GTA: Supervised-Guided Reinforcement Learning for Text Classification with Large Language Models

Min Zeng, Jingfei Sun, Xueyou Luo, Caiquan Liu, Shiqi Zhang, Li Xie, Xiaoxin Chen
vivo AI Lab
zengmin.ai@vivo.com

Abstract

In natural language processing tasks, pure reinforcement learning (RL) fine-tuning methods often suffer from inefficient exploration and slow convergence; while supervised fine-tuning (SFT) methods, although efficient in training, have limited performance ceiling and less solid theoretical foundation compared to RL. To address efficiency-capability trade-off, we propose the **G**uess-**T**hink-**A**nswer (**GTA**) framework that combines the efficiency of SFT with the capability gains of RL in a unified training paradigm. GTA works by having the model first produce a provisional guess (optimized via cross-entropy loss), then reflect on this guess before generating the final answer, with RL rewards shaping both the final output and the format of the entire GTA structure. This hybrid approach achieves both faster convergence than pure RL and higher performance ceiling than pure SFT. To mitigate gradient conflicts between the two training signals, we employ loss masking and gradient constraints. Empirical results on four text classification benchmarks demonstrate that GTA substantially accelerates convergence while outperforming both standalone SFT and RL baselines.

1 Introduction

Text classification, as a foundational task in natural language processing (NLP), has been widely employed for sentiment analysis (Pang et al., 2002), intent recognition (Chen et al., 2016), and news categorization (Johnson and Zhang, 2015). Early NLP solutions primarily relied on rule-based systems and statistical models—including hidden Markov models (HMMs) and support vector machines (SVMs)—to learn patterns from annotated corpora (Joachims, 1998). The emergence of deep learning and Transformer (Vaswani et al., 2017) architectures dramatically enhanced classification performance, with Bidirectional Encoder Representations from Transformers (BERT)'s (Devlin, 2018) bidirectional pre-training paradigm capturing rich contextual representations and achieving breakthroughs across multiple tasks.

Large language models (LLMs)—such as GPT (Achiam et al., 2023), Llama (Grattafiori et al., 2024), Qwen (Yang et al., 2024), and DeepSeek (Guo et al., 2025)—have demonstrated remarkable capabilities across numerous NLP tasks, including text classification (Kostina et al., 2025). While SFT has been the predominant approach to adapt these models for specific tasks, it faces inherent limitations: SFT methods directly learn to produce correct answers without explicit reasoning, leading to limited generalization capabilities and performance ceilings. Chain-of-thought (CoT) prompting (Wei et al., 2022)—a technique that guides models to generate intermediate reasoning steps before producing final answers—has shown significant improvements across various reasoning tasks (Kojima et al., 2022; Xia et al., 2025). However, applying CoT within the SFT paradigm requires extensive human annotation of reasoning chains, resulting in substantial costs and susceptibility to annotator biases and quality inconsistencies (Tan et al., 2024; Byun et al., 2024).

RL offers a promising alternative that can theoretically overcome these limitations by combining the benefits of CoT reasoning with optimization-based learning (Xu et al., 2025; Wang et al., 2024). From reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) to advanced frameworks like group relative policy optimization (GRPO) (Shao et al., 2024), RL techniques can explore and optimize intermediate reasoning processes without requiring manually annotated reasoning chains. This approach holds particular promise for enhancing model performance beyond what SFT can achieve. However, the application of RL to text classification tasks remains challenging due to fundamental limitations: unlike supervised learning's direct approach, RL methods

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 1050–1060
November 4-9, 2025

must discover optimal reasoning strategies through self-guided exploration—a process often hampered by inefficient exploration, slow convergence, and potential training instability (Chen et al., 2025a). These efficiency challenges have hindered pure RL-based methods from consistently outperforming SFT approaches despite their stronger theoretical foundation.

To address these challenges, this work introduces a novel Guess–Think–Answer (GTA) framework that seamlessly integrates the advantages of SFT and RL within a unified single-stage training process. In our approach, the model first generates an intuitive guessed answer, then engages in a "think" step—reasoning explicitly over the guessed answer and the input question—before producing a refined final answer. Our main contributions can be summarized as follows:

1. We propose a novel GTA framework that structures the reasoning process into three distinct phases: an initial intuitive guess, an explicit reasoning step that reflects on this preliminary prediction, and a refined final answer that incorporates this reasoning.

2. We develop a unified training approach that seamlessly integrates SFT and RL within a single-stage process. Our method applies cross-entropy loss to the guessed answer while optimizing the reasoning process and final answer through RL-based rewards. To ensure effective cooperation between these learning paradigms, we introduce a specialized loss masking strategy and gradient cosine adjustment technique that mitigates potential gradient conflicts.

3. Our framework eliminates the need for manual annotation of reasoning chains by enabling the model to spontaneously learn effective reasoning patterns through reinforcement. Experimental results demonstrate that GTA significantly outperforms both standard SFT baselines and state-of-the-art RL methods across multiple text classification benchmarks.

2 Related Work

CoT prompting has emerged as a powerful technique to enhance the reasoning capabilities of LLMs by guiding them to generate intermediate reasoning steps before producing final answers (Wei et al., 2022; Kojima et al., 2022; Xia et al., 2025). While CoT significantly improves performance across various reasoning tasks, its application in text classification often requires extensive human annotation of reasoning chains, leading to substantial costs and quality inconsistencies (Tan et al., 2024; Byun et al., 2024). To address these limitations, researchers have explored RL approaches that can optimize model behavior without requiring manual annotation of intermediate steps (Xu et al., 2025; Wang et al., 2024).

Traditional RL methods in NLP, however, often suffer from inefficient exploration and slow convergence, making it challenging for pure RL-based methods to consistently outperform SFT approaches despite their stronger theoretical foundation (Chen et al., 2025a). Recent advancements like the GRPO algorithm (Shao et al., 2024) have improved RL efficiency by estimating baselines from group scores, thereby reducing computational costs. Additionally, when combining multiple learning objectives—such as SFT and RL—gradient conflicts can arise, leading to suboptimal convergence. Techniques such as gradient masking and cosine similarity adjustments have been developed to mitigate these conflicts by aligning gradients toward compatible directions (Yu et al., 2020; Chen et al., 2018), which inspires our approach to harmonizing supervised and RL signals within our proposed GTA framework.

3 Methodology

In this section, we present the proposed GTA framework in detail. We begin with an overview of the overall design, followed by the description of each component and training objective, highlighting how the method effectively integrates reinforcement learning with reasoning-oriented supervision.

3.1 Prompt Design

To accelerate convergence during RL training, we propose the GTA, which introduces a novel `Guess` stage to the conventional reasoning process. As illustrated in Figure 1, our prompt design guides the model to sequentially produce three components: `Guess`, `Think`, and `Answer`. The right side of the figure presents an example of a model-generated response adhering to this structure.

- **Guess**: In this initial stage, the model generates a preliminary answer based on intuition or prior knowledge. This guess serves as a reference point for subsequent reasoning and

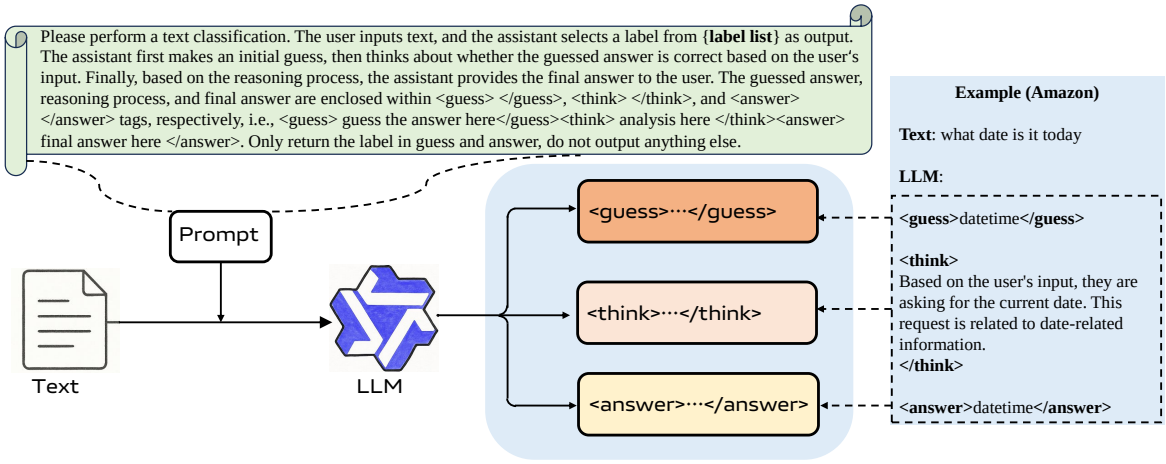


Figure 1: Overview of the Guess–Think–Answer framework.

the final answer. During training, the `Guess` component is supervised using cross-entropy loss.

- **Think**: Building upon the initial guess and the input question, the model produces a sequence of reasoning steps. This process aids in task comprehension and enhances the model's generalization capabilities.

- **Answer**: The final answer is generated by integrating insights from both the `Guess` and `Think` stages. This answer may align with or differ from the initial guess. The quality of the `Answer`, along with the overall output structure, is evaluated using a reward signal, which guides the RL component of the training.

By incorporating supervised signals in the `Guess` stage, our framework accelerates RL convergence and fosters the generation of interpretable reasoning processes.

3.2 Training Objective

We propose a unified training framework that combines SFT and RL within a single optimization process. As shown in Figure 2, our approach exploits the GTA output format to assign distinct training objectives to the model's various outputs.

SFT Loss. For the *Guess* segment, which represents the model's initial prediction based on intuition or prior knowledge, we employ a standard cross-entropy loss. To ensure that the loss computation focuses solely on this segment, we apply a masking strategy that assigns a special token (e.g., `-100`) to tokens outside the *Guess* span. Formally, the SFT loss is defined as:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t \in \mathcal{G}} \log P_\theta(y_t | y_{<t}, x) \tag{1}$$

where $\mathcal{G}$ denotes the set of token positions corresponding to the *Guess* segment, y_t is the target token at position t, $y_{<t}$ represents the sequence of preceding tokens, and x is the input text.

RL objective function. In optimizing the model's final output, we introduce the GRPO (Shao et al., 2024) algorithm with minor modifications. GRPO improves sample efficiency by generating multiple candidate outputs for the same input prompt and computing relative advantages without a separate value function, thereby reducing training resource consumption. In LLMs, reward signals can be categorized into model-based and rule-based rewards; text classification tasks are particularly amenable to rule-based rewards, which are assigned by directly comparing the model's final prediction to the ground-truth label. The reward definitions are as follows:

$$R_{\text{format}} = \begin{cases} 1, & \text{if format correct,} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

$$R_{\text{accuracy}} = \begin{cases} 1, & \text{if classification correct,} \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

$$R_{\text{total}} = R_{\text{format}} + R_{\text{accuracy}}. \tag{4}$$

where R_{format} denotes the format reward, which is granted whenever the model's output adheres to the prescribed GTA format. R_{accuracy} denotes the accuracy reward, which is awarded when the model's

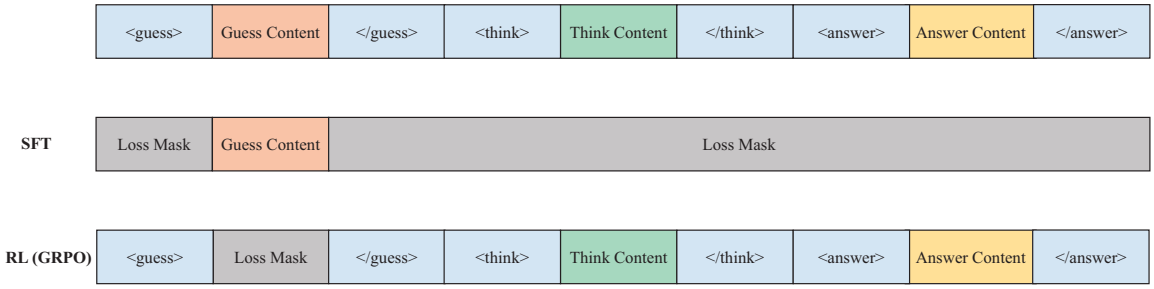


Figure 2: Illustration of the loss masking strategy applied during training.

final prediction matches the true label. R_{total} represents the overall reward signal. The overall RL training objective is defined as follows:

$$\mathcal{J}(\theta) = \mathbb{E}\big[q \sim P(Q),\ \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|q)\big] \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t \notin \mathcal{G}} \min \left(r_i(\theta), \text{clip}(r_i(\theta), 1-\epsilon,\ 1+\epsilon) \right) \hat{A}_{i,t} \right] - \beta\, \mathbb{D}_{\text{KL}}\big[\pi_\theta \| \pi_{\text{ref}}\big] \tag{5}$$

where G denotes the group size, representing the number of output sequences sampled in parallel for the same prompt. The i-th output sequence is denoted as o_i, where a loss mask is applied to the text within the *Guess* segment to selectively include tokens in the loss computation. $\hat{A}_{i,t} = (R_{\text{total},i} - \mu)/\sigma$ denotes the advantage function, obtained by subtracting the group's mean reward μ from each individual reward and then dividing by the group's reward standard deviation σ. The ratio $r_i(\theta) = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$ corresponds to the probability ratio between the new and old policies. The hyperparameter ϵ defines the clipping range for gradient updates, and β adjusts the weight of the Kullback–Leibler (KL) divergence term. Since our training involves backpropagating two distinct loss components, we jointly constrain the magnitude of model updates using both the clipping mechanism and the KL divergence term. While $\mathbb{D}_{\text{KL}}$ represents the KL divergence term and can be further expressed as:

$$\mathbb{D}_{\text{KL}}\big[\pi_\theta \,\big\|\, \pi_{\text{ref}}\big] = \frac{\pi_{\text{ref}}\big(o_{i,t} \,|\, q,\, o_{i,<t}\big)}{\pi_\theta\big(o_{i,t} \,|\, q,\, o_{i,<t}\big)} - \log \frac{\pi_{\text{ref}}\big(o_{i,t} \,|\, q,\, o_{i,<t}\big)}{\pi_\theta\big(o_{i,t} \,|\, q,\, o_{i,<t}\big)} - 1\,. \tag{6}$$

(7)

we adopt the same KL divergence term computation as in the original GRPO, but instead of using a static base model as the reference, we periodically update the reference model with the current model during training. This strategy prevents the model's updates from being constrained too closely to the base model, which could otherwise hinder performance improvements. Policy optimization algorithms maximize the objective function $\mathcal{J}(\theta)$ via gradient ascent, which is equivalent to finding the minimum of $-\mathcal{J}(\theta)$ by gradient descent; accordingly, the RL loss function can be expressed as follows:

$$\mathcal{L}_{\text{RL}} = -\mathcal{J}(\theta) \tag{8}$$

Total loss function. The total loss function is defined as the sum of two distinct loss components, and is computed as follows:

$$\mathcal{L}_{\text{Total}} = \lambda_1 \mathcal{L}_{\text{SFT}} + \lambda_2 \mathcal{L}_{\text{RL}}, \tag{9}$$

where λ_1 and λ_2 are two hyperparameters that balance the weights of the SFT and RL loss terms.

Loss Mask. During training, the masking strategy ensures that each loss component only affects its intended segment. When computing the SFT loss, tokens outside the *Guess* section are masked, enabling the language model to learn the correct labels via cross-entropy loss solely on the *Guess* portion. Conversely, during RL, tokens within the *Guess* section are masked in the loss calculation to prevent adverse learning signals from the guessed

labels. This approach effectively isolates the two loss computations, reducing gradient conflicts during backpropagation.

3.3 Gradient Conflict Detection and Resolution

Despite the loss masking mechanism described above and the assignment of distinct weights to each objective to reduce gradient conflicts, multi-task learning cannot fully avoid such interference. To further mitigate gradient conflicts, we analyze the gradients of the two losses during backpropagation and integrate theoretical insights from PCGrad (Yu et al., 2020) into the training process. As illustrated in Figure 3, we use cosine similarity to detect gradient conflicts: a positive cosine similarity between gradients from the two losses indicates no conflict during backpropagation, whereas a negative cosine similarity denotes the occurrence of a gradient conflict. The calculation formula of cosine similarity under the gradient can be expressed as follows:

$$\cos(\theta) = \frac{\nabla \mathcal{L}_{\text{SFT}} \cdot \nabla \mathcal{L}_{\text{RL}}}{\|\nabla \mathcal{L}_{\text{SFT}}\| \cdot \|\nabla \mathcal{L}_{\text{RL}}\|}. \quad (10)$$

where θ is the angle between the two gradient vectors, and ∇ denotes the gradient calculated by backpropagation under the corresponding loss. The final loss update rule is:

$$\mathcal{L}_{\text{Final}} = \begin{cases} \mathcal{L}_{\text{Total}}, & \text{if } \nabla \mathcal{L}_{\text{SFT}} \cdot \nabla \mathcal{L}_{\text{RL}} > 0, \\ \mathcal{L}_{\text{RL}}, & \text{otherwise}, \end{cases} \quad (11)$$

During parameter updates, when gradient conflicts arise, we mitigate such conflicts by retaining only the loss component associated with the RL objective.

4 Experimental Setup

This section outlines the experimental setup used to evaluate our approach. We describe the datasets, baseline models, and implementation details.

4.1 Datasets

Considering the resource constraints of our experimental process, we selected four datasets reflecting distinct classification scenarios based on recent studies (Chen et al., 2024; Menon and Srivastava, 2024; Chen et al., 2025b): SST-5 (Socher et al., 2013), Amazon (FitzGerald et al., 2022), Emotion (Saravia et al., 2018), and BBC News (Greene and Cunningham, 2006), each containing multiple categories. The detailed descriptions of these datasets are presented in Table 1; they cover four domains—movie reviews, intent recognition, English tweets, and News—with class counts ranging from five to eighteen.

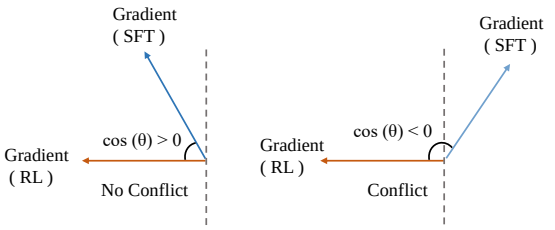


Figure 3: Illustration of gradient conflict analysis via gradient cosine similarity

4.2 Models

Below we provide a concise summary of LLMs used in our experiments:

Qwen2.5 (3B)[1] is Alibaba's open-source 3 billion parameters instruction-tuned LLM supporting long-context understanding (up to 128K tokens) and generation (up to 8K tokens), making it adept at handling extended dialogues and complex prompts. It features robust multilingual comprehension across 29 languages, ensuring broad applicability in diverse language settings. It excels in structured output generation (e.g., JSON) and instruction following.

Qwen3 (4B)[2] is Alibaba's latest open-source large language model with 4 billion parameters. Trained on a substantially larger corpus of 36 trillion tokens across 119 languages and dialects, it delivers robust performance across diverse reasoning and understanding tasks. It supports hybrid reasoning modes, seamlessly switching between CoT thinking and direct-response generation.

Llama3.2 (3B)[3] is a 3 billion parameter open-source model released by Meta. It is designed as a lightweight variant of the Llama 3 family, trained on a diverse multilingual corpus and optimized for efficiency on resource-constrained environments. Despite its relatively small size, Llama3.2 demonstrates competitive performance on a wide range of reasoning and classification tasks.

[1] https://huggingface.co/Qwen/Qwen2.5-3B-Instruct

[2] https://huggingface.co/Qwen/Qwen3-4B

[3] https://huggingface.co/meta-llama/Llama-3.2-3B

Dataset	Description	Training	Testing	Classes
SST-5	The Stanford Sentiment Treebank five-class benchmark is a standard corpus for fine-grained sentiment classification. It consists of sentences drawn from movie reviews, each annotated at the sentence level with one of five sentiment labels—very negative, negative, neutral, positive, and very positive.	8,544	2,210	5
Amazon	This is the English (en-US) subset of the Massive Scenario Classification task from the Massive Text Embedding Benchmark (MTEB), aimed at intent prediction in voice assistant interactions. The dataset covers 18 scenario classes (such as alarm, audio, iot, calendar, play, news, and weather).	11,514	2,974	18
Emotion	The emotion dataset is a carefully curated subset of English tweets annotated with six basic emotions—sadness, joy, love, anger, fear, and surprise—providing a standardized benchmark for evaluating emotion recognition models . Each sample consists of a tweet paired with its corresponding label.	16,000	2,000	6
BBC News	Dataset on BBC News Topic Classification published on the BBC News website corresponding during 2004-2005. Each article is labeled under one of 5 categories: business, entertainment, politics, sport or tech.	1,225	1,000	5

Table 1: Detailed description of datasets utilized in the experimental process. Each dataset differs in terms of the number of classes, training samples, and test samples.

4.3 Hyperparameters

All experiments were carried out on a multi-node cluster, each node hosting four NVIDIA L40s GPUs (48 GB each) and coordinated via DeepSpeed with ZeRO Stage 2. For SFT, we employed the open-source ModelScope Swift framework, while RL baselines used the TRL library's GRPO implementation and our proposed GTA built atop GRPO. Inputs were truncated to a maximum of 4,096 tokens, and models were trained in bfloat16 precision with a per-device batch size of 4. In the RL phase, each prompt generated 16 candidate answers, and we applied importance sampling with a reuse factor of 4, and included a KL penalty weighted by $\beta = 0.01$ to stabilize policy updates. In our GTA, losses are assigned equal weights of 1. Each dataset is trained for 3–4 epochs.

5 Results and Analysis

In this section, we report the main experimental results and provide in-depth analyses. We first compare GTA with baseline methods on multiple benchmarks, then investigate convergence behavior, reasoning robustness, and case studies to gain further insights into its effectiveness.

5.1 Performance

The experimental results presented in Table 2 evaluate the fine-tuning performance of three methods—SFT, GRPO, and GTA—across two model scales: Qwen2.5 (3B), Qwen3 (4B), and Llama3.2 (3B). The evaluation spans four classification benchmarks: SST-5, Amazon, Emotion, and BBC News. Performance metrics include accuracy and weighted F_1 scores. Accuracy reflects the proportion of correct predictions over the total number of instances. Weighted F_1 score computes the harmonic mean of precision and recall, weighted by the number of instances in each class. This metric handles class imbalance by assigning higher weights to classes with more instances.

Across all datasets and different models, GTA consistently outperforms both SFT and GRPO in terms of accuracy and weighted F_1 Score. Notably, on the Emotion dataset, GTA achieves an F_1 score of 92.47% with Qwen2.5, 92.94% with Qwen3, and 93.36% with Llama3.2 surpassing the other

Dataset	Base		SFT		GRPO		GTA	
	Acc	F_1	Acc	F_1	Acc	F_1	Acc	F_1
Qwen2.5 (3B)								
SST-5	11.76	13.34	60.72	59.59	58.60	57.05	**61.58**	**61.52**
Amazon	54.84	55.48	91.96	91.92	90.82	91.03	**92.47**	**92.46**
Emotion	58.75	58.63	91.35	91.41	82.50	81.54	**92.45**	**92.47**
BBC News	81.50	82.88	97.70	97.70	95.40	95.47	**98.50**	**98.50**
Qwen3 (4B)								
SST-5	45.88	39.80	61.67	60.87	59.28	58.70	**61.95**	**60.94**
Amazon	68.96	70.28	92.57	92.58	90.55	90.32	**92.87**	**92.92**
Emotion	51.15	54.00	92.20	92.09	84.55	84.23	**92.95**	**92.94**
BBC News	80.40	81.79	97.70	97.70	94.90	95.01	**97.90**	**97.91**
Llama3.2 (3B)								
SST-5	38.42	36.11	59.91	50.65	56.33	53.40	**61.18**	**60.13**
Amazon	19.60	18.79	91.69	91.52	84.13	83.01	**92.84**	**92.84**
Emotion	41.65	42.57	93.00	92.92	74.40	74.42	**93.30**	**93.36**
BBC News	34.30	25.15	97.40	97.41	91.30	91.49	**97.50**	**97.60**

Table 2: Fine-tuning performance (%) on four benchmarks using SFT, GRPO, and GTA across two model sizes (Base refers to the model without fine-tuning).

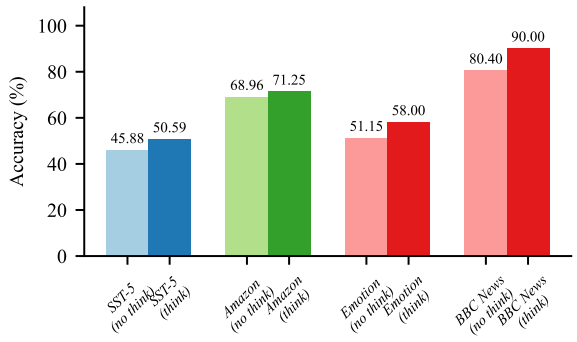


Figure 4: Accuracy comparison of Qwen3 (4B) *think* and *no think* across multiple datasets

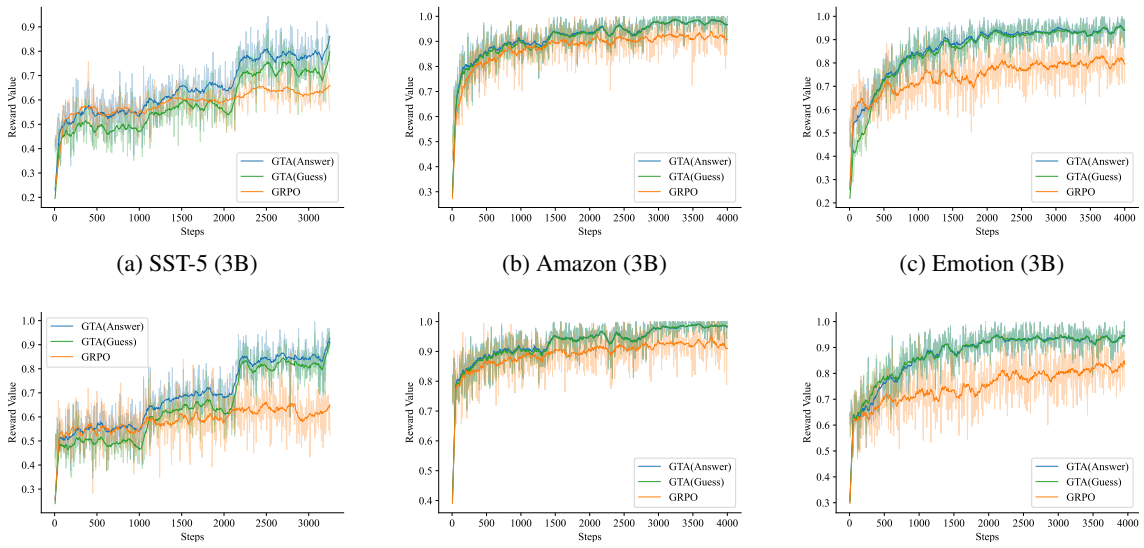
methods by a significant margin. Similarly, on the Amazon dataset, GTA attains the highest accuracy and F_1 scores, indicating its robustness across different domains. These results underscore the effectiveness of the proposed GTA method in enhancing model performance across diverse datasets and model scales. The consistent improvements in both accuracy and F_1 scores highlight GTA's robustness.

5.2 Exploring Performance Boundaries in Zero-Shot

We first investigated whether reasoning-enhanced prompting could improve classification performance without model fine-tuning, as this would establish important baselines and validate a core premise of our GTA framework: that explicit reasoning steps can elevate model capabilities. This exploration addresses a fundamental question in LLM deployment—whether performance limitations stem from model capabilities themselves or from suboptimal reasoning processes that can be enhanced through structured prompting. Using the base Qwen3 (4B) model with its native "think" mode toggle, we compared standard direct answering against explicit reasoning-then-answering across four benchmarks: SST-5, Amazon, Emotion, and BBC News. Figure 4 demonstrates consistent performance improvements across all datasets when reasoning steps are incorporated. The accuracy on SST-5 increases from 45.88% to 50.59%, Amazon review classification improves from 68.96% to 71.25%, Emotion classification improves from 51.15% to 58.00%, and BBC News classification shows the most significant gain from 80.40% to 90.00%. These results reveal that models operating in a deliberate reasoning mode possess substantially higher performance ceilings than those constrained to direct response generation.

5.3 Convergence Analysis of RL

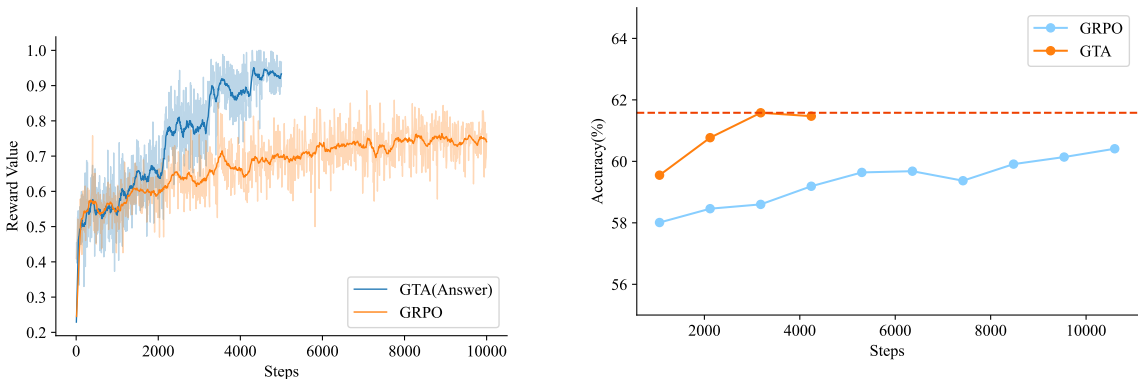
RL-based fine-tuning typically suffers from slow convergence due to lengthy exploration cycles. To assess how our GTA method addresses this limitation, we compared its convergence speed against



(a) SST-5 (3B) (b) Amazon (3B) (c) Emotion (3B)

(d) SST-5 (4B) (e) Amazon (4B) (f) Emotion (4B)

Figure 5: Variations in accuracy reward values on the SST-5, Amazon, and Emotion datasets during GTA and GRPO fine-tuning



(a) SST-5 (3B) Accuracy Reward (b) Accuracy (%) on the SST-5 (3B) Test Set

Figure 6: Accuracy reward and test set accuracy trends of GRPO and GTA on SST-5

the GRPO baseline across multiple datasets and model scales. Since BBC News is a relatively easy dataset with fewer categories and less training data, its convergence trends are less informative for analyzing RL training dynamics. We therefore focus our convergence analysis on SST-5, Amazon, and Emotion. Figure 5 illustrates the accuracy reward trajectories for both 3B and 4B models on SST-5, Emotion, and Amazon datasets, tracking three curves: GTA's answer segment (blue), GTA's guess segment (green), and GRPO (orange). On SST-5 and Emotion datasets, GTA demonstrates a substantial convergence advantage, with both guess and answer rewards consistently outperforming GRPO. While GRPO achieves faster initial convergence on the Amazon dataset, GTA maintains superior overall performance throughout training. These results confirm that the supervised guess segment effectively guides the answer segment, dramatically accelerating convergence while preserving the optimization benefits of RL—offering significant advantages for applying RL to language models in classification tasks.

To investigate whether GRPO might eventually match GTA's performance given sufficient training time, we extended our experiments on the SST-5 dataset to 10,000 steps (approximately 10 epochs). Figure 6 presents this extended analysis using the 3B-parameter model, with Figure 6(a) showing training reward progression and Figure 6(b) depicting test accuracy evolution. Even in this extended training scenario, GRPO's convergence remains

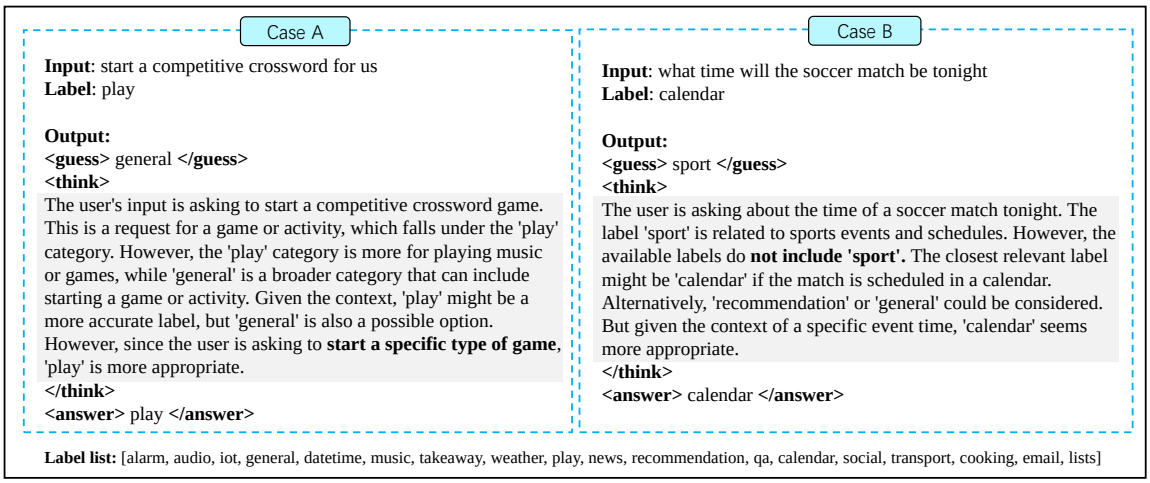


Figure 7: Illustration of GTA reasoning: from incorrect guess to correct answer

	GRPO	GTA (RL)	GTA (SFT+RL)
Acc	58.60	57.42	61.58
F_1	57.05	56.52	61.52

Table 3: Performance (%) of Qwen2.5 (3B) on GTA trained by RL on SST-5. GTA (RL) means that all stages are trained using RL, and GTA (SFT + RL) indicates that the loss of the guess is calculated through SFT.

substantially slower than GTA's. Despite showing gradual improvements in both reward and accuracy, GRPO fails to match GTA's performance even after 10,000 steps. These findings demonstrate that GTA not only accelerates training but also achieves a higher performance ceiling, offering significant practical advantages for deploying RL in language model fine-tuning for classification tasks.

5.4 Ablation on Guess

To validate the contribution of the supervised Guess stage, we replace the *Guess* component with RL updates. The results are shown in Table 3, We can observe that replacing supervised Guess with RL training does not improve the final accuracy. In fact, it yields a drop. This indicates the effectiveness of using supervised loss in the *Guess* stage.

5.5 Reasoning Process Analysis

We find that although making a guess under the guidance of supervision can accelerate convergence, the model does not blindly commit to the guessed answer as the final prediction. As show in Figure 7, when the model produces an incorrect guess, subsequent reasoning steps end to allow it to correct previous mistakes. In Case A, the model first predicts an incorrect label but gradually revises its reasoning and ultimately derives the correct answer. In Case B, the model not only outputs the correct final label but also explicitly indicates that the candidate label set does not include *"sport"*, thereby mitigating hallucination issues commonly observed in SFT. These examples, together with the overall accuracy improvements, highlight that GTA exhibits stronger robustness than purely SFT methods that only optimize the *Guess* segment.

6 Conclusion and Future Work

In this work, we present GTA, a novel training framework that addresses the efficiency-capability trade-off between SFT and RL by introducing a *Guess* stage to traditional CoT outputs. Under this framework, model outputs are organized into *Guess*, *Think*, and *Answer* segments, where the *Guess* is optimized via SFT while the overall format and final output are optimized through RL. To mitigate gradient conflicts, we apply loss masking and cosine-similarity constraints. Experimental results on four text-classification benchmarks show that GTA consistently outperforms pure SFT and GRPO baselines in accuracy and F_1 scores while achieving significantly faster convergence than pure RL approaches. Through analysis of training dynamics, we demonstrate that GTA successfully combines the efficiency of supervised learning with the performance gains of RL, substantially accelerating convergence and addressing the exploration inefficiency inherent to RL-based fine-tuning. Theoretically, our approach is not limited to text classification tasks, and we plan to explore extending GTA to broader NLP tasks in future work.

Limitations

While publicly available text classification datasets often contain noisy data, we did not perform preprocessing or sample high-quality subsets in this study. Additionally, due to resource constraints, we validated our proposed method only on 3B and 4B parameter models, without extending the evaluation to larger-scale models. Given that RL heavily relies on the underlying model's capacity, more powerful and generalizable models may yield greater benefits.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and Shyamal et al. Anadkat. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ju-Seung Byun, Jiyun Chun, Jihyung Kil, and Andrew Perrault. 2024. Ares: Alternating reinforcement learning and supervised fine-tuning for enhanced multi-modal chain-of-thought reasoning through diverse ai feedback. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4410–4430.

Junfan Chen, Richong Zhang, Junchi Chen, and Chunming Hu. 2024. Open-set semi-supervised text classification via adversarial disagreement maximization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2170–2180.

Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z Pan, Wen Zhang, Huajun Chen, and Fan et al. Yang. 2025a. Research: Learning to reason with search for llms via reinforcement learning. *arXiv preprint arXiv:2503.19470*.

Si-An Chen, Hsuan-Tien Lin, and Chih-Jen Lin. 2025b. Preserving zero-shot capability in supervised fine-tuning for multi-label text classification. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5699–5712.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6045–6049. IEEE.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, and Richa et al. Singh. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex et al. Vaughan. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, and Xiao et al. Bi. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2015*, pages 103–112. Association for Computational Linguistics (ACL).

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Arina Kostina, Marios D Dikaiakos, Dimosthenis Stefanidis, and George Pallis. 2025. Large language models for text classification: Case study and comprehensive review. *arXiv preprint arXiv:2501.08457*.

Rakesh R Menon and Shashank Srivastava. 2024. Discern: Decoding systematic errors in natural language for text classifiers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19565–19583.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, and Alex et al. Ray. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and Y et al. Wu. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Shuhe Wang, Shengyu Zhang, Jie Zhang, Runyi Hu, Xiaoya Li, Tianwei Zhang, Jiwei Li, Fei Wu, Guoyin Wang, and Eduard Hovy. 2024. Reinforcement learning enhanced llms: A survey. *arXiv preprint arXiv:2412.10400*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny et al. Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yu Xia, Rui Wang, Xu Liu, Mingyan Li, Tong Yu, Xiang Chen, Julian McAuley, and Shuai Li. 2025. Beyond chain-of-thought: A survey of chain-of-x paradigms for llms. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10795–10809.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, and Fanjin et al. Meng. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, and Haoran et al. Wei. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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