Selective Fine-tuning via Excess Loss for Enhanced Reasoning in Large Language Models

Anonymous ACL submission

Abstract

While supervised fine-tuning on chain-of-thought (CoT) traces can markedly boost reasoning capabilities of large language models (LLMs), not all tokens in a CoT trace equally contribute to that gain. We propose a selective fine-tuning framework that embeds the token-selection ideas of Selective Language Modeling (SLM) into reasoning-oriented training. In specific, by measuring each token's excess loss with a reference model, we pinpoint the fragments most critical to reasoning and apply one of three tailored objectives: token-selective, token-weighted, or segment-selective, so gradient updates focus only on those high-value tokens or spans. When applied to Qwen2.5-1.5B and evaluated on GSM8K and MATH, this strategy outperforms standard fine-tuning, with the token-selective variant raising accuracy by up to 5.6 percentage points. This approach not only enhances model performance and training efficiency, but also improves the coherence and reliability of multi-step reasoning, offering a scalable solution for developing advanced reasoning models.

1 Introduction

Large Language Models (LLMs) have transformed natural language processing, achieving strong zero- and few-shot results on translation, question answering, and basic reasoning without task-specific fine-tuning (Brown et al., 2020). Yet they still struggle with complex multi-step problems that demand explicit logical deduction or arithmetic. Supplying *chain-of-thought* (CoT) rationales—intermediate reasoning traces—substantially narrows this gap (Wei et al., 2023): prompting LLMs to "think step by step" allows them to solve arithmetic and commonsense tasks far better than direct-answer baselines. Researchers have further transplanted this skill into smaller models by fine-tuning them on CoT corpora produced by stronger teachers (Ho et al., 2023), enabling resource-efficient deployment.

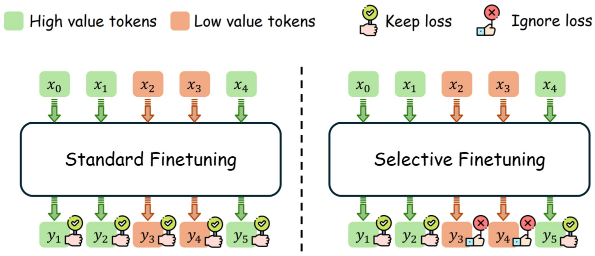

Figure 1: Poorly curated or noisy data can hinder deep-reasoning development. **Left:** Standard fine-tuning applies loss uniformly to all tokens. **Right:** Our proposed *selective fine-tuning* applies loss only to the most informative tokens, ignoring the rest.

Despite these advances, LLM reasoning remains unreliable. The auto-regressive generation process means that an early hallucination can cascade through a long rationale and yield an incorrect conclusion (Ferrag et al., 2025). Models also exhibit weak self-consistency, sometimes contradicting themselves within a single chain. Demonstrating that a model *can* reason in controlled settings is therefore insufficient; we must ensure it does so *consistently* across diverse problems (Xu et al., 2025; Boye and Moell, 2025; Li et al., 2025).

Supervised fine-tuning (SFT) is an attractive avenue because it integrates reasoning knowledge directly into model weights. Conventional next-token SFT, however, treats every token equally, wasting capacity on trivial continuations and failing to target fragile parts of long CoT sequences. Its effectiveness is therefore highly sensitive to data cleanliness; noisy examples can degrade reasoning rather than enhance it.

We address these issues with a *selective fine-tuning* framework that leverages a reference model and the notion of *excess loss*. Inspired by "Not All Tokens Are What You Need" (Lin et al., 2025), we first train a strong reference model and compute its token-level loss on the training set. The excess loss—the difference between the candidate model's loss and the reference loss—identifies difficult tokens. During fine-tuning we focus the gradient on high-excess-loss tokens while down-weighting the rest. By doing so, the model devotes more capacity to learning the non-trivial reasoning components that it hasn't mastered, rather than over-processing tokens it already predicts well. This simple selective fine-tuning strategy is designed to both improve training efficiency and steer the model's optimization toward better reasoning performance. In summary, our approach directly tackles the inefficiency of uniform token-level training by prioritizing the most informative tokens (as identified via a reference model), thereby aiming to produce an LLM that is more adept at multi-step reasoning.

The empirical results demonstrate that our proposed selective fine-tuning approach substantially improves reasoning performance. When applied to Qwen2.5-1.5B on the GSM8K and MATH benchmarks, token-selective fine-tuning achieves 68.5% and 41.2% accuracy respectively, representing increases of 4.8 percentage points over standard fine-tuning in both cases. Our token-weighted variant performs even better, achieving 68.4% on GSM8K and 42.0% on MATH. These improvements are particularly notable on the more challenging MATH benchmark, suggesting that selective fine-tuning effectively targets the difficult reasoning steps essential for complex problem-solving.

The main contributions of this paper are: (1) A novel selective fine-tuning framework that leverages excess loss to identify and prioritize the most informative tokens in chain-of-thought reasoning traces; (2) Three complementary selective loss functions that operate at different granularities (token-selective, token-weighted, and segment-selective); (3) Empirical evidence that focusing on high-value tokens during fine-tuning significantly enhances reasoning capabilities while improving training efficiency; and (4) Insights from ablation studies on the importance of reference model quality, adaptive selection strategies, and curated training data for optimal reasoning performance.

2 Related Work

Data–efficient supervised fine-tuning. A growing body of evidence shows that high-quality, well-matched data can rival—or outperform—training on the full corpus during the SFT stage (Liu et al., 2025b). Liu et al. (2024) cast data selection as an optimal-transport alignment problem, adding a diversity regularizer and demonstrating that ~1 % of carefully picked examples can beat 100 % of the data. In a complementary line, Yang et al. (2024) fine-tune a small proxy model, cluster examples by their loss trajectories, and then train the target LLM only on clusters deemed most useful.

Selectivity *during* training. Rather than filter examples beforehand, several approaches inject selectivity into the loss computation itself. SelectIT (Liu et al., 2025a) prompts the model to solve an instruction and self-evaluate its confidence; instructions that trigger low confidence or wrong answers receive greater weight in subsequent updates. LESS (Xia et al., 2024) estimates each example's influence on a probe set via low-rank gradient sketches, selecting those with gradients most aligned to desired skills.

Token-level focus in pretraining Whole-example selection can still waste effort on uninformative tokens. Mindermann et al. (2022) introduced RHO-LOSS, which scores data points by how much they are expected to reduce hold-out loss, avoiding noisy or already-mastered content. Li et al. (2025) adapt this principle to LLMs with *Selective Language Modeling* (SLM): a reference model assigns each token an *excess loss* score, and the main model updates only on high-score tokens, ignoring or down-weighting the rest.

Together, these studies suggest that both *which* examples and *which* tokens receive gradient signal critically influence downstream reasoning performance. Our work extends this idea by combining reference-based excess-loss scoring with an efficient, end-to-end fine-tuning routine tailored for long chain-of-thought sequences.

3 Selective Fine-tuning for Reasoning

Our selective fine-tuning framework, illustrated in Figure 2, consists of three key phases: (1) Training a reference model on high-quality, curated CoT samples; (2) Using this reference model to calculate excess loss on tokens in the main training dataset; and (3) fine-tuning the main model with one of three selective loss strategies that focus gradient updates on the most informative portions of reasoning traces.

The core insight is that not all tokens in a CoT trace contribute equally to reasoning performance.

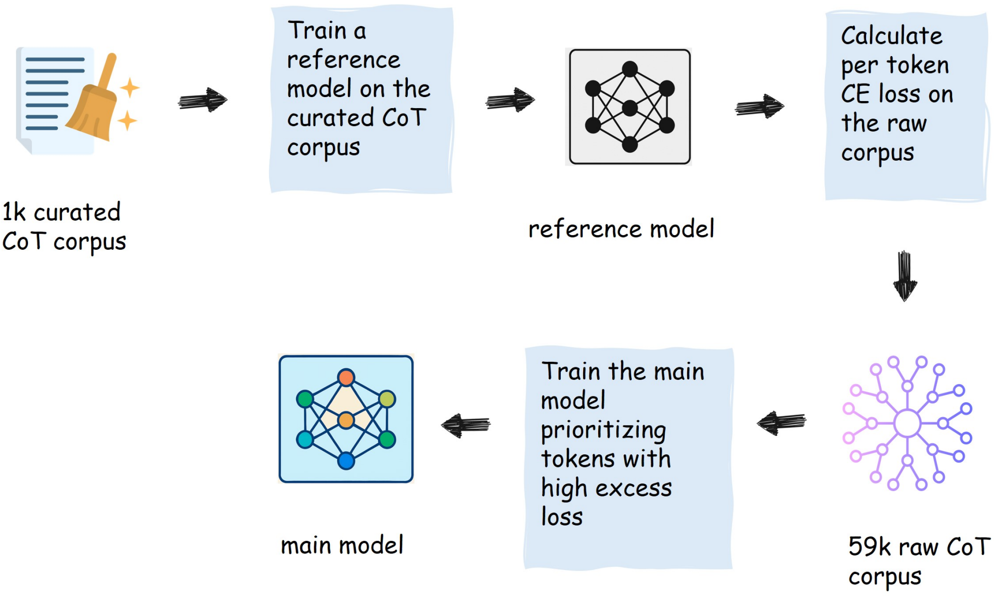

Figure 2: **The pipeline of Selective Fine-tuning via Excess Loss.** This includes three major steps: training a reference model on the curated CoT samples, calculating the excess loss using the reference model on the raw CoT samples during the main model fine-tuning process, and finally fine-tuning the main model on the most valuable tokens based on the excess loss.

By comparing the token-level loss of our candidate model against a strong reference model, we identify tokens that are particularly challenging or informative, and concentrate the learning process on these high-value regions. This approach efficiently allocates computational resources while preventing the model from overfitting to template language or trivial patterns in the training data.

We extend *Selective Language Modeling* (SLM) to the SFT phase of chain-of-thought traces by introducing three complementary loss functions that decide *where* to spend gradient budget:

- *Token-Selective Loss* (§3.1) — trains only on the hardest tokens,
- *Token-Weighted Loss* (§3.2) — reweights every token by its difficulty,
- *Segment-Selective Loss* (§3.3) — selects whole reasoning steps instead of single tokens.

All three rely on a *reference model* to measure *excess loss*: the difference between the main model's token loss and the reference's. The higher the excess loss, the more a token (or segment) can teach the model. The following subsections detail each method, focusing on their implementation and application during the SFT stage for enhancing reasoning capabilities in long CoT data.

3.1 Token-Selective Loss

The token-selective loss method, inspired by the SLM approach from Li et al. (2025), prioritizes training on a subset of tokens with high excess loss during the fine-tuning stage. For a given batch of input sequences $\mathbf{x} = \{x_1, \ldots, x_T\}$ and corresponding target tokens $\mathbf{y} = \{y_1, \ldots, y_T\}$, we compute the per-token cross-entropy loss for both the main model M and the reference model R:

$$L_M = \sum_{t=1}^{T} \ell_M(t) = -\sum_{t=1}^{T} \log P_M(y_t \mid x, y_{<t}),$$

$$L_R = \sum_{t=1}^{T} \ell_R(t) = -\sum_{t=1}^{T} \log P_R(y_t \mid x, y_{<t}).$$

where P_M and P_R are the probability distributions over the vocabulary predicted by the main and reference models, respectively. The **excess loss** is then defined as:

$$L_\Delta(t) = L_M(t) - L_R(t).$$

Tokens with an excess loss above a threshold, determined by a selection ratio $k \in [0, 1]$, are selected for training. Specifically, we select the top $k \cdot N$ tokens with the highest $L_\Delta(t)$, where N is the number of valid tokens. A binary mask $\mathbf{m}$ is

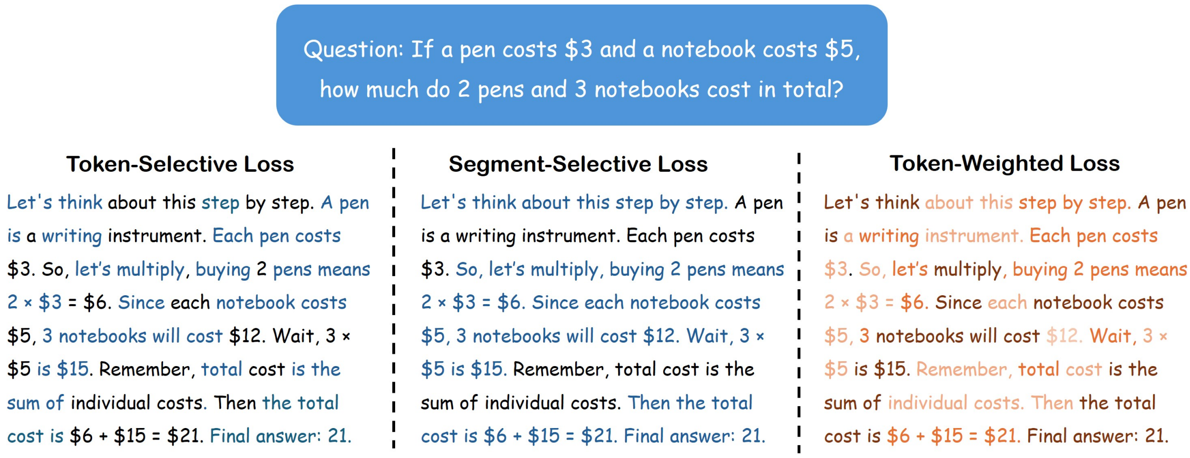

Figure 3: **Selective Fine-tuning for CoT Reasoning Tasks.** This diagram illustrates various selective fine-tuning strategies applied to CoT reasoning. In the left and middle examples, blue tokens denote positions receiving gradient updates, while **black** tokens are excluded during fine-tuning. In the right example, tokens are colored in shades of orange, where darker orange indicates higher weight. Selective fine-tuning strategically filters noisy or redundant information, concentrating learning on the most informative reasoning components to improve LLM performance on multi-step reasoning tasks. *Note: This is a simplified, illustrative scenario and does not fully capture the complexity of real token dynamics in large-scale training. Additional examples are available in the Appendix 5.6.

created, where $m_t = 1$ if token t is selected and $m_t = 0$ otherwise. The final loss is computed as:

$$\mathcal{L}_{\text{token-selective}} = \sum_{t=1}^{T} m_t \cdot \text{CE}(\hat{\mathbf{y}}_t, y_t),$$

where CE is the cross-entropy loss, and $\hat{\mathbf{y}}_t$ are the logits from the main model. To balance comprehensive learning and selective efficiency, we employ a linear decay scheduler for the selection ratio, starting at $k = 1.0$ in the first epoch (full-data fine-tuning) and decaying to $k = 0.6$ by the final epoch.

3.2 Token-Weighted Loss

The token-weighted loss method extends the token-selective approach by assigning continuous weights to tokens based on their excess loss, rather than a binary selection. Using the same excess loss $L_\Delta(t)$ as defined above, we compute weights via a softmax function with a temperature parameter τ:

$$w_t = \frac{\exp(L_\Delta(t)/\tau)}{\sum_{s \in \mathcal{V}} \exp(L_\Delta(s)/\tau)},$$

where $\mathcal{V}$ is the set of valid tokens, and $\tau = 2$ controls the softness of the weight distribution. These weights are applied to the per-token cross-entropy loss to compute the final loss:

$$\mathcal{L}_{\text{token-weighted}} = \frac{\sum_{t \in \mathcal{V}} w_t \cdot \text{CE}(\hat{\mathbf{y}}_t, y_t)}{\sum_{t \in \mathcal{V}} w_t + \epsilon},$$

where $\epsilon = 10^{-8}$ prevents division by zero. This method allows the model to focus on tokens with higher excess loss while still considering all valid tokens, potentially improving generalization compared to the binary selection in token-selective loss.

3.3 Segment-Selective Loss

For long CoT data, where reasoning errors often arise from entire steps rather than individual tokens, we propose a segment-selective loss that operates at a higher granularity. Segments are defined as coherent units of reasoning steps, identified using an unsupervised approach with sentence embeddings and DBSCAN clustering. Given segment labels $\mathbf{s} = \{s_1, \dots, s_T\}$ that assign each token to a segment (or an ignore index for padding), we compute the excess loss $L_\Delta(t)$ as in the token-selective method. The segment-level excess loss is then calculated by averaging the per-token excess loss within each segment:

$$L_\Delta^{\text{seg}}(i) = \frac{1}{|\mathcal{S}_i|} \sum_{t \in \mathcal{S}_i} L_\Delta(t),$$

where $\mathcal{S}_i$ is the set of tokens in segment i, and $|\mathcal{S}_i|$ is the number of tokens in that segment. The top $k \cdot G$ segments with the highest $L_\Delta^{\text{seg}}(i)$ are selected, where G is the number of unique segments, and k is the selection ratio. A binary mask $\mathbf{m}$ is created such that $m_t = 1$ if token t belongs to a selected segment and $m_t = 0$ otherwise. The loss is then:

$$\mathcal{L}_{\text{segment-selective}} = \sum_{t=1}^{T} m_t \cdot \text{CE}(\hat{\mathbf{y}}_t, y_t).$$

This approach preserves the contextual integrity of reasoning steps, ensuring that the model trains on semantically meaningful units.

3.4 Practical Integration

All three methods are integrated into the SFT stage using the LitGPT framework, as described in the provided code. The reference model—trained on high-quality instruction data—provides stable $\ell_R(t)$. The selection ratio k for selective methods is dynamically adjusted using a linear decay scheduler, and the temperature τ for weighted methods is fixed at 5.0. Experiments employ Qwen2.5-1.5B (131k-token context), so long CoT sequences fit without truncation.

4 Experimental Setup

All experiments are carried out in LITGPT using the 1.5 B-parameter **Qwen2.5** backbone, whose 131 K token window comfortably accommodates full chain-of-thought (CoT) transcripts.

4.1 Reference Model

We first train a reference model—also Qwen2.5-1.5B—on the curated S1k instruction set from Muennighoff et al. (2025). The set contains 1 000 high-quality examples with detailed reasoning traces. Training lasts 5 epochs with a cosine-decayed learning-rate peak of 5×10^{-5}, 15 warm-up steps, 20 000-token sequences, and a per-device batch size of 1 (to avoid out-of-memory).

4.2 Fine-tuning Data

For the main experiments we use the authors' larger, unfiltered split (59k examples). Its noise and diversity provide a realistic stress test for selective fine-tuning.

4.3 Methods Compared

We evaluate four SFT strategies, each run for 3 epochs on the 59k corpus with the same scheduler and hyper-parameters as the reference model (batch 1, max-LR 5×10^{-5}):

- **Full fine-tuning** (baseline): standard cross-entropy on every token.
- **Token-Selective Loss**: hard-masking the top k tokens by excess loss, with k linearly annealed from 1.0 to 0.6.
- **Token-Weighted Loss**: soft weights from a τ=5 softmax over excess loss.
- **Segment-Selective Loss**: hard selection of entire reasoning segments, using the same k schedule as (2).

4.4 Datasets

Following Lin et al. (2025), we measure few-shot CoT accuracy on GSM8K (grade-school arithmetic) and MATH (advanced competition problems). Accuracy is reported as the percentage of problems solved correctly under identical prompting conditions.

5 Results

Research questions. Our experiments address five questions:

- *RQ1* — Does selective fine-tuning beat standard full fine-tuning on chain-of-thought (CoT) reasoning?
- *RQ2* — Which granularity is more helpful, token-level or segment-level?
- *RQ3* — Is soft weighting superior to hard masking?
- *RQ4* — How does the amount and quality of reference data influence token selection?
- *RQ5* — How important is the choice of the selection-ratio scheduler?

5.1 Effects of Selective Fine-tuning for Reasoning

Table 1 presents the CoT reasoning performance of the baseline and proposed methods, alongside reference base models, on the GSM8K and MATH benchmarks.

The **token-selective loss** method, which prioritizes tokens with high excess loss $L_\Delta(t)$, achieved 68.5% on GSM8K and 41.2% on MATH, yielding an average accuracy of 54.9%. This represents improvements of 4.8% and 4.8% over the baseline full fine-tuning (63.7% on GSM8K, 36.4% on MATH) on the same dataset, demonstrating the effectiveness of focusing training on informative tokens.

Table 1: Few-shot CoT reasoning results on math benchmarks. $|\theta|$ denotes the number of parameters, and Data indicates the fine-tuning dataset (s1k: 1,000 samples; s1-full59k: 59,000 samples).

Model	$\|\theta\|$	**Data**	**GSM8K**	**MATH**	**Average**
		Base Models			
Gemma3	4B	-	38.4	24.2	31.3
Mistral	7B	-	52.2	13.1	32.7
Qwen2.5	1.5B	-	52.5	31.6	42.0
		Fine-tuning on Qwen2.5-1.5B			
Full fine-tuning (baseline)	1.5B	s1k	65.6	31.8	48.7
Full fine-tuning (baseline)	1.5B	s1-full59k	63.7	36.4	50
Selective fine-tuning (Token-level)	1.5B	s1-full59k	**68.5**	41.2	**54.9**
Selective fine-tuning (Segment-level)	1.5B	s1-full59k	67.9	37.2	52.6
Weighted fine-tuning (Token-level)	1.5B	s1-full59k	68.4	**42.0**	**55.2**

The method's strong MATH performance highlights its capability in addressing complex, multi-step reasoning challenges by optimizing for tokens critical to logical inference.

The **segment-selective loss** method, operating at the granularity of reasoning steps, achieved 67.9% on GSM8K and 37.2% on MATH, with an average of 52.6%. While it outperformed the baseline on both benchmarks, its MATH performance lagged behind the token-selective method by 4.0%. This suggests that segment-level selection, while preserving contextual integrity, may be less effective for tasks requiring fine-grained token-level reasoning, such as advanced mathematical problems.

The **token-weighted loss** method, which assigns continuous weights based on $L_\Delta(t)$, achieved 68.4% on GSM8K and 42.0% on MATH, yielding the highest average accuracy of 55.2% among all methods. Its balanced performance across both datasets suggests that continuous weighting offers a good trade-off between generalization and specificity, supporting both simple and complex reasoning tasks.

Overall, the proposed selective and weighted loss functions significantly enhance SFT efficiency and reasoning performance compared to standard full fine-tuning. Among them, the **token-weighted loss** method achieves the highest overall accuracy, while the **token-selective loss** method remains highly robust, particularly on more complex benchmarks like MATH. These results validate the hypothesis that prioritizing high-excess-loss tokens or segments enables more effective learning of complex reasoning patterns. The ablation study further elucidates the importance of adaptive selection ratios and curated reference data in achieving these gains.

5.2 Token-level vs. Segment-level Granularity

The segment-selective loss method, operating at the granularity of reasoning steps, achieved 67.9% on GSM8K and 37.2% on MATH, with an average of 52.6%. While it outperformed the baseline on both benchmarks, its MATH performance lagged behind the token-selective method by 4.0%. This suggests that segment-level selection, while preserving contextual integrity, may be less effective for tasks requiring fine-grained token-level reasoning, such as advanced mathematical problems.

Our results indicate that token-level approaches generally outperform segment-level methods, particularly on tasks that require precise, step-by-step reasoning like those found in the MATH benchmark. However, segment-level selection still offers meaningful improvements over the baseline, suggesting that the optimal granularity may depend on the specific reasoning task.

5.3 Hard Masking vs. Soft Weighting

The token-weighted loss method, which assigns continuous weights based on $L_\Delta(t)$, achieved 68.4% on GSM8K and 42.0% on MATH, yielding the highest average accuracy of 55.2% among all methods. Its balanced performance across both datasets suggests that continuous weighting offers a good trade-off between generalization and speci-

Table 2: Ablation study results on GSM8K and MATH benchmarks for the Qwen2.5 1.5B model fine-tuned on the 59,000-sample dataset.

Method	Configuration	GSM8K	MATH	Average
Token Selection Method				
1,000 Samples, Linear Decay	Default	68.5	**41.2**	**54.9**
500 Samples, Linear Decay	Ablation	67.6	33.8	50.7
100 Samples, Linear Decay	Ablation	**70.0**	36.8	53.4
Random 1,000 Samples, Linear Decay	Self-Reference	67.9	34.4	51.2
1,000 Samples, Fixed Ratio (0.6)	Ablation	69.6	33.8	51.7
1,000 Samples, Fixed Ratio (0.8)	Ablation	69.1	36.8	52.9

ficity, supporting both simple and complex reasoning tasks.

When comparing hard masking (token-selective) versus soft weighting approaches, we find that soft weighting tends to perform marginally better, particularly on more complex reasoning tasks like MATH. This advantage may stem from the weighted approach's ability to maintain a more nuanced gradient signal that preserves contextual information while still prioritizing high-value tokens.

5.4 Impact of Reference Data Quality and Quantity

To investigate the influence of reference data on token selection, we conducted ablation experiments with varying amounts of curated data for training the reference model. The default configuration using 1,000 high-quality samples achieved the strongest overall performance (54.9% average accuracy).

Reducing the reference dataset to 500 samples resulted in a performance drop to 67.6% on GSM8K, 33.8% on MATH, and an average of 50.7%. The significant decline in MATH performance suggests that a smaller reference dataset compromises the accuracy of excess loss estimation, particularly for tasks requiring deep, multi-step reasoning.

Interestingly, using just 100 samples achieved the highest GSM8K performance at 70.0%, but a lower MATH score of 36.8%, yielding an average of 53.4%. This indicates that for simpler reasoning tasks, even a small high-quality reference set can effectively identify informative tokens, while complex reasoning benefits from larger reference datasets.

Furthermore, our "Self-Reference" configuration using 1,000 randomly selected (rather than curated) samples achieved only 67.9% on GSM8K and 34.4% on MATH (51.2% average), highlighting the importance of high-quality reference data for effective token selection.

5.5 Selection Ratio Scheduler Optimization

Our experiments compared different selection ratio schedulers for the token-selective method. The linear decay scheduler (gradually reducing selection from 1.0 to 0.6) outperformed fixed ratios, achieving 68.5% on GSM8K and 41.2% on MATH (54.9% average).

Fixed selection ratios of 0.6 and 0.8 achieved 69.6%/33.8% and 69.1%/36.8% on GSM8K/MATH respectively, resulting in averages of 51.7% and 52.9%. The significant performance drop on MATH with fixed ratios suggests that adaptive selection is particularly important for complex reasoning tasks.

The linear decay approach provides an optimal balance by allowing the model to learn broadly in early epochs and gradually focus on high-excess-loss tokens as training progresses. This strategy appears crucial for developing robust reasoning capabilities across diverse problem types.

5.6 Qualitative Examples

Figure 4 and 5 shows the complete CoT trace with tokens and segmented selected. The tokens marked in blue represent the actual tokens trained while the remaining **black** tokens are not trained during the fine-tuning process.

6 Conclusion

This paper introduces a novel selective fine-tuning framework for LLMs, centered on token and segment-level excess loss to enhance complex reasoning through supervised fine-tuning. By lever-

```
think The set of numbers is $\{87, 85, 80, 83, 84, x\}$. The mean of the set is 83.5. The number of elements in the set is 6. The mean of a set of numbers is the sum of the numbers divided by the number of elements. So, the sum of the numbers is the mean multiplied by the number of elements. Sum of the numbers = $83.5 \times 6$ Sum of the numbers = $501$ The sum of the given numbers is $87 + 85 + 80 + 83 + 84 + x$. $87 + 85 + 80 + 83 + 84 + x = 501$ $419 + x = 501$ $x = 501 - 419$ $x = 82$ Now the set of numbers is $\{87, 85, 80, 83, 84, 82\}$. To find the median of the set, we need to order the numbers from least to greatest. Ordered set: $\{80, 82, 83, 84, 85, 87\}$ Since there are an even number of elements in the set (6), the median is the average of the two middle numbers. The two middle numbers are the $\frac{6}{2} = 3$rd and the $\frac{6}{2} + 1 = 4$th numbers in the ordered set. The 3rd number is 83. The 4th number is 84. Median = $\frac{83 + 84}{2}$ Median = $\frac{167}{2}$ Median = $83.5$ The mean of the set of numbers $\{87,85,80,83,84,x\}$ is 83.5. The number of elements is 6. The sum of the elements is $6 \times 83.5 = 501$. The sum of the given numbers is $87+85+80+83+84+x = 419+x$. So, $419+x = 501$. $x = 501 - 419 = 82$. The set of numbers is $\{87, 85, 80, 83, 84, 82\}$. Order the numbers from least to greatest: $\{80, 82, 83, 84, 85, 87\}$. Since there are 6 numbers, the median is the average of the middle two numbers. The middle two numbers are the 3rd and 4th numbers, which are 83 and 84. Median = $\frac{83+84}{2} = \frac{167}{2} = 83.5$. Express the answer as a decimal to the nearest tenth. The median is 83.5. answer Let the set of numbers be $S = \{87, 85, 80, 83, 84, x\}$. The mean of the set is given as 83.5. The number of elements in the set is 6. The formula for the mean of a set of numbers is the sum of the numbers divided by the number of elements. Mean $= \frac{\text{Sum of the numbers}}{\text{Number of elements}}$ $83.5 = \frac{87 + 85 + 80 + 83 + 84 + x}{6}$ Multiply both sides by 6 to find the sum of the numbers: Sum of the numbers $= 83.5 \times 6$ Sum of the numbers $= 501$ Now, we can write the equation for the sum of the numbers: $87 + 85 + 80 + 83 + 84 + x = 501$ Add the known numbers: $419 + x = 501$ Subtract 419 from both sides to solve for x: $x = 501 - 419$ $x = 82$ Now we have the complete set of numbers: $\{87, 85, 80, 83, 84, 82\}$. To find the median, we need to arrange the numbers in ascending order: $\{80, 82, 83, 84, 85, 87\}$ Since there are an even number of elements (6), the median is the average of the two middle numbers. The middle numbers are the $\frac{6}{2} = 3$rd and the $\frac{6}{2} + 1 = 4$th numbers in the ordered set. The 3rd number is 83. The 4th number is 84. Median $= \frac{83 + 84}{2}$ Median $= \frac{167}{2}$ Median $= 83.5$ The median of the set of six numbers is 83.5. The question asks to express the answer as a decimal to the nearest tenth. In this case, 83.5 is already in that format. Final Answer: The final answer is $\boxed{83.5}$
```

Figure 4: An example of selecting specific **tokens** during the fine-tuning process

```
think The problem is about the energy threshold for the generation of two electronically excited hydrogen atoms from a hydrogen molecule and determining the bond energy of the hydrogen molecule. The reaction is: H2(v=0) -> H*(n=2) + H*(n=2) The energy threshold for this reaction is given as 24.9 eV. We need to determine the bond energy of H2. The energy required for this reaction can be considered as the sum of the energy required to break the H-H bond and the energy required to excite each hydrogen atom to the n=2 state. Let E_C be the bond energy of H2. The energy required to break the H-H bond is E_C. This will result in two ground state hydrogen atoms: H2 -> H(n=1) + H(n=1) Energy = E_C The energy required to excite a hydrogen atom from the ground state (n=1) to the first excited state (n=2) can be calculated using the energy levels of the hydrogen atom. The energy of an electron in the nth energy level of a hydrogen atom is given by: E_n = -13.6 eV / n^2 For n=1 (ground state): E_1 = -13.6 eV / 1^2 = -13.6 eV For n=2 (first excited state): E_2 = -13.6 eV / 2^2 = -13.6 eV / 4 = -3.4 eV The energy difference between these two levels is the excitation energy: Î”E = E_2 - E_1 = -3.4 eV - (-13.6 eV) = -3.4 eV + 13.6 eV = 10.2 eV Since there are two hydrogen atoms being excited to the n=2 state, the total excitation energy for both atoms is 2 * Î”E. Total excitation energy = 2 * 10.2 eV = 20.4 eV The total energy threshold for the reaction is the sum of the bond energy and the total excitation energy: Energy Threshold = E_C + Total excitation energy Given that the energy threshold is 24.9 eV: 24.9 eV = E_C + 20.4 eV Now we can solve for the bond energy E_C: E_C = 24.9 eV - 20.4 eV E_C = 4.5 eV The bond energy of H2 is 4.5 eV. We need to provide the answer to the first decimal place. In this case, it is already to the first decimal place. Let's double check the calculations. Energy levels of hydrogen atom: E_n = -13.6/n^2 eV Energy of H atom in ground state (n=1): E_1 = -13.6 eV Energy of H atom in excited state (n=2): E_2 = -13.6/4 = -3.4 eV Excitation energy for one H atom: E_2 - E_1 = -3.4 - (-13.6) = 10.2 eV Excitation energy for two H atoms: 2 * 10.2 eV = 20.4 eV The total energy required for the process is the sum of the bond dissociation energy and the excitation energies of the two H atoms. Energy threshold = Bond energy + Excitation energy of H1 + Excitation energy of H2 24.9 eV = E_C + 10.2 eV + 10.2 eV 24.9 eV = E_C + 20.4 eV E_C = 24.9 eV - 20.4 eV E_C = 4.5 eV The bond energy of H2 is 4.5 eV. The question asks for the value to the first decimal place, which it already is. answer The energy threshold for the generation of two electronically excited hydrogen atoms $\mathrm{H}^{\star}(n=2)$ from $\mathrm{H}_{2}(v=0)$ is given by the energy required to break the bond in $\mathrm{H}_{2}$ plus the energy required to excite each of the resulting hydrogen atoms to the $n=2$ state. The reaction is: $\mathrm{H}_{2}(v=0) \rightarrow \mathrm{H}^{\star}(n=2) + \mathrm{H}^{\star}(n=2)$ The energy threshold for this reaction is $24.9 \mathrm{eV}$. The process can be broken down into two steps: 1. Breaking the bond in $\mathrm{H}_{2}$ to form two ground state hydrogen atoms: $\mathrm{H}_{2} \rightarrow \mathrm{H}(n=1) + \mathrm{H}(n=1)$ The energy required for this step is the bond energy $E_{\mathrm{C}}$. 2. Exciting each ground state hydrogen atom to the $n=2$ state: $\mathrm{H}(n=1) \rightarrow \mathrm{H}^{\star}(n=2)$ The energy of an electron in the $n$-th energy level of a hydrogen atom is given by the formula: $E_n = -\frac{13.6}{n^2} \mathrm{eV}$ For the ground state ($n=1$): $E_1 = -\frac{13.6}{1^2} = -13.6 \mathrm{eV}$ For the first excited state ($n=2$): $E_2 = -\frac{13.6}{2^2} = -\frac{13.6}{4} = -3.4 \mathrm{eV}$ The energy required to excite a hydrogen atom from $n=1$ to $n=2$ is: $\Delta E = E_2 - E_1 = -3.4 \mathrm{eV} - (-13.6 \mathrm{eV}) = -3.4 \mathrm{eV} + 13.6 \mathrm{eV} = 10.2 \mathrm{eV}$ Since there are two hydrogen atoms being excited, the total excitation energy is $2 \times 10.2 \mathrm{eV} = 20.4 \mathrm{eV}$. The total energy threshold for the reaction is the sum of the bond energy and the total excitation energy: Energy Threshold = Bond Energy + Total Excitation Energy $24.9 \mathrm{eV} = E_{\mathrm{C}} + 20.4 \mathrm{eV}$ Solving for the bond energy $E_{\mathrm{C}}$: $E_{\mathrm{C}} = 24.9 \mathrm{eV} - 20.4 \mathrm{eV}$ $E_{\mathrm{C}} = 4.5 \mathrm{eV}$ The bond energy of $\mathrm{H}_{2}$ is $4.5 \mathrm{eV}$. The value is already given to the first decimal place. Final Answer: The final answer is $\boxed{4.5}$
```

Figure 5: An example of selecting specific **segments** during the fine-tuning process

aging a reference model to compute excess loss, we identify and prioritize high-value tokens and reasoning steps, significantly improving training efficiency and model accuracy on CoT tasks. Among the proposed methods, both the token-selective loss and token-weighted approaches consistently outperformed baseline and alternative strategies, achieving substantial gains on both GSM8K and MATH benchmarks. The ablation study on the token-selection method further confirmed the critical role of curated reference data and adaptive selection strategies in optimizing reasoning performance.

7 Future Work

Building upon the findings of this study, there are several avenues for future research: One can evaluate the proposed selective fine-tuning methods on a broader range of reasoning tasks (e.g., logical puzzles, scientific reasoning, commonsense QA) and across different domains to assess the generalizability of the approach. Applying and analyzing the effectiveness of these selective techniques on larger, state-of-the-art LLMs to determine if the observed benefits scale with model size and capability. Further theoretical analysis is needed to deepen the understanding of why prioritizing high-excess-loss tokens specifically benefits complex reasoning tasks and how it influences the internal representations learned by the model. Finally, developing more sophisticated criteria for token or segment selection beyond excess loss, potentially incorporating measures of uncertainty, semantic importance, or structural role within the reasoning chain would be an interesting future direction. Exploring dynamic thresholding or adaptive weighting schemes based on training progress could also be beneficial.

8 Limitations

Despite the promising results, our approach has several limitations that warrant further investigation: The quality of our method depends heavily on the reference model's performance. If the reference model has biases or weaknesses in certain reasoning domains, these limitations may propagate to the selective fine-tuning process. Our method requires training an additional reference model and computing token-wise excess loss during fine-tuning, introducing computational overhead compared to standard fine-tuning approaches. Our evaluation primarily focused on mathematical reasoning tasks (GSM8K and MATH). The effectiveness of selective fine-tuning on other reasoning domains (e.g., logical reasoning, common sense reasoning) remains to be thoroughly explored. Also, the segment-selective approach relies on unsupervised segmentation of reasoning traces, which may not always align with the true logical structure of the solution. More sophisticated segmentation methods might further improve performance. finally, our experiments were conducted on the 1.5B-parameter Qwen2.5 model. The scalability and effectiveness of selective fine-tuning on larger models (10B+ parameters) remains an open question.

References

Johan Boye and Birger Moell. 2025. Large language models and mathematical reasoning failures.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. 2025. Reasoning beyond limits: Advances and open problems for llms.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G. Patil, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. Llms can easily learn to reason from demonstrations structure, not content, is what matters!

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2025. Rho-1: Not all tokens are what you need.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. 2025a. Selectit: Selective instruction tuning for llms via uncertainty-aware self-reflection.

Ziche Liu, Rui Ke, Yajiao Liu, Feng Jiang, and Haizhou Li. 2025b. Take the essence and discard the dross: A rethinking on data selection for fine-tuning large language models.

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. 2024. TSDS: Data selection for task-specific model finetuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models.

Yu Yang, Siddhartha Mishra, Jeffrey N Chiang, and Baharan Mirzasoleiman. 2024. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.