + \delta_t; G^{\mathrm{sel}}) + \epsilon_t \leq Q(u + \delta_t + \delta; F) + \epsilon_t + \epsilon. \end{aligned} \tag{11}$$

There is no additional failure probability in this conversion: (10) is a deterministic comparison between the two reference CDFs. The probability η_t is charged once for the common boundary event, not once for each side.

To account for the tested location's score error, let $\mathcal{O}_i = \{|\hat{s}_i - s_i| \leq \epsilon\}$. Assumption 4.2 gives $\mathbb{P}_i(\mathcal{O}_i) \geq 1 - \delta$. The union bound yields

$$\mathbb{P}_i(\mathcal{E} \cap \mathcal{O}_i) = 1 - \mathbb{P}_i(\mathcal{E}^c \cup \mathcal{O}_i^c) \geq 1 - \mathbb{P}_i(\mathcal{E}^c) - \mathbb{P}_i(\mathcal{O}_i^c) \geq 1 - \eta_t - \delta. \tag{12}$$

This argument is valid even if $\mathcal{E}$ and $\mathcal{O}_i$ are dependent. It therefore allows the tested score to contribute to the quantile estimate.

Suppose first that $s_i \geq Q(u + \delta_t + \delta; F) + \epsilon_t + 2\epsilon$. On $\mathcal{O}_i$, the score cannot underestimate s_i by more than ϵ. Combining this fact with the upper bound in (11), on $\mathcal{E} \cap \mathcal{O}_i$ we obtain

$$\hat{s}_i \geq s_i - \epsilon \geq Q(u + \delta_t + \delta; F) + \epsilon_t + \epsilon \geq q^{\mathrm{sel}}. \tag{13}$$

Thus $\mathcal{E} \cap \mathcal{O}_i \subseteq \{\hat{s}_i \geq q^{\mathrm{sel}}\}$, and (12) gives $\mathbb{P}_i(\hat{s}_i \geq q^{\mathrm{sel}}) \geq 1 - \eta_t - \delta$. Equality in the importance condition is allowed because the screening rule accepts equality at its boundary.

Suppose instead $s_i < Q(u - \delta_t - \delta; F) - \epsilon_t - 2\epsilon$. On $\mathcal{O}_i$, the score cannot overestimate s_i by more than ϵ. Together with the lower bound in (11), this gives, on $\mathcal{E} \cap \mathcal{O}_i$,

$$\hat{s}_i \leq s_i + \epsilon < Q(u - \delta_t - \delta; F) - \epsilon_t - \epsilon \leq q^{\mathrm{sel}}. \tag{14}$$

Hence passing the boundary requires $\mathcal{E}^c \cup \mathcal{O}_i^c$, giving $\mathbb{P}_i(\hat{s}_i \geq q^{\mathrm{sel}}) \leq \mathbb{P}_i(\mathcal{E}^c \cup \mathcal{O}_i^c) \leq \eta_t + \delta$. The strict importance inequality is essential here: if the comparisons became equalities throughout, the rule would accept the location. This completes both implications in Theorem 4.3. □

The proof identifies two sources of the 2ϵ margin: converting a quantile of observed scores into a quantile of underlying importance, and bounding the tested location's own score error. The rank shift δ arises from the CDF comparison, whereas the failure probability δ concerns the tested location. These are different uses of the same observation envelope, not a union bound over all candidate locations. Consequently, neither m nor an additional concentration parameter appears explicitly in the theorem. Candidate sample size still affects the estimator errors $\delta_t, \epsilon_t, \eta_t$.

The theorem fixes i before inspecting its observed score, then conditions on its inclusion in the candidate set. It does not condition on that score passing the boundary, and it does not assert simultaneous correctness for all candidates. The two importance conditions leave an intervening region where either screening outcome is possible under the error bounds. An upper bound on the selection boundary is needed to control missed important locations; a lower bound is needed to control acceptance of unimportant locations.

If $\delta_t = \epsilon_t = \epsilon = \delta = 0$, the two conditions become $s_i \geq Q(1 - \rho^{\mathrm{sel}}; F)$ and $s_i < Q(1 - \rho^{\mathrm{sel}}; F)$. On the estimator's success event, these are exactly the accepted and rejected cases. If the online boundary is exact as well, $\eta_t = 0$, so both decisions have probability one. If only the boundary is exact, the remaining importance margins and failure probability come solely from ϵ, δ.

For a quantitative score-space interpretation, suppose $v \mapsto Q(v; F)$ is locally κ-Lipschitz around $u = 1 - \rho^{\mathrm{sel}}$ over all queried levels. Then

$$\begin{aligned} Q(u + \delta_t + \delta; F) + \epsilon_t + 2\epsilon &\leq Q(u; F) + \kappa(\delta_t + \delta) + \epsilon_t + 2\epsilon, \\ Q(u - \delta_t - \delta; F) - \epsilon_t - 2\epsilon &\geq Q(u; F) - \kappa(\delta_t + \delta) - \epsilon_t - 2\epsilon. \end{aligned} \tag{15}$$

Thus a fixed score margin exceeding $\kappa(\delta_t + \delta) + \epsilon_t + 2\epsilon$ suffices on either side. This extra regularity is only used to express the transition region in score units; the theorem itself permits discrete score distributions and gaps in their quantiles. As estimator errors vanish, fixed observation errors remain. Convergence of the estimated quantile alone need not give correct classification at a tied boundary.

A.3 Proof of Theorem 4.4

Proof. Fix a round, a realization of the current matrix and replacement threshold, and a location j to be removed. Write $W = W_t$, $\delta_j = \delta_{t,j}$, and $q^{\mathrm{rep}} = q_t^{\mathrm{rep}}$. Removing the update at j changes its weight from $(W_0)_j + \delta_j$ to $(W_0)_j$, leaving every other weight unchanged. The coordinate box

$$\mathcal{K} = \{V : \min((W_0)_i, W_i) \leq V_i \leq \max((W_0)_i, W_i) \text{ for every } i \in \mathcal{I}\} \tag{16}$$

therefore contains $W - \theta\delta_j e_j$ for every $\theta \in [0, 1]$. Smoothness means that $\mathcal{R}$ is differentiable on a neighborhood of $\mathcal{K}$ and $\|\nabla\mathcal{R}(V) - \nabla\mathcal{R}(V')\|_2 \leq C\|V - V'\|_2$ for $V, V' \in \mathcal{K}$, with matrices vectorized for the norm.

The removal rule gives

$$0 \leq |\delta_j| \leq q^{\mathrm{rep}}. \tag{17}$$

This uses the current accumulated update, not its value at the first-stabilization event. It is a deterministic inequality, so no quantile-estimation success event is needed.

Since $\|e_j\|_2 = 1$, the fundamental theorem of calculus along the segment gives

$$\begin{aligned}\mathcal{R}(W - \delta_j e_j) - \mathcal{R}(W) &= -\delta_j \int_0^1 \nabla_j \mathcal{R}(W - \theta\delta_j e_j)\, d\theta \\ &= -\delta_j \nabla_j \mathcal{R}(W) + \delta_j \int_0^1 \{\nabla_j \mathcal{R}(W) - \nabla_j \mathcal{R}(W - \theta\delta_j e_j)\}\, d\theta \\ &\le -\delta_j \nabla_j \mathcal{R}(W) + \frac{C}{2}\delta_j^2. \end{aligned} \tag{18}$$

Indeed, the absolute difference between the two partial derivatives is at most the norm of the gradient difference, which is bounded by $C\theta|\delta_j|$. The integral term is consequently at most $\int_0^1 C\theta\delta_j^2\, d\theta = C\delta_j^2/2$. Finally, (17) yields

$$-\delta_j \nabla_j \mathcal{R}(W) \le |\delta_j|\, |\nabla_j \mathcal{R}(W)| \le q^{\mathrm{rep}} |\nabla_j \mathcal{R}(W)|, \qquad \delta_j^2 \le (q^{\mathrm{rep}})^2. \tag{19}$$

Substituting these inequalities into (18) proves (5). If $\delta_j = 0$, removing the update leaves the model unchanged and the actual risk difference is zero. □

Proof of Corollary 4.5. Write $U = U_t = Q(\rho^{\mathrm{rep}} + \delta_t; G^{\mathrm{rep}}) + \epsilon_t$. Assumption 4.1 gives $0 \le q^{\mathrm{rep}} \le U$ with probability at least $1 - \eta_t$. On this event, $q^{\mathrm{rep}}|\nabla_j \mathcal{R}(W)| \le U|\nabla_j \mathcal{R}(W)|$ and $(q^{\mathrm{rep}})^2 \le U^2$. Substituting into (5) proves (6). The threshold event is common to all eligible locations, so j may be chosen after observing the threshold and stability statistics without an additional probability factor. □

To isolate quantile uncertainty, write $U = U_t$, $q_* = Q(\rho^{\mathrm{rep}}; G^{\mathrm{rep}})$, and $d = Q(\rho^{\mathrm{rep}} + \delta_t; G^{\mathrm{rep}}) - q_* + \epsilon_t$. Then $U = q_* + d$ and $d \ge 0$. For the same location and current derivative, replacing q_* by U in the numerical envelope increases it by

$$d|\nabla_j \mathcal{R}(W)| + C q_* d + \frac{C}{2} d^2. \tag{20}$$

This follows by expanding (6) around q_*. It compares numerical envelopes, not the losses of two removal policies: a location eligible under the estimated threshold need not satisfy $|\delta_j| \le q_*$. A jump in the historical quantile function can make d large even when the level error is small, so the quantile is retained explicitly.

The historical event distribution need not equal the distribution of current trainable-update magnitudes, and the recorded magnitude of an update need not equal its current magnitude. The deterministic theorem uses only the current inequality (17) and smoothness. The corollary additionally uses the online boundary guarantee, with no separate independence or stationarity assumption on the event values.

If $\nabla_j \mathcal{R}(W) = 0$, the deterministic cost is at most $C(q^{\mathrm{rep}})^2/2$, and on the corollary's success event it is also at most $CU^2/2$. The quadratic dependence can be attained: take $\mathcal{R}(V) = C\|V - W\|_2^2/2$ and $|\delta_j| = q^{\mathrm{rep}}$. Then W is stationary, and resetting the update at this location increases risk by exactly $C(q^{\mathrm{rep}})^2/2$. If $q^{\mathrm{rep}} = U$, the quantile-based envelope is attained as well.

For a fixed location, finite current derivative, and curvature bound, the deterministic envelope tends to zero as $q^{\mathrm{rep}} \to 0$; the quantile-based envelope does so as $U \to 0$. Small changes over recent rounds do not imply a zero derivative, so the first-order term remains. The result concerns the immediate risk change from resetting one update, with all other parameters fixed and before subsequent optimization.

A.4 Stationary and drifting streams

We derive a two-sided quantile bound for independent observations with possibly different distributions, then explain its use in the candidate-conditioned selection theorem. This calculation concerns empirical quantiles of weighted observations. Appendix A.5 treats the normalized EMA of per-round empirical quantiles separately.

Fix the history before the current candidate draw. Write $p_t = \mathbb{P}(i \in \mathcal{C}_t \mid \text{history}) > 0$; for a uniform sample of m distinct locations, $p_t = m/|\mathcal{I}_t|$ for every i. If a boundary failure event $\mathcal{E}^c$ has probability

at most $p_t\eta_t$ under this history, then

$$\mathbb{P}(\mathcal{E}^c \mid \text{history}, i \in \mathcal{C}_t) = \frac{\mathbb{P}(\mathcal{E}^c \cap \{i \in \mathcal{C}_t\} \mid \text{history})}{p_t} \le \frac{p_t\eta_t}{p_t} = \eta_t. \tag{21}$$

The inclusion event need not be independent of the boundary. This conservative conversion is sufficient when the tested score is reused for estimation. An unconditional error bound at level η_t cannot, by itself, be substituted into a candidate-conditioned statement at that same level. The reference history must also be the same on both sides of this conversion.

Weighted empirical quantiles. Under the reference probability law, consider independent observations $X_1, \dots, X_N$ with fixed CDFs G_k^{sel}, and a fixed current reference CDF G_t^{sel}. Choose deterministic nonnegative weights with $\sum_k w_k = 1$, set $\widehat{G}(x) = \sum_k w_k \mathbb{1}(X_k \le x)$, and let $\widehat{Q}(u) = Q(u; \widehat{G})$. For $0 < \eta_t < 1/2$, define

$$D = \sum_{k=1}^{N} w_k \sup_x |G_k^{\mathrm{sel}}(x) - G_t^{\mathrm{sel}}(x)|, \qquad a = \sqrt{\frac{\log(2/(p_t\eta_t))}{2} \sum_{k=1}^{N} w_k^2}, \qquad \delta_t = D + a. \tag{22}$$

If $u \pm \delta_t \in (0,1)$, then, with probability at least $1 - p_t\eta_t$,

$$Q(u - \delta_t; G_t^{\mathrm{sel}}) \le \widehat{Q}(u) \le Q(u + \delta_t; G_t^{\mathrm{sel}}). \tag{23}$$

When this independent-observation model holds conditional on the reference history, (21) gives both selection bounds jointly with conditional probability at least $1 - \eta_t$. Thus $u = 1 - \rho_t^{\mathrm{sel}}$ and $q_t^{\mathrm{sel}} = \widehat{Q}(u)$ give Assumption 4.1 with $\epsilon_t = 0$.

To prove the envelope, set $l = Q(u - \delta_t; G_t^{\mathrm{sel}})$ and $v = Q(u + \delta_t; G_t^{\mathrm{sel}})$. By (7), $G_t^{\mathrm{sel}}(l-) \le u - \delta_t$ and $G_t^{\mathrm{sel}}(v) \ge u + \delta_t$. The Kolmogorov distance bounds left limits as well as CDF values, giving

$$\begin{aligned} \mathbb{E} \sum_k w_k \mathbb{1}(X_k < l) &= \sum_k w_k G_k^{\mathrm{sel}}(l-) \le u - \delta_t + D = u - a, \\ \mathbb{E} \sum_k w_k \mathbb{1}(X_k \le v) &= \sum_k w_k G_k^{\mathrm{sel}}(v) \ge u + \delta_t - D = u + a. \end{aligned} \tag{24}$$

Each summand lies in an interval of length w_k. Hoeffding's inequality therefore bounds each of the following failure probabilities by $\exp\{-2a^2/\sum_k w_k^2\} = p_t\eta_t/2$: the total weight strictly below l is at least u, or the weight at or below v is less than u. Outside their union, $\widehat{Q}(u) \ge l$ and $\widehat{Q}(u) \le v$. This proves the joint envelope. The left-limit comparison ensures validity with repeated scores. Only two fixed thresholds are tested; no uniform empirical-CDF bound is needed. Weights chosen after observing the same values would require a different concentration argument.

Stationary and changing distributions. If all observed-score CDFs equal G^{sel} and $w_k = 1/N$, then $D = 0$ and $\delta_t = \sqrt{\log(2/(p_t\eta_t))/(2N)}$. Write this radius as a_N. Substitution into Theorem 4.3 gives, conditional on the specified candidate being sampled,

$$\begin{aligned} s_i \ge Q(1 - \rho^{\mathrm{sel}} + a_N + \delta; F) + 2\epsilon &\implies \mathbb{P}_i(\hat{s}_i \ge q^{\mathrm{sel}}) \ge 1 - \eta_t - \delta, \\ s_i < Q(1 - \rho^{\mathrm{sel}} - a_N - \delta; F) - 2\epsilon &\implies \mathbb{P}_i(\hat{s}_i \ge q^{\mathrm{sel}}) \le \eta_t + \delta. \end{aligned} \tag{25}$$

The sampling radius vanishes if $\log(2/(p_t\eta_t))/N \to 0$. This includes fixed confidence and a candidate-inclusion probability bounded away from zero, and also allows confidence to increase sufficiently slowly with N. For exact observations and a quantile function continuous at the target level, both importance thresholds approach $Q(1 - \rho^{\mathrm{sel}}; F)$. If $\eta_t \to 0$ as well, every candidate with a fixed positive margin from that threshold is classified correctly with probability tending to one. With fixed ϵ, δ, the remaining thresholds are $Q(1 - \rho^{\mathrm{sel}} + \delta; F) + 2\epsilon$ and $Q(1 - \rho^{\mathrm{sel}} - \delta; F) - 2\epsilon$, at continuity points of the corresponding quantiles.

A cumulative empirical quantile of a fixed stationary stream has growing N. A fixed-size window does not gain effective samples just because it is refreshed. With general weights, the effective sample size is $1/\sum_k w_k^2$. An exactly known quantile is handled directly with $\delta_t = \epsilon_t = \eta_t = 0$, not by substituting zero into a finite-sample logarithm.

For a changing distribution, suppose observation k corresponds to optimizer step $h_k \le h_t$ and

$$\sup_x |G_k^{\mathrm{sel}}(x) - G_t^{\mathrm{sel}}(x)| \le L\frac{h_t - h_k}{T}, \tag{26}$$

where T is the total step count. Substitution into (22) gives

$$\delta_t \le \sqrt{\frac{\log(2/(p_t\eta_t))}{2}\sum_k w_k^2} + \frac{L}{T}\sum_k w_k(h_t - h_k). \tag{27}$$

The first term is sampling uncertainty; the second is the average age of the observations multiplied by the distribution's rate of change. For N equally weighted observations from at most w optimizer steps ago, the bound is $\sqrt{\log(2/(p_t\eta_t))/(2N)} + Lw/T$. Theorem 4.3 then gives acceptance and rejection probabilities relative to the current F_t, with wider importance margins when past observations become stale. Setting $L = 0$ recovers the stationary bound.

This regularity concerns CDFs, not quantile values. Individual locations may change importance even if the aggregate distribution changes slowly. A rapid distributional change increases the bound unless the estimator reduces its reliance on older observations; without controlling that change, a small tracking error cannot be guaranteed.

A sufficient observation model transferring drift control from true-score CDFs to observed-score CDFs is $\hat{s} = s + \xi$, where ξ is independent of s and has the same law at all steps. Indeed,

$$G_k^{\mathrm{sel}}(x) = \mathbb{E}_\xi F_k(x - \xi), \qquad \sup_x |G_k^{\mathrm{sel}}(x) - G_t^{\mathrm{sel}}(x)| \le \sup_x |F_k(x) - F_t(x)|. \tag{28}$$

If $\mathbb{P}(|\xi| > \epsilon) \le \delta$, this model also supplies the score-observation envelope. It is a sufficient model, not a requirement of the main theorem. The pointwise observation-error bound alone does not imply that the observed-score distribution is stationary whenever true importance is stationary.

The weighted calculation is an independent-stream model with fixed laws. To use its concentration bound after conditioning on a training history, these conditions must hold under that conditional law. Previously observed scores do not become independent fresh draws after the history containing them is fixed. Thus an arbitrary adaptive reuse of past scores requires its own conditional analysis. Assumption 4.1 admits such analyses without specifying the estimator. The next subsection supplies a conditional one-round bound for the normalized EMA and separately studies its accumulated lag. All screening probabilities above are pointwise in the round and candidate; they do not assert simultaneous correctness throughout training.

A.5 Normalized EMA and its score scale

The normalized EMA in (51) averages per-round empirical quantiles divided by empirical score means. It is not the quantile of an exponentially weighted empirical CDF. We give a separate sufficient envelope, including the randomness of the normalization.

Consider K selection rounds within a phase with a fixed target level $u = 1 - \rho^{\mathrm{sel}}$, at steps $h_0 < h_1 < \cdots < h_K$. Let G_t^{sel} be the conditional observed-score CDF at round t and $b_t^\circ = \int x\, dG_t^{\mathrm{sel}}(x) > 0$ its population mean. Define $Y_{t,i} = \hat{s}_{t,i}/b_t^\circ$, let H_t be its conditional CDF, and put $v_t = Q(u; H_t)$. Thus $\mathbb{E}[Y_{t,i} \mid \text{history}] = 1$ and $Q(u; G_t^{\mathrm{sel}}) = b_t^\circ v_t$.

Assume the m observations in each round are conditionally independent draws from G_t^{sel}, and $0 \le Y_{t,i} \le M$ almost surely for a common finite M. Assume also that $|Q(v; H_t) - Q(u; H_t)| \le \kappa|v - u|$ for the levels used below and that

$$\sup_x |H_t(x) - H_{t-1}(x)| \le L\frac{h_t - h_{t-1}}{T}. \tag{29}$$

This is an independent-draw model for the estimator analysis. Sampling distinct locations does not by itself imply independent score observations, since their gradients are computed on shared training batches. No independence between a current-round quantile and the mean of the same candidate scores is imposed. Bounds on raw scores together with a positive lower bound on their population mean imply the bounded normalized-score condition. Local inverse regularity is needed here because an EMA operates on numerical quantile values.

One-round estimation error. Fix the preceding history, including the previous normalized state z_{t-1}. Use the candidate-inclusion probability p_t from (21), and define

$$a_t = \sqrt{\frac{\log(4/(p_t\eta_t))}{2m}}, \qquad d_t = \min\{M, \sqrt{6M}\}\, a_t < 1, \qquad E_t = \frac{\kappa a_t + M d_t}{1 - d_t}. \tag{30}$$

Require $u \pm a_t$ to lie in the inverse-regularity neighborhood and $b_t^\circ(1 - d_t) \geq \varepsilon$, so the denominator floor is inactive on the event considered below. The two-threshold quantile argument gives a conditional failure probability at most $2e^{-2ma_t^2}$ for the current-round quantile interval. The mean bound derived below gives the same failure probability for $|\bar{Y}_t - 1| > d_t$, where $\bar{Y}_t = m^{-1}\sum_i Y_{t,i}$. Their union has probability at most $4e^{-2ma_t^2} = p_t\eta_t$. Equation (21) therefore controls both errors jointly with probability at least $1 - \eta_t$, conditional on the specified candidate's inclusion.

To obtain the mean bound, use both $0 \leq Y_{t,i} \leq M$ and its conditional mean of one, which implies $M \geq 1$. Convexity gives $e^{\lambda y/M} \leq 1 + (y/M)(e^\lambda - 1)$ for $0 \leq y \leq M$ and any real λ. Taking conditional expectations and using independence yields

$$\mathbb{E}\Big[e^{\lambda \sum_i Y_{t,i}/M} \mid \text{history}\Big] \leq \exp\Big\{\frac{m}{M}(e^\lambda - 1)\Big\}.$$

For $0 < r < 1$, exponential Markov bounds with $\lambda = \log(1 + r)$ and $\lambda = \log(1 - r)$ give upper- and lower-tail exponents $-m\{(1 + r)\log(1 + r) - r\}/M$ and $-m\{r + (1 - r)\log(1 - r)\}/M$, respectively. The expressions in braces are at least $r^2/3$ and $r^2/2$, so

$$\mathbb{P}(|\bar{Y}_t - 1| > r \mid \text{history}) \leq 2e^{-mr^2/(3M)}.$$

Hoeffding's inequality also gives $2e^{-2mr^2/M^2}$. If $M \leq 6$, applying the latter at $r = Ma_t$ suffices; if $M > 6$, applying the former at $r = \sqrt{6M}\, a_t$ suffices. Both give $2e^{-2ma_t^2}$. Thus the refined d_t is never larger than Ma_t and improves its dependence on M when $M > 6$, without an additional assumption.

On this event, the current-round quantile $\widetilde{v}_t = Q(u; \{Y_{t,i}\})$ satisfies $|\widetilde{v}_t - v_t| \leq \kappa a_t$. Since $0 \leq v_t \leq M$ and $\bar{Y}_t \geq 1 - d_t$,

$$\left|\frac{\widetilde{v}_t}{\bar{Y}_t} - v_t\right| \leq \frac{|\widetilde{v}_t - v_t| + v_t|1 - \bar{Y}_t|}{\bar{Y}_t} \leq E_t. \tag{31}$$

The previous state is fixed by the history. Expanding the current normalized update gives $|z_t - v_t| \leq (1 - \beta)|z_{t-1} - v_t| + \beta E_t$. For the returned boundary, it is sharper to cancel the current normalization before applying the triangle inequality:

$$\begin{aligned} q_t^{\mathrm{sel}} - Q(u; G_t^{\mathrm{sel}}) &= b_t^\circ\{(1 - \beta)\bar{Y}_t z_{t-1} + \beta\widetilde{v}_t - v_t\} \\ &= b_t^\circ\{(1 - \beta)\bar{Y}_t(z_{t-1} - v_t) + (1 - \beta)v_t(\bar{Y}_t - 1) + \beta(\widetilde{v}_t - v_t)\}. \end{aligned} \tag{32}$$

Here we used $q_t^{\mathrm{sel}} = b_t^\circ \bar{Y}_t z_t$ and the fact that the denominator floor is inactive on the event under consideration. Bounding each of the three terms yields

$$|q_t^{\mathrm{sel}} - Q(u; G_t^{\mathrm{sel}})| \leq b_t^\circ\{(1 - \beta)(1 + d_t)|z_{t-1} - v_t| + (1 - \beta)Md_t + \beta\kappa a_t\}. \tag{33}$$

Thus Assumption 4.1 holds for selection with $\delta_t = 0$ and ϵ_t equal to this right-hand side. It is fixed by the preceding history and the conditional score law. No assumption that the previous-state error is small is needed. When $\beta = 1$, the previous state and empirical-mean error cancel exactly, leaving the ordinary current-round quantile bound $b_t^\circ \kappa a_t$. At a phase reset the estimator also returns the current-round quantile, so the same bound applies.

For fixed u within a phase, the CDF drift condition and inverse regularity imply $|z_{t-1} - v_t| \leq |z_{t-1} - v_{t-1}| + \kappa L(h_t - h_{t-1})/T$. Consequently, distribution drift enters the current screening margins through the error in the retained state. The analysis below bounds how this error accumulates along a training path; it is separate from conditioning on the current candidate. The conditional independence used here is a sufficient sampling model for the estimator calculation, not an assumption needed by Theorem 4.3.

Tracking over successive rounds. Fix deterministic positive failure levels $\eta_1, \dots, \eta_K$, set $\eta_* = \min_{1\le t\le K} \eta_t$, and define

$$a = \sqrt{\frac{\log(4K/\eta_*)}{2m}}, \qquad d = \min\{M, \sqrt{6M}\}\, a < 1, \qquad E = \frac{\kappa a + Md}{1-d}. \tag{34}$$

Require that $u \pm a$ and the drift-shifted levels lie in the stated inverse-regularity neighborhood. We first analyze the empirical generalized inverse; the interpolation adjustment used in numerical implementations is given below. Also require $b_t^\circ(1-d) \ge \varepsilon$, where ε is the positive denominator floor in (51).

The two-threshold argument of Appendix A.4, conditional on the preceding history, gives

$$\mathbb{P}(Q(u-a; H_t) \le Q(u; \{Y_{t,i}\}) \le Q(u+a; H_t) \,|\, \text{history}) \ge 1 - 2e^{-2ma^2}. \tag{35}$$

The sample mean $\bar{Y}_t = m^{-1}\sum_i Y_{t,i}$ satisfies $\mathbb{P}(|\bar{Y}_t - 1| > d \mid \text{history}) \le 2e^{-2ma^2}$ by the same mean bound. Applying a union bound over the two errors and all K rounds gives a joint event of probability at least $1-\eta_*$. Conditional estimates can be integrated over each adaptive preceding history, so independence across rounds is not required for this step.

On the joint event, $|\widetilde{v}_t - v_t| \le \kappa a$, where $\widetilde{v}_t = Q(u; \{Y_{t,i}\})$, and $1-d \le \bar{Y}_t \le 1+d$. The denominator floor is inactive. The actual input to the EMA is $\widetilde{v}_t/\bar{Y}_t$, so

$$\left|\frac{\widetilde{v}_t}{\bar{Y}_t} - v_t\right| \le \frac{|\widetilde{v}_t - v_t| + v_t|1-\bar{Y}_t|}{\bar{Y}_t} \le \frac{\kappa a + Md}{1-d} = E. \tag{36}$$

We used $0 \le v_t \le M$. The estimate controls the ratio jointly, rather than treating the empirical scale as fixed.

To convert distributional change into movement of the normalized target, set $r_t = L(h_t - h_{t-1})/T$. Equation (29) and the generalized-inverse argument give

$$Q(u - r_t; H_{t-1}) \le v_t \le Q(u + r_t; H_{t-1}), \qquad |v_t - v_{t-1}| \le \kappa r_t. \tag{37}$$

The last inequality uses inverse regularity of H_{t-1} at the shifted levels. This identifies where that regularity is needed; a CDF-distance bound alone does not control movement across a jump in the quantile function. Writing the recursion as $z_t = (1-\beta)z_{t-1} + \beta\widetilde{v}_t/\bar{Y}_t$ and adding and subtracting $(1-\beta)v_{t-1} + \beta v_t$ therefore yields

$$|z_t - v_t| \le (1-\beta)|z_{t-1} - v_{t-1}| + \beta E + (1-\beta)\frac{\kappa L}{T}(h_t - h_{t-1}). \tag{38}$$

Successively substitute the same inequality for $t-1, t-2, \dots, 1$. The initial error receives weight $(1-\beta)^t$, the repeated batch error receives total weight $1-(1-\beta)^t$, and the drift increments retain their individual ages. Hence

$$|z_t - v_t| \le (1-\beta)^t|z_0 - v_0| + \{1-(1-\beta)^t\}E + \frac{\kappa L}{T}\sum_{k=1}^{t}(1-\beta)^{t-k+1}(h_k - h_{k-1}) =: B_t. \tag{39}$$

Let $B_0 = |z_0 - v_0|$. Since $|z_{t-1} - v_t| \le B_{t-1} + \kappa r_t$ on the joint event, the exact identity (32) gives the raw-score bound

$$|q_t^{\text{sel}} - Q(u; G_t^{\text{sel}})| \le b_t^\circ\{(1-\beta)(1+d)(B_{t-1} + \kappa r_t) + (1-\beta)Md + \beta\kappa a\}. \tag{40}$$

As in the one-round bound, setting $\beta = 1$ leaves only $b_t^\circ \kappa a$ because the current empirical scale cancels. This joint pathwise event bounds the accumulated state error. It is not automatically a conditional guarantee after an arbitrary realized history is fixed. The candidate-conditioned guarantee is (33); the recursion above describes the size of its previous-state term along paths with accurate batch estimates. These error radii are analysis quantities, not inputs required by the algorithm.

After a phase reset, z_1 is the normalized quantile from the first round, so $|z_1 - v_1| \le E$ and the raw boundary error is at most $b_1^\circ \kappa a$. For $t \ge 2$, expanding the recursion from that state gives

$$|z_t - v_t| \le (1-\beta)^{t-1}|z_1 - v_1| + \{1-(1-\beta)^{t-1}\}E + \frac{\kappa L}{T}\sum_{k=2}^{t}(1-\beta)^{t-k+1}(h_k - h_{k-1}). \tag{41}$$

Both the initialization exponent and the accumulated batch-error coefficient change when the recursion starts at round 1. The target level is held fixed within each phase, and this reset starts a new analysis when the selection quantile changes between fill and grow.

If the normalized law is stationary, $L = 0$, the initialization contribution decays geometrically and B_t approaches the batch envelope E. Fixed m and fixed β retain finite-window uncertainty. If m grows with $\log(K/\eta_*)/m \to 0$, with bounded M, κ and bounded population scales, then a, d, E vanish; after initialization has decayed, the raw interface error vanishes as well.

If $h_t - h_{t-1} \le H$, the drift contribution satisfies

$$\frac{\kappa L}{T}\sum_{k=1}^{t}(1-\beta)^{t-k+1}(h_k - h_{k-1}) \le \frac{\kappa L H}{T}\frac{1-\beta}{\beta}. \tag{42}$$

This bounds the accumulated lag entering the conditional error radius in (33), and hence the importance margins in Theorem 4.3. At $\beta = 1$, this historical lag is exactly zero; the boundary uses the current-round quantile. At $L = 0$, the stationary envelope is recovered.

The distinction between scale and shape has an exact ideal case. If $\hat{s}_{t,i} = c_t Y_{t,i}$ for positive scales c_t and one common nonnegative law of Y with mean one, then H_t is constant even when c_t changes sharply. With exact per-round score means and quantiles, an inactive denominator floor, and a correctly initialized normalized state, $q_t^{\mathrm{sel}} = Q(u; G_t^{\mathrm{sel}})$ at every round. This property explains the normalization without treating a random empirical mean as a deterministic rescaling.

The effect of averaging on variance can be evaluated directly for independent, identically distributed candidate-score samples across rounds. Let Z_t denote the normalized current-round quantile actually fed to the EMA, including its denominator floor. Write $\bar{z} = \mathbb{E}Z_t$ and $V = \mathrm{Var}(Z_t) < \infty$. For deterministic initialization,

$$\begin{aligned}\mathbb{E}z_t - v &= (1-\beta)^t(z_0 - v) + \{1-(1-\beta)^t\}(\bar{z} - v),\\ \mathrm{Var}(z_t) &= \beta^2\sum_{k=1}^{t}(1-\beta)^{2(t-k)}V = \frac{\beta}{2-\beta}\{1-(1-\beta)^{2t}\}V.\end{aligned} \tag{43}$$

The equality follows by expanding the recursion and using independence of the Z_t. If z_0 is random and independent of the subsequent rounds' candidate score samples, the variance also contains $(1-\beta)^{2t}\,\mathrm{Var}(z_0)$, accounting for initialization from a separate initial candidate-score sample. The stationary variance approaches $\beta V/(2-\beta)$, so a larger β retains more variance in the normalized state, while (42) shows that it reduces historical lag. A finite-sample quantile or a quantile-to-mean ratio need not be unbiased, so $\bar{z} - v$ is kept explicitly. The variance formula concerns z_t; multiplying by the current empirical scale creates dependence, which is controlled by (40) rather than by assuming independent factors.

A.6 Historical retirement reservoirs

Corollary 4.5 uses a one-sided boundary guarantee against the entire historical event CDF. We derive a sufficient level error for a uniform priority reservoir on a fixed event sequence $x_1, \ldots, x_n$. The values may repeat and need not arise from identically distributed events.

Let R be the reservoir capacity. Assign independent continuous random priorities to the events, independently of their values, and keep the $r = \min\{R, n\}$ largest priorities. By exchangeability of the priorities, every subset of r event indices has the same probability; the retained values are a uniform sample without replacement from the fixed history. An online update can keep the same subset by retaining the largest priorities among its old reservoir and each new set of stabilization records.

If $n \le R$, the reservoir contains all event values, so its empirical generalized-inverse quantile equals $Q(\rho^{\mathrm{rep}}; G^{\mathrm{rep}})$ exactly. For $n > R$ and $0 < \eta_t < 1/2$, let

$$a_R = \sqrt{\frac{\log(1/\eta_t)}{2R}}, \qquad v = Q(\rho^{\mathrm{rep}} + a_R; G^{\mathrm{rep}}), \tag{44}$$

with $\rho^{\mathrm{rep}} + a_R < 1$. The population fraction of event values at or below v is $G^{\mathrm{rep}}(v) \ge \rho^{\mathrm{rep}} + a_R$. The fraction in a uniform sample without replacement falls below ρ^{rep} with probability at most

$e^{-2Ra_R^2} = \eta_t$, by the finite-population form of Hoeffding's inequality (Hoeffding, 1963). Outside that event, the reservoir quantile is at most v. Thus the replacement bound in Assumption 4.1 holds with $\delta_t = a_R$ and $\epsilon_t = 0$.

Substitution into Corollary 4.5 gives the explicit envelope

$$U = Q\left(\rho^{\mathrm{rep}} + \sqrt{\frac{\log(1/\eta_t)}{2R}}; G^{\mathrm{rep}}\right) \tag{45}$$

for a filled reservoir. The empty-reservoir case is excluded by the retirement warm-up. Increasing R reduces the uncertainty in which historical lower tail is used, and hence in the deletion-cost envelope. Once full, reservoir precision is controlled by R. A two-sided quantile bound follows by the same argument at the lower quantile, with $\log(2/\eta_t)$ in place of $\log(1/\eta_t)$.

The uniform-sample calculation conditions on event values that are independent of the reservoir priorities. When a reservoir influences retirement and thereby future event values, conditioning on the realized history does not automatically preserve uniformity. Theorem 4.4 remains a deterministic consequence of the removal rule and smoothness, while Corollary 4.5 applies when the replacement boundary guarantee holds; the fixed-history calculation is a sufficient sampling model, not a proof for every feedback-dependent stream. The same distinction applies when pseudorandom priorities are modeled as independent random priorities. Throughout, G^{rep} denotes the empirical CDF of the complete event history.

A.7 Numerical quantiles and ties

The proofs above use the empirical generalized inverse. To cover linear interpolation, order a sample of size r as $x_{(1)} \le \cdots \le x_{(r)}$ and let $\widetilde{Q}_r(u)$ interpolate at position $1 + (r-1)u$. This position is $ru + (1-u)$, between ru and $ru + 1$. Comparison with the order-statistic indices at levels $u \pm 1/r$ gives

$$Q(u - 1/r; \widehat{G}) \le \widetilde{Q}_r(u) \le Q(u + 1/r; \widehat{G}), \tag{46}$$

whenever those levels lie in $(0, 1)$. The upper comparison alone only needs $u + 1/r < 1$. Repeated values preserve these inequalities.

For a filled retirement reservoir, apply its one-sided concentration argument at the shifted empirical level $\rho^{\mathrm{rep}} + 1/R$. It gives, with the same failure probability η_t,

$$q^{\mathrm{rep}} \le Q(\rho^{\mathrm{rep}} + 1/R + a_R; G^{\mathrm{rep}}), \tag{47}$$

provided $\rho^{\mathrm{rep}} + 1/R + a_R < 1$. Thus interpolation adds $1/R$ to the level error in (45). If all n events fit in the reservoir, there is no sampling error, and the same comparison uses $1/n$ alone. When a shifted level would leave $(0, 1)$, interpolation can instead be bounded in value: its distance from the generalized inverse is at most the largest adjacent order-statistic gap. A bound on the range of the complete fixed event history also bounds this gap and can be included in ϵ_t without an additional probability event. For a single retained value the gap is zero, and both conventions agree.

For the normalized EMA, only a lower-tail deviation at empirical level $u - 1/m$ and an upper-tail deviation at level $u + 1/m$ are needed. Applying the two-threshold proof at these respective levels gives

$$Q(u - 1/m - a; H_t) \le \widetilde{Q}_m(u) \le Q(u + 1/m + a; H_t) \tag{48}$$

with failure probability at most $2e^{-2ma^2}$. There are still two one-sided deviations, not two full two-sided bounds. The mean estimate therefore retains its original probability budget, and all the previous confidence levels remain valid.

Provided the shifted levels are in the inverse-regularity neighborhood, replace κa by $\kappa(a + 1/m)$ in the batch-quantile, normalization, and raw-boundary bounds. For the candidate-conditioned calculation, make the corresponding replacement $\kappa a_t \mapsto \kappa(a_t + 1/m)$. The mean-error radii d, d_t are unchanged. In particular, at $\beta = 1$ the raw error bound is precisely that of the interpolated current-round quantile. All screening statements retain the rule $\hat{s}_i \ge q^{\mathrm{sel}}$; they do not impose an exact Top-$\mathcal{B}$ tie-breaking rule.

B Implementation and Experimental Details

B.1 Importance Score Definitions

Fix a round and keep its candidate set unchanged for H steps. At step h, let $\mathcal{D}_h$ be the training minibatch and $g_{h,i} = \nabla_i \mathcal{L}(\mathcal{D}_h, W_h)$ its mean loss gradient at candidate i. For the instantaneous population reference, let g_i be the mean loss gradient on a fresh training minibatch at the model state immediately before round t, and let $\mathbb{E}_t$ condition on the preceding training history. Define

$$(\mu_i, a_i, v_i) = \mathbb{E}_t(g_i, |g_i|, g_i^2), \qquad \sigma_i = (v_i - \mu_i^2)^{1/2}. \tag{49}$$

These moments define the instantaneous importance $s_{t,i}$ for the chosen scoring rule. We estimate them from the H steps in the round, during which the model may change. Both sampling error and within-round parameter change contribute to the observation error in Assumption 4.2. With a numerical constant $\varepsilon = 10^{-8}$ and $d_i = |(W_0)_i| + \varepsilon$, the seven supported scores are

$$(s_i^{(1)}, \dots, s_i^{(7)}) = \left(a_i, |\mu_i|, v_i, \frac{\sqrt{v_i}}{d_i}, \frac{|\mu_i|}{d_i}, \frac{|\mu_i|}{\sigma_i + \varepsilon}, \frac{|\mu_i|}{v_i + \varepsilon} \right). \tag{50}$$

Empirical scores replace these moments with $\hat{\mu}_i = H^{-1} \sum_h g_{h,i}$, $\hat{a}_i = H^{-1} \sum_h |g_{h,i}|$, $\hat{v}_i = H^{-1} \sum_h g_{h,i}^2$, and $\hat{\sigma}_i = \{\max(\hat{v}_i - \hat{\mu}_i^2, 0)\}^{1/2}$. We call $s^{(6)}$ SNR as a descriptive consistency ratio, not a significance test. The implementation keeps only the sums required by the chosen score. Main experiments use SNR, with $H = 5$ on Math10K, $H = 10$ on commonsense reasoning, and $H = 4$ on code generation.

B.2 Online Estimators and Candidate Traversal

This section gives further details of OnlineQuantileSel and OnlineQuantileRep from Section 3.2, together with the candidate-selection settings used in our experiments. Each adapted matrix maintains its own estimator states. The index t denotes a selection round, with H steps between rounds.

Upper-Tail Estimation. For candidate scores $\{\hat{s}_{t,i} : i \in \mathcal{C}_t\}$, we compute the current-round quantile and normalize it by the mean score before updating an exponential moving average:

$$b_t = m^{-1} \sum_{i \in \mathcal{C}_t} |\hat{s}_{t,i}|, \qquad \widetilde{q}_t = Q(1 - \rho_t^{\mathrm{sel}}; \{\hat{s}_{t,i} : i \in \mathcal{C}_t\}),$$
$$z_t = (1 - \beta) z_{t-1} + \beta \frac{\widetilde{q}_t}{\max(b_t, \varepsilon)}, \qquad q_t^{\mathrm{sel}} = b_t z_t. \tag{51}$$

Here Q applied to a finite collection denotes its empirical quantile, $\beta \in (0, 1]$ is the averaging rate, and $\varepsilon > 0$ prevents division by zero. At the first round of Stage I and after the transition to Stage II, we initialize z_t to the normalized current-round quantile. Normalization allows past quantile values to be reused relative to their score scales, while multiplication by b_t returns the estimate to the current score units. We use $\beta = 0.2$. When $\beta = 1$ and $b_t \geq \varepsilon$, the estimate reduces to the current-round quantile. Appendix A.5 analyzes this update, including normalization error and tracking lag.

Lower-Tail Estimation. We implement OnlineQuantileRep in (3) using pseudorandom priorities. The reservoir $\mathcal{V}_t$ retains magnitude–priority records, and the quantile is computed from their magnitudes.

The reservoir starts empty at Stage II and receives each location's magnitude once during its current stay in the trainable set. Later changes to that coefficient, including its removal, do not alter the recorded value. If no new records arrive, the boundary is unchanged. Appendix A.6 gives the corresponding sampling bound under its stated independence conditions.

We use the stability criterion from Section 3.2 with $\tau = 0.01$, $P = 3$, and $\varepsilon_c = 10^{-8}$. Replacement begins after at least 128 stabilization records and 10 Stage II rounds, with $R = 4{,}096$ per matrix. At each round, eligibility is checked using the update's current stability and current magnitude $|\delta_{t,j}| \leq q_t^{\mathrm{rep}}$, not the magnitude stored in the reservoir.

Candidate Traversal and Score Accumulation. The experiments traverse locations through seeded 64×64 blockwise permutations, skipping trainable locations without materializing a matrix-sized permutation. Each candidate set contains m distinct unselected locations and remains fixed for H optimizer steps. A new permutation begins when the current traversal is exhausted. The blocks organize traversal only; the selected weights are individual scalar weights.

Candidates are gradient probes and are not updated by the optimizer. Their gradients are accumulated across microbatches and averaged across distributed workers, so each recorded gradient corresponds to one optimizer step's global batch. This training batch comprises the microbatches accumulated across devices for that step. Over the H steps, we retain only the sums required by the chosen score, such as sums of gradients, absolute gradients, or squared gradients. Weight-normalized scores divide by the corresponding pretrained weight magnitudes, with a numerical offset. Appendix B.1 gives the score formulas.

Per-Round Selection Limits. We use a lower bound (floor) and an upper bound (cap) on the number of candidate proposals per matrix per round as safeguards. Among candidates with finite positive scores, we first select those passing the upper-tail boundary. If their number exceeds the cap, we keep the highest-scoring candidates up to the cap; if it falls below the floor, we add the highest-scoring remaining candidates until the floor is reached or no valid candidates remain. Both operations are restricted to the sampled candidate set. The bounds apply to proposals: actual additions remain subject to the remaining matrix budget in Stage I and the availability of eligible replacement partners in Stage II. Table 4 lists the floor and cap settings. Across the three main experiments, the observed activation frequency of these safeguards in StreamSelect was below 0.1%.

Stage II pairs higher-scoring candidates with eligible weights in increasing order of $c_{t,j}$ and applies the optional replacement decay. Removed updates are discarded, and new coefficients and optimizer states are initialized to zero.

Storage and Computation. Each matrix stores coefficients and optimizer state for at most $\mathcal{B}$ trainable weights, statistics for m candidates, a reservoir of at most R magnitudes, and the scalar selection-estimator state. Persistent adaptation state uses $O(\mathcal{B}+m+R)$ storage. StreamSelect reduces persistent score storage and selection overhead by maintaining multi-step importance statistics only for sampled candidates and avoiding full-matrix importance ranking.

B.3 Main Configurations

Table 4 collects the three main protocols. All use blockwise candidate traversal, SNR scoring, bf16, weight decay zero, and seed 42; paired entries in the two reasoning columns list LLaMA2/LLaMA3.

Code generation. We train on all 111,183 Magicoder-Evol-Instruct-110K examples without a validation split and export the final in-memory adapter. The 224 adapted matrices comprise the seven projections in Table 4 across all 32 layers of LLaMA3-8B-Instruct. Optimization uses cosine decay, 3% warmup, gradient clipping at one, and 869 optimizer steps. We evaluate with EvalPlus 0.3.1 on the canonical 164 HumanEvalPlus and 378 MBPPPlus tasks using greedy decoding, one completion per task, and at most 1,024 new tokens.

The SpIEL runs in Table 5 use the larger, nonuniform trainable budgets reported in the main experiments, rather than the per-matrix budget matched in Section 5.4. In Table 5, H denotes the interval between selection events: SpIEL reselects every 10 steps using gradients accumulated over five steps, whereas StreamSelect accumulates candidate gradient statistics over all H steps between selection rounds.

On Math10K, StreamSelect has 12.3% and 8.4% higher throughput than SpIEL on LLaMA2-7B and LLaMA3-8B. Its StepPeak is 297 MiB higher in the first case and 1,779 MiB lower in the second. On code generation, it uses 68.0% fewer coefficients, has 19.2% lower StepPeak, and has 22.8% higher throughput. These are end-to-end measurements under each method's native allocation. The code runs also differ in sparse optimizer, selection schedule, and learning rate, so the measured differences reflect the complete training configurations.

For Math10K, two GPUs use microbatch 4 each and gradient accumulation one, giving 3,534 optimizer steps. The linear scheduler has warmup ratio 0.03, maximum gradient norm one, and

Table 4: StreamSelect configurations for the reported main results. Paired entries list LLaMA2 / LLaMA3; a dash under decay denotes no gradual pre-cutoff decay.

Setting	Commonsense LLaMA2/LLaMA3	Math10K LLaMA2/LLaMA3	Code/LLaMA3-8B Instruct
Trainable parameters	28.80M (.43%/.36%)	28.80M (.43%/.36%)	53.760M (.67%)
Adapted projections	q/k/v/up/down	q/k/v/up/down	q/k/v/o/gate/up/down
$\mathcal{B}$ / m	180k / 10k	180k / 40k	240k / 80k
H	10	5	4
Fill quantile / cap	q75 / 5k	q87.5 / 10k	q87.5 / 20k
Grow quantile / floor / cap	q95 / 100 / 1k	q97.5 / 100 / 2k	q97.5 / 100 / 4k
ρ^{rep} / τ / P	.10 / .01 / 3	.30 / .01 / 3	.30 / .01 / 3
Quantile update β	.2	.2	.2
Replacement decay λ	–	2	2
Fixed-support start	70%	60%	50%
Learning rate	$1.5 \times 10^{-4}/8 \times 10^{-5}$	$4 \times 10^{-4}/5 \times 10^{-5}$	1.5×10^{-4}
Epochs / global batch	3 / 32	3 / 8	1 / 128
Maximum sequence length	256	512	3,072
Evaluation beams	1	4	1

Table 5: Native-budget end-to-end efficiency. StepPeak is peak allocated memory, and throughput includes each method's training and selection work. Comparisons are within each task/model block.

Task/model	Method	Trainable	H	StepPeak	Steps/s
Math / LLaMA2-7B	SpIEL	56.10M	10	26,433 MiB	1.566
	StreamSelect	28.80M	5	26,730 MiB	1.758
Math / LLaMA3-8B	SpIEL	56.10M	10	33,938 MiB	1.640
	StreamSelect	28.80M	5	32,159 MiB	1.778
Code / LLaMA3-8B-Instruct	SpIEL	167.772M	10	48,811 MiB	0.03048
	StreamSelect	53.760M	4	39,428 MiB	0.03742

validation checkpoints at 75, 80, 85, 90, 95, and 100% of training. All runs complete the prescribed training horizon; the exported adapter is the eligible checkpoint with minimum validation loss. Candidate statistics are accumulated over optimizer steps and averaged across distributed ranks before scoring. Evaluation uses deterministic beam search with four beams.

Only sparse coefficients are registered with the optimizer, so trainable parameter and optimizer state scale with $\mathcal{B}$, while persistent candidate statistics scale with m.

B.4 Controlled Ablations

Table 6 examines replacement intensity and the choice of importance score on Math10K. In panel (a), the intermediate replacement setting achieves the highest Avg. and Weighted Avg. among the tested settings, suggesting a balance between admitting new locations and retaining learned updates. Panel (b) compares seven scores under the same 40k/$H = 5$ policy with the remaining settings held fixed; SNR achieves the highest Avg. and Weighted Avg., supporting its use in the main experiments. StepPeak and throughput are omitted from the score comparison because they vary by only 49 MiB and 0.032 steps/s, respectively.

B.5 Motivation Protocols

A model with a fixed random support of 180k locations in each of 160 target matrices is trained for 100 steps. The probe uses 100,000 fixed inactive locations in layer 15 q-projection. At model states $0, 5, \ldots, 100$, the same five global calibration batches are evaluated without updates, and the dropout RNG is reset so masks are identical across states and then restored so calibration does not alter

Table 6: Controlled Math10K ablations. **(a)** Realized turnover intensity; Repl./$\mathcal{B}$ divides total replacements by the 28.8M full-model active budget. **(b)** Seven importance scores under the same 40k/$H = 5$ policy. Score formulas use the empirical moments in Appendix B.1, with $d_i = |(W_0)_i| + \varepsilon$. Bold and underline mark the best and second-best accuracy in each column.

(a) Turnover intensity

Scale	ρ^{rep}	$1 - \rho_t^{\text{sel}}$	Repl./$\mathcal{B}$	GSM8K	AQuA	MAWPS	SVAMP	Avg.	Weighted Avg.
0.8×	0.24	0.980	0.370	<u>47.61</u>	26.77	<u>86.97</u>	**62.30**	<u>55.92</u>	<u>54.29</u>
1.0×	0.30	0.975	0.465	**48.52**	**28.35**	**88.24**	**62.30**	**56.85**	**54.96**
1.2×	0.36	0.970	0.565	45.19	<u>27.17</u>	**88.24**	<u>62.20</u>	55.70	53.26

(b) Importance score

Score	GSM8K	AQuA	MAWPS	SVAMP	Avg.	Weighted Avg.
$\hat{a}_i$	45.49	25.98	86.13	**62.70**	55.08	53.29
$\lvert\hat{\mu}_i\rvert$	<u>47.69</u>	24.80	84.03	60.70	54.31	53.33
$\hat{v}_i$	44.05	20.87	86.13	59.90	52.74	51.16
$\sqrt{\hat{v}_i}/d_i$	46.78	23.62	<u>87.39</u>	60.40	54.55	52.97
$\lvert\hat{\mu}_i\rvert/d_i$	<u>47.69</u>	<u>26.77</u>	<u>87.39</u>	60.30	55.54	53.65
SNR	**48.52**	**28.35**	**88.24**	62.30	**56.85**	**54.96**
$\lvert\hat{\mu}_i\rvert/(\hat{v}_i + \varepsilon)$	46.78	25.59	<u>87.39</u>	<u>62.60</u>	<u>55.59</u>	<u>53.93</u>

training. Empirical SNR is computed offline using the moment-based variance estimator defined in Appendix B.1. Relative to step 0, Spearman/Top-10% Jaccard are 0.895/0.664 at step 20, 0.466/0.237 at step 40, and 0.203/0.116 at step 100. The probe isolates state-dependent change in one matrix under one fixed-support trajectory; it does not estimate a universal drift rate.