Sandwich Reasoning: An Answer-Reasoning-Answer Approach for Low-Latency Query Correction

Anonymous ACL submission

Abstract

Query correction is a critical entry point in modern search pipelines, demanding high accuracy strictly within real-time latency constraints. Chain-of-Thought (CoT) reasoning improves accuracy but incurs prohibitive latency for real-time query correction. A potential solution is to output an answer before reasoning to reduce latency; however, under autoregressive decoding, the early answer is independent of subsequent reasoning, preventing the model from leveraging its reasoning capability to improve accuracy. To address this issue, we propose Sandwich Reasoning (SandwichR), a novel approach that explicitly aligns a fast initial answer with post-hoc reasoning, enabling low-latency query correction without sacrificing reasoning-aware accuracy. SandwichR follows an "Answer–Reasoning–Answer" paradigm, producing an initial correction, an explicit reasoning process, and a final refined correction. To align the initial answer with post-reasoning insights, we design a consistency-aware reinforcement learning (RL) strategy: a dedicated consistency reward enforces alignment between the initial and final corrections, while margin-based rejection sampling prioritizes borderline samples where reasoning drives the most impactful corrective gains. Additionally, we construct a high-quality query correction dataset, addressing the lack of specialized benchmarks for complex query correction. Experimental results demonstrate that SandwichR achieves SOTA accuracy comparable to standard CoT while delivering a 40–70% latency reduction, resolving the latency-accuracy trade-off in online search.

1 Introduction

Query correction (Ye et al., 2023; Pande et al., 2022; Zhang et al., 2025b) serves as the first line of defense in modern Information Retrieval (IR) systems. User queries often contain various noise, such as phonetic errors, typos, and semantic ambiguities, which can significantly degrade retrieval relevance. While Large Language Models (LLMs) have demonstrated remarkable capabilities in text processing, deploying them for real-time query correction faces a critical dilemma: the trade-off between accuracy and inference latency.

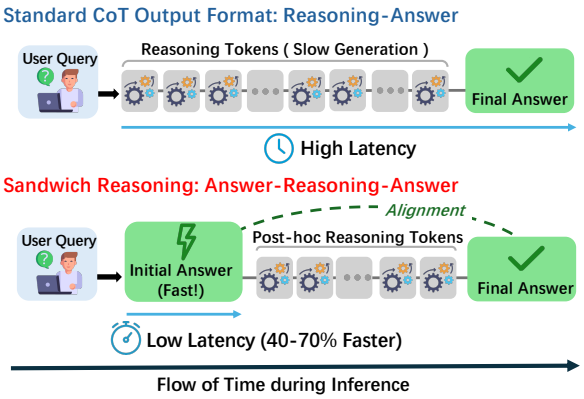

Figure 1: Comparison of reasoning paradigms: traditional Chain-of-Thought (CoT) reasoning vs the proposed sandwich reasoning in this paper.

Chain-of-Thought (CoT) reasoning has demonstrated its effectiveness across a wide range of tasks (Wang and Zhou, 2024; Han et al., 2024; Wang et al., 2023), it can enhance query correction accuracy by allowing the model to "think" before it outputs the corrected query. However, this *reasoning-first* paradigm incurs high computational costs and unacceptable latency for online search scenarios. An intuitive solution is to reverse the CoT order: generate an answer first, then reason about it (Dong et al., 2025). This promises the efficiency of a direct response. However, within a standard autoregressive model, this simple *answering-first* approach suffers from a decoupling problem: the initial answer is generated in isolation, blind to the reasoning that follows, thus gaining no actual benefit from it.

To bridge this gap, we introduce a novel

* Equal contribution.

answering-first approach for query correction, **Sandwich Reasoning**, which we refer to as **SandwichR**. Unlike standard CoT, as shown in Figure 1, our SandwichR outputs a sequence in a "Answer-Reasoning-Answer" format: an initial correction, followed by a reasoning trajectory, and a final correction. This structure allows the downstream search engine to utilize the initial correction for low-latency retrieval.

To ensure the first-generated query correction correlates with the subsequent reasoning process, we propose a consistency-aware strategy during training. Specifically, we design a specialized reward function that encourages consistency between the initial correction and the final correction. Through reinforcement learning (RL), we effectively distill the model's own reasoning capabilities back into its immediate intuition, enabling the initial correction to achieve accuracy comparable to CoT even without explicit reasoning steps during inference. Due to the lack of public benchmarks, we also construct a high-quality dataset based for query correction. Furthermore, to stabilize RL training, we employ a rejection sampling strategy, filtering for samples where the model shows potential for self-correction. Our contributions are as follows:

- We propose SandwichR, a "Answer-Reasoning-Answer" framework that resolves the dilemma between accuracy and latency by decoupling the reasoning process from the initial response for latency.
- We design a consistency-aware RL strategy with margin-based rejection sampling. This approach aligns the model's fast intuition with its slow reasoning, effectively distilling CoT capabilities into the initial answer and theoretically ensuring the initial answer's accuracy matches that of standard CoT approach's the reasoning-enhanced answer.
- We construct a realistic query correction benchmark based on a retrieval dataset and demonstrate that our method achieves SOTA performance while delivering a remarkable 40–70% speedup over the standard CoT approach, balancing high correction precision with the low latency required for real-time search.

2 Related Work

Query Correction (QC) is cruial in search engine pipelines (Ye et al., 2023; Gao et al., 2010), directly influencing retrieval recall and user satisfaction. Early approaches treat QC as a sequence-to-sequence translation task, evolving from statistical language models to Pre-trained Language Models (PLMs) such as BART (Shao et al., 2024a), which map noisy queries to their corrected forms based on contextual likelihood. With the emergence of Large Language Models (LLMs), recent studies have explored leveraging the extensive world knowledge of LLMs for correction via few-shot prompting (Davis et al., 2024; Li et al., 2023) or supervised fine-tuning (Fan et al., 2023). While LLMs demonstrate superior semantic understanding compared to smaller models, they often suffer from over-correction—erroneously altering correct named entities or shifting the user's original intent. Crucially, most existing works treat correction as an immediate generation task with few exploring explicit reasoning mechanisms to QC, which limits the model's robustness when facing ambiguous or complex errors.

Reasoning Large Language Models (LLMs) have advanced significantly via Chain-of-Thought (CoT) reasoning, improving performance in complex domains such as mathematics and logic (Guo et al., 2025; Jaech et al., 2024). To further optimize reasoning capabilities, researchers have focused on both data-centric approaches—selecting high-quality reasoning trajectories (Ye et al., 2025)—and algorithmic innovations, such as designing granular process rewards or employing RL to align model behaviors (Aggarwal and Welleck, 2025; Han et al., 2024). However, the prevailing paradigm in these studies follows a *reasoning-first* structure, where the rationale (e.g., wrapped in <think> tags) strictly precedes the final answer. While beneficial for accuracy, this sequential dependency imposes a severe penalty on inference latency which is operationally unacceptable in real-time scenarios like query correction. This creates a critical dilemma: standard CoT is too slow for search, while direct generation lacks the depth for complex correction. Therefore, this paper explores the SandwichR method, which strikes a balance between efficiency and accuracy, to better complete query correction.

3 Problem Formulation and Data Construction

3.1 Problem Formulation

In this paper, we focus on the task of Query Correction (QC). Formally, we are given an annotated

dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where N denotes the sample size. For each sample in the dataset, x_i denotes the original user query (requiring correction), and y_i represents the corresponding correct query (ground truth). While traditional QC approaches treat this as a direct generation task—mapping the original query directly to the corrected query—we aim to endow Large Language Models (LLMs) with explicit reasoning capabilities. Specifically, our objective is to generate the corrected query y from the input x by leveraging an intermediate reasoning process R.

3.2 Data Construction

Since there is no large-scale, open-source dataset specifically for complex query correction, we construct a dataset based on Multi-CPR (Long et al., 2022). We simulate real-world search errors by injecting noise into the original queries (Q_{clean}) to generate corrupted queries (Q_{noise}). Specifically, we introduce three representative types of query errors:

- **Wrong Words:** Randomly substituting a word with a visually or phonetically similar, or commonly confused character to simulate spelling errors, phonetic or visual confusions.
- **Missing Words:** Omitting a word (e.g., a function or content word) to simulate accidental deletion.
- **Disorder Words:** Randomly swapping adjacent words to simulate word order errors in hurried typing or others.

Diagrams for data construction and examples of three error types are illustrated in Figure 2. In this work, we limit each query to contain only one error. This design is based on the fact that the queries in the original dataset are relatively short, and in real-world scenarios, the error rate within such short queries is typically low, thereby making the constructed data more representative of actual search scenarios. Nevertheless, our data construction framework is flexible: higher error ratios and more complex error patterns can be readily generated by repeatedly applying or combining these three basic error types.

The final dataset consists of pairs $(Q_{\text{noise}}, Q_{\text{clean}})$, also referred to as (x, y) pairs.

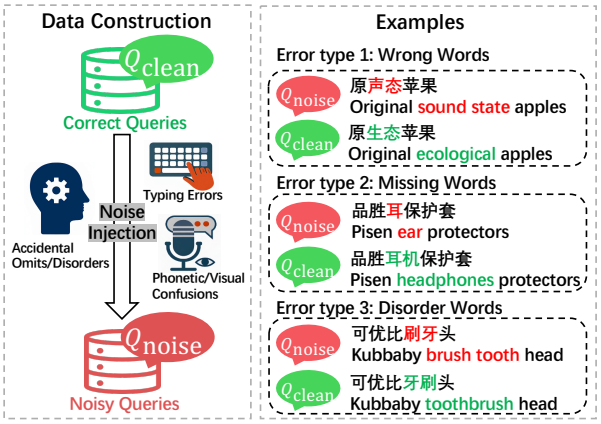

Figure 2: Examples of three types of query errors including wrong words, missing words, disorder words.

4 The Proposed Approach: SandwichR

In this section, we first present the overall architecture of SandwichR and then detail its two-stage training pipeline: SandwichR Ability Acquisition via SFT and Consistency-Aware Reinforcement Learning.

4.1 Approach Overview

Let x denote the input noisy query. We define the model's output y as a SandwichR-structured sequence:

$$y = [C_{\text{init}},\ R,\ C_{\text{final}}], \tag{1}$$

where C_{init} is the initial correction, R represents the correction reasoning process (i.e., reasoning trajectory), and C_{final} is the final correction derived from the reasoning. This structure allows the model to return C_{init} immediately to the user, satisfying the low-latency requirement of search engines.

Figure 3 illustrates the overall training of SandwichR. The training of SandwichR is conducted in two stages: (1) SandwichR Ability Acquisition via Supervised Fine-Tuning (SFT), which teaches the model to produce the SandwichR-structured output; and (2) Consistency-Aware Reinforcement Learning, which refines the model by explicitly reinforcing the alignment between the initial correction C_{init} and the reasoning process R, ensuring the first answer benefits from the subsequent reasoning for higher accuracy.

4.2 SandwichR Ability Acquisition via SFT

We first adapt a base LLM to the SandwichR format via Supervised Fine-Tuning. We utilize GPT-4o to generate high-quality reasoning trajectories and corrections.

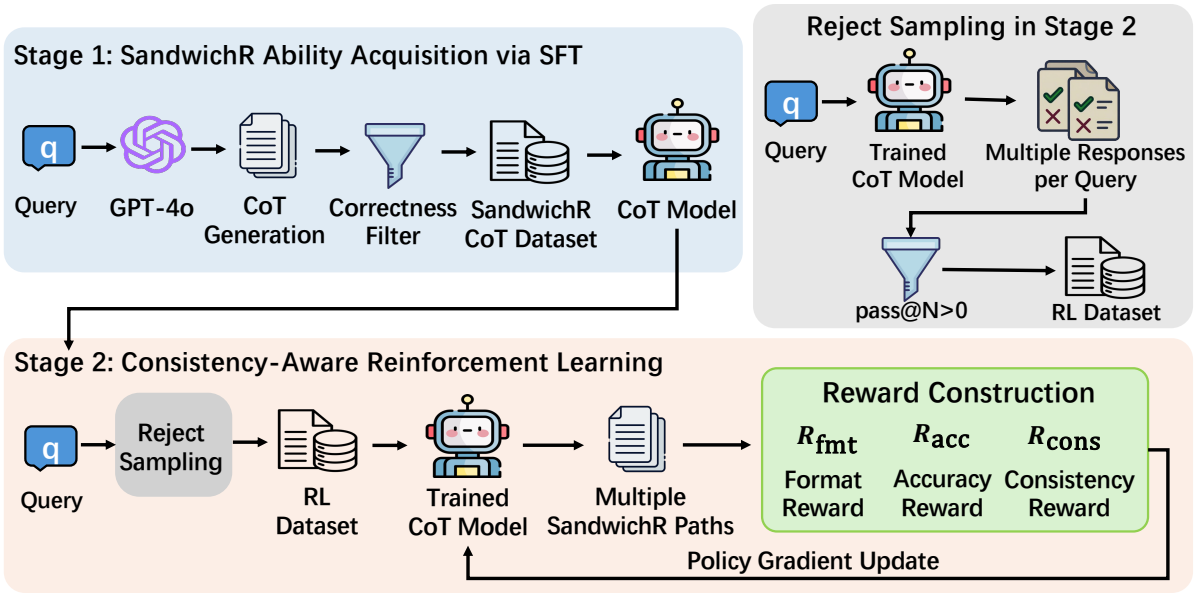

Figure 3: The overall workflow of the proposed SandwichR. It consists of two stages: (1) **SandwichR Ability Acquisition** via Supervised Fine-Tuning (SFT), and (2) **Consistency-Aware** Reinforcement Learning (RL).

Specifically, we employ the **Reasoning-Answer** format (as shown in Appendix A.1) to prompt GPT-4o, which instructs it to first generate its internal reasoning within <reasoning> tags, followed by the final corrected output within <answer> tags. This format aligns with the standard cognitive process of reasoning before answering, and thus ensures high-quality reasoning trajectories.

We then post-process these generations to restructure them into the SandwichR-structured sequence as Eq. (1). This restructured data is subsequently used for fine-tuning the base model to acquire the Sandwich Reasoning abilities.

4.3 Consistency-Aware Reinforcement Learning

SFT alone fails to align C_{init} with the reasoning-enhanced C_{final}. We employ GRPO (Shao et al., 2024b) to further optimize the model.

4.3.1 Margin-based Rejection Sampling

Training on random samples is inefficient. We observe that RL works best when learning from "borderline" cases. We define "borderline cases" as inputs where the model exhibits inconsistent performance across multiple attempts—sometimes yielding an acceptable correction and sometimes not. These cases reside at the performance margin where the model demonstrates nascent capability but requires refinement to consistently generate correct outputs.

Thus to select these high-value training instances, we implement a margin-based rejection sampling strategy. Specifically, for a given input, we sample N independent reasoning trajectories from the CoT-finetuned model from stage 1. A trajectory is deemed acceptable if the corresponding correction answer achieves an $F_{0.5}$-score > 0 in practice. The input is added to the RL training dataset only if at least one of the N sampled trajectories is acceptable (pass@N > 0); otherwise, it is rejected.

4.3.2 Reward Design

The reward function R_{total} is the key driver of our approach, composed of three terms:

$$R_{\text{total}} = w_{\text{acc}} \cdot R_{\text{acc}} + w_{\text{fmt}} \cdot R_{\text{fmt}} + w_{\text{cons}} \cdot R_{\text{cons}}$$

- Accuracy Reward (R_{acc}): Given the high precision requirement of query correction (avoiding over-correction), we use the $F_{0.5}$ score between C_{init} and the ground truth, rather than simple accuracy or F1 score. In practice, to encourage meaningful corrections, R_{acc} is set to 0 if C_{init} is identical to the original query Q_{noise}, and to the $F_{0.5}$ score otherwise.

$$R_{\text{acc}} = \begin{cases} 0, \text{if } C_{\text{init}} = Q_{\text{noise}}, \\ F_{0.5}(C_{\text{init}}, Q_{\text{noise}}, Q_{\text{clean}}), \text{otherwise.} \end{cases}$$

- Format Reward (R_{fmt}): A binary reward that penalizes the model if it fails to follow the

strict [Correct -> Reason -> Correct] structure, ensuring parsability.

- Consistency Reward (R_{cons}): To force the model to internalize the reasoning into the first step, we reward the identity between the initial and final output:

$$R_{\text{cons}} = \mathbb{I}(C_{\text{init}} = C_{\text{final}}),$$

where $\mathbb{I}(\cdot)$ is the indicator function (1 if the condition holds, 0 otherwise). This reward is crucial: it penalizes "blind guessing" (where C_{init} is right by luck but C_{final} changes it) and "disconnect" (where reasoning fixes the error in C_{final} but C_{init} remains wrong). It drives the policy to maximize $P(C_{\text{init}} = \text{Ground Truth})$ by leveraging the gradients back-propagated from the reasoning process. In practice, the format and consistency requirements are jointly enforced by computing $R_{\text{fmt}} \times R_{\text{cons}}$ as a unified term. This yields a reward of 1 only when the output adheres to the required structure and C_{init} matches C_{final}.

4.4 Discussion

A core advantage of our SandwichR is its ability to maintain the accuracy of the Reasoning-Answer (Rea-Ans) paradigm while achieving low-latency inference. Importantly, this is an achievement that the direct Answer-Reasoning (Ans-Rea) approach cannot achieve. This alignment in SandwichR stems from the structured dependency between the initial answer, reasoning process, and final answer enforced by SandwichR, which mirrors the core logic of Rea-Ans, where correct answers are inherently guided by explicit reasoning. Mathematically, the correctness of the final answer $C_{\text{Rea-Ans}}$ in the Rea-Ans paradigm can be modeled as a marginalization over all possible reasoning trajectories:

$$\begin{aligned} &P_{\text{Rea-Ans}}(C_{\text{Rea-Ans}} = y^* \mid x) \qquad (2)\\ &= \int P(R \mid x) \cdot P(C_{\text{Rea-Ans}} = y^* \mid x, R)\, \mathrm{d}R, \end{aligned}$$

where y^* denotes the ground truth, R represents the reasoning process, and x is the input noisy query. In contrast, Ans-Rea suffers from a fundamental decoupling: the correctness of its initial answer $C_{\text{Ans-Rea}}$ is independent of subsequent reasoning, i.e., $P_{\text{Ans-Rea}}(C_{\text{Ans-Rea}} = y^* \mid x) \perp P(R \mid x)$, leaving the initial answer unable to benefit from the model's reasoning capabilities.

SandwichR resolves this gap through its "Answer-Reasoning-Answer" structure and the consistency constraint $C_{\text{init}} = C_{\text{final}}$, which binds the initial answer to the reasoning-augmented final answer. Its joint probability distribution is defined as:

$$\begin{aligned} &P_{\text{SandwichR}}(C_{\text{init}} = y^*, R, C_{\text{final}} \mid x)\\ &= P(R \mid x) \cdot P(C_{\text{final}} = y^* \mid x, R) \cdot \mathbb{I}(C_{\text{init}} = C_{\text{final}}). \end{aligned}$$

Marginalizing over R and C_{final} yields the correctness probability of SandwichR's initial answer:

$$\begin{aligned} &P_{\text{SandwichR}}(C_{\text{init}} = y^* \mid x)\\ &= \int P(R \mid x) \cdot P(C_{\text{final}} = y^* \mid x, R)\, \mathrm{d}R, \end{aligned}$$

which is mathematically equivalent to the correctness probability of the Rea-Ans paradigm in Eq. (2). This equivalence demonstrates that SandwichR's initial answer inherits the reasoning-guided accuracy of Rea-Ans, as the consistency constraint effectively distills the information from R into C_{init}.

5 Experiments

This section presents our experimental setup and comprehensive results, covering datasets, evaluation metrics, baseline models, implementation details, and analysis of key findings.

5.1 Experimental Settings

5.1.1 Dataset and Metric

We conducted experiments on three error correction datasets constructed based on Multi-CPR-QC, namely E-commerce, Video, and Medical. These datasets contain queries from different domains, and all original queries were collected from real search engine systems within Alibaba Group (Long et al., 2022). The statistical analysis of these three datasets is presented in Table 1. Following the commonly used evaluation metrics in the error correction field (Zhang et al., 2025b; Xu et al., 2022), we adopted $F_{0.5}$-score and Accuracy (Acc) to evaluate the model's error correction performance.

5.1.2 Baselines

For the baselines, we selected models from different domains for testing, including **mT5** (Xue et al., 2021) and **BART** (Lewis et al., 2019)—two representative models in the traditional error correction field. We adopted three Chain-of-Thought (CoT) prompting strategies (see Appendix A.1 for their

Dataset	#Train	#Dev	#Test	Tra_Len	Dev_Len	Tes_Len
E-commerce	90,511	1,000	1,000	6.43	6.39	6.45
Video	88,736	1,000	1,000	7.09	7.14	7.12
Medical	94,176	1,000	1,000	16.08	16.19	16.34

Table 1: Statistics of our LexNum. **#Train**, **#Dev**, **#Test** denote the number of the train, development and test datasets, while **Tra_Len**, **Dev_Len** and **Tes_Len** represent their average query lengths, respectively.

full prompts): **Ans-Rea**: The model follows an Answer-Reasoning format, presenting the correction outcome first, followed by the reasoning process. **Rea-Ans**: The model follows a Reasoning-Answer format, providing the reasoning process prior to the final correction result. **SandwichR**: The model uses an Answer-Reasoning-Answer format for output. Additionally, we implemented two optimization methods: **x-SFT**: Enhancing the model's correction capability via supervised fine-tuning. **x-RL**: Further boosting performance by integrating reinforcement learning. **GrammarGPT-7B** (Fan et al., 2023): A grammar error correction LLM that has been SFT on a grammar correction dataset. We also evaluated larger-scale models, including DeepSeek-R1-Distill-Qwen-7B (**Deepseek-R1-7B**), DeepSeek-R1-Distill-Qwen-32B (**Deepseek-R1-32B**), **QwQ-32B-Preview** (Team, 2024),**GPT-4o-mini**, **GPT-4o**, and **DeepSeek-R1**.

5.1.3 Implementation Details

We adopt Qwen2.5-1.5B-Instruct as our base LLM. This relatively small-scale model is chosen for its suitability to query correction where low latency and deployment cost are critical.

All our training and testing were conducted on NVIDIA A100 40GB GPUs. During SFT training, we performed LoRA (Hu et al.) fine-tuning. The LoRA hyperparameters were set to r=8 and alpha=16. Training was performed on a curated dataset of 1,000 samples with a batch size of 32 and a learning rate of 5e-5. During GRPO training, we conducted full-parameter fine-tuning which utilized a dataset of 200 samples with a batch size of 8. We employed a learning rate of 1e-5 and 20 epochs. The maximum completion length was 256 tokens. More experimental details can be found in Appendix A.2 and `https://anonymous.4open.science/r/SandwichR-C612`.

5.2 Main Results

We evaluate the performance of our proposed method, SandwichR, against various baselines across three query correction datasets. The main results are presented in Table 2. Based on the experimental outcomes, we draw the following conclusions:

SandwichR achieves superior performance. As shown in Table 2, our complete method SandwichR, refined through both SFT and RL, consistently achieves the best correction performance across all three datasets among models of similar scale, demonstrating the effectiveness of our approach. It outperforms all traditional correction models and even surpasses some untrained, larger-scale models (e.g., DeepSeek-R1-32B) in correction performance, which underscores the crucial role of our tailored two-stage training. Traditional Seq2Seq models exhibit lower performance due to their lack of the deep semantic understanding and reasoning capabilities inherent in LLMs. Some of the untrained large-scale LLMs, despite their vast general knowledge, show suboptimal performance due to the lack of task-specific adaptation. While some larger-scale LLMs (e.g., GPT-4o) achieve higher accuracy, they come with prohibitive computational cost and latency, making them impractical for real-time search. SandwichR offers the optimal balance for deployment-sensitive scenarios.

Effectiveness of the SandwichR Architecture. To isolate the effect of the architecture, we compare models built on the same base LLM with different reasoning formats. Post-training analysis reveals that the SandwichR strategy consistently outperforms both Rea-Ans and Ans-Rea structures across all datasets, confirming the validity of this structural design. By leveraging the autoregressive nature of LLMs, this structure ensures that the reasoning process explicitly informs and refines the final result (C_{final}).Crucially, the enforced consistency between the initial (C_{init}) and final results allows us to achieve high efficiency (by utilizing C_{init} for inference) without compromising quality. The SandwichR structure effectively guides the model to internalize the reasoning process required for complex query correction.

RL further enhances model performance. While Supervised Fine-Tuning (SFT) improves performance to a certain extent, we observe that Reinforcement Learning (RL) consistently outperforms SFT across all three datasets. This indicates that RL can further optimize the model's reasoning preferences and alignment following SFT. It is particularly effective for training LLMs to possess robust reasoning capabilities for query correction tasks.

Table 2: Performance comparison between SandwichR and the baseline on the three datasets, with the best performance among the trained LLMs is highlighted in bold, and the second-best method is underlined.

Category	Models	E-commerce		Video		Medical	
		$F_{0.5}$	Acc	$F_{0.5}$	Acc	$F_{0.5}$	Acc
Traditional Model	mT5 SFT	0.09	0.079	0.198	0.165	0.166	0.117
	BART SFT	0.150	0.124	0.303	0.27	0.377	0.35
Trained Base LLM (1.5B)	Rea-Ans SFT	0.178	0.164	0.278	0.255	0.347	0.309
	Ans-Rea SFT	0.199	0.181	0.280	0.250	0.366	0.312
	SandwichR SFT	0.202	0.187	0.293	0.253	0.376	0.338
	Ans-Rea SFT+RL	0.211	0.200	0.316	0.292	<u>0.392</u>	0.363
	Rea-Ans SFT+RL	<u>0.216</u>	<u>0.207</u>	<u>0.318</u>	<u>0.301</u>	0.387	<u>0.364</u>
	SandwichR SFT+RL (Ours)	**0.221**	**0.213**	**0.325**	**0.307**	**0.396**	**0.375**
Large-scale LLM	GrammarGPT-7B	0.045	0.037	0.092	0.081	0.161	0.148
	Deepseek-R1-7B	0.071	0.053	0.095	0.064	0.140	0.072
	Deepseek-R1-32B	0.212	0.185	0.249	0.203	0.350	0.261
	GPT-4o-mini	0.227	0.199	0.296	0.264	0.392	0.316
	QwQ-32B-Preview	0.295	0.214	0.329	0.256	0.427	0.301
	GPT-4o	0.299	0.259	0.354	0.310	0.475	0.384
	Deepseek-R1	0.333	0.244	0.371	0.284	0.452	0.321

Table 3: Robustness under Strict Token Budgets. Time in seconds per sample. Full budget allows 256 tokens; Limited budget simulates low-latency constraints. Δ indicates the change from Full to Limited Budget.

Method	Setting	E-commerce		Video		Medical	
		Acc	Time (s)	Acc	Time (s)	Acc	Time (s)
Rea-Ans	Full Budget	0.207	1.959	0.301	1.143	0.364	1.550
	Limited Budget	0.000	0.457	0.000	0.484	0.009	0.900
	Δ (%)	**-100.00**	-76.67	**-100.00**	-57.66	**-97.53**	-41.94
Ans-Rea	Full Budget	0.200	1.487	0.292	2.182	0.363	1.089
	Limited Budget	0.200	0.464	0.292	0.446	0.359	0.893
	Δ (%)	**0.00**	-68.80	**0.00**	-79.56	**-1.10**	-18.00
SandwichR	Full Budget	0.213	1.613	0.307	1.174	0.375	1.683
	Limited Budget	0.213	0.467	0.307	0.474	0.374	0.924
	Δ (%)	**0.00**	-71.05	**0.00**	-59.63	**-0.27**	-45.10

5.3 Efficiency Analysis

To evaluate the practical efficiency of different reasoning formats, we conducted a comparative analysis of three reasoning formats: Rea-Ans, Ans-Rea, and SandwichR—under two distinct token budget conditions: a sufficient budget (256 tokens) and a strictly limited budget (20 tokens for E-commerce and Video queries; 40 tokens for the longer Medical queries) to simulate real-time, resource-constrained deployment scenarios with strict low-latency requirements. The results, including inference latency and accuracy, are presented in Table 3.

Under token limits, both Ans-Rea and SandwichR significantly reduce inference time with only a minor accuracy drop. SandwichR stands out by maintaining the highest accuracy alongside low latency. In contrast, the widely-adopted standard CoT mode, Rea-Ans, often fails to output a final answer before reaching the token limit, as its initial reasoning consumes available tokens. SandwichR overcomes this limitation and achieves 40%–70% faster inference than Rea-Ans while maintaining comparable accuracy. These results confirm that SandwichR achieves low latency without compromising accuracy, validating its advantage for practical, resource-constrained deployment.

5.4 Analysis of Different Error Types

We analyze model performances on three error types: Wrong Words, Missing Words, and Disorder words as illustrated in Table 4. The overall average accuracy ranks these tasks by difficulty, with Disorder words being the easiest, followed by Wrong Words, and Missing Words the most challenging. SandwichR shows consistent advantages: it achieves the best accuracy for the hardest task Missing Words on both Medical and E-commerce datasets, for Wrong Words on Medical, and for

Table 4: Analysis of accuracy of Different Error Types for SFT+RL Models on Three Datasets.Best performance among each error type and dataset is highlighted in bold.

Dataset	Model	Wrong Words	Missing Words	Disorder words
Medical	Ans-Rea	0.3904	0.3003	0.3982
	Rea-Ans	0.3694	0.3003	**0.4222**
	SandwichR	**0.3934**	**0.3153**	0.4162
E-commerce	Ans-Rea	0.1862	0.1317	0.2823
	Rea-Ans	0.1532	0.1407	**0.3273**
	SandwichR	**0.1892**	**0.1467**	0.3033
Video	Ans-Rea	0.2733	0.2312	0.3713
	Rea-Ans	0.2853	**0.2492**	0.3683
	SandwichR	**0.2913**	0.2462	**0.3832**
Average	–	0.2813	0.2291	0.3636

Table 5: Ablation Study on Training Components

Models	E-commerce		Video		Medical	
	$F_{0.5}$	Acc	$F_{0.5}$	Acc	$F_{0.5}$	Acc
SandwichR SFT+RL	0.221	0.213	0.325	0.307	0.396	0.375
w/o Reject Sampling	0.217	0.196	0.313	0.293	0.383	0.351
SandwichR SFT	0.202	0.187	0.293	0.253	0.376	0.338

Disorder words on Video. It outperforms Ans-Rea in all 9 (Dataset × Error Type) categories and surpasses Rea-Ans in 6 out of 9. This robust and comprehensive performance across diverse errors highlights the effectiveness of the Sandwich Reasoning paradigm in providing reliable correction.

We also conducted multi-task merged training (details in Appendix A.3), which yielded better performance by enabling the model to learn generalizable features from cross-task data.

5.5 Ablation Study

The ablation study as illustrated in Table 5 validates the contribution of each training component. Applying GRPO on top of CoT-finetuned provides a consistent performance gain across all datasets. Furthermore, removing the Reject Sampling strategy leads to a noticeable drop in accuracy, which confirms that our data selection strategy is crucial for identifying and learning from valuable "borderline" cases during RL training.

5.6 Data Efficiency of RL Training

To verify the data efficiency of the RL training stage, we scaled the training data on the E-commerce dataset. As shown in Figure 4, increasing the data size from 200 to 500 and even 1000 samples did not lead to significant performance gains. This indicates that RL training combined with rejection sampling is highly data-efficient, allowing the model to converge quickly by learning from a small set of high-quality "borderline" examples. Consequently, we selected the configuration of 200 samples as it offers an optimal balance between performance and computational cost.

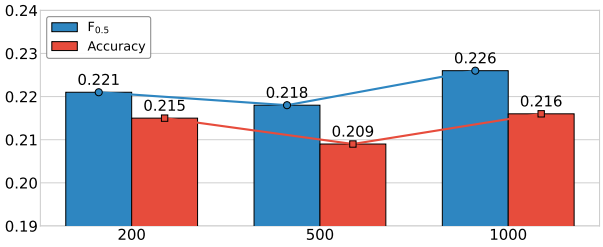

Figure 4: Performance Comparison of SandwichR with different RL training data sizes on E-commerce Dataset. The X-axis represents the RL training data size.

6 Conclusion

In this paper, we tackle the core accuracy-latency dilemma in real-world query correction by proposing Sandwich Reasoning (SandwichR), a novel "Answer–Reasoning–Answer" generation framework. It evolves from the conventional robust yet high-latency Reasoning–Answer paradigm and its low-latency but less robust variant Answer–Reasoning. By front-loading the answer while aligning it with subsequent reasoning, SandwichR ensures the first answer both fast and accurate, as it can benefit from the model's reasoning capability. We achieve this through the consistency-aware reinforcement learning strategy, which employs a consistency reward that forces the initial answer to align with the reasoning-enhanced final answer, thereby transforming the initial answer from a blind guess into a pre-emptive output that incorporates reasoning benefits. Extensive experiments validate that SandwichR effectively balances high precision with low latency, presenting a practical solution for deployment-sensitive scenarios.

7 Limitations

Despite its superior correction performance and high efficiency, SandwichR still has several limitations: In this paper, due to resource constraints, we only conducted experiments using a publicly available large language model (LLM) with a scale of 1.5 billion parameters. Employing larger-scale LLMs with richer world knowledge as a component of SandwichR is expected to yield better correction performance. Furthermore, with advances in LLM research, techniques such as Retrieval-Augmented Generation (RAG) (Jin et al., 2025; Zhang et al., 2025a)—which rely on external knowledge bases—offer opportunities to correct queries beyond real-time knowledge and the internal knowledge of the model. In future work, we plan to further improve the query correction performance by integrating these techniques.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

Christopher Davis, Andrew Caines, Øistein Andersen, Shiva Taslimipoor, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. Prompting open-source and commercial language models for grammatical error correction of english learner text. *arXiv preprint arXiv:2401.07702*.

Chenhe Dong, Shaowei Yao, Pengkun Jiao, Jianhui Yang, Yiming Jin, Zerui Huang, Xiaojiang Zhou, Dan Ou, Haihong Tang, and Bo Zheng. 2025. Taosr1: The thinking model for e-commerce relevance search. *Preprint*, arXiv:2508.12365.

Yaxin Fan, Feng Jiang, Peifeng Li, and Haizhou Li. 2023. Grammargpt: Exploring open-source llms for native chinese grammatical error correction with supervised fine-tuning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 69–80. Springer.

Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk, and Xu Sun. 2010. A large scale ranker-based system for search query spelling correction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 358–366.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Jiajie Jin, Yutao Zhu, Zhicheng Dou, Guanting Dong, Xinyu Yang, Chenghao Zhang, Tong Zhao, Zhao Yang, and Ji-Rong Wen. 2025. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 737–740.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Preprint*, arXiv:1910.13461.

Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023. On the (in) effectiveness of large language models for chinese text correction. *arXiv preprint arXiv:2307.09007*.

Dingkun Long, Qiong Gao, Kuan Zou, Guangwei Xu, Pengjun Xie, Ruijie Guo, Jian Xu, Guanjun Jiang, Luxi Xing, and Ping Yang. 2022. Multi-cpr: A multi domain chinese dataset for passage retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3046–3056.

Madhura Pande, Vishal Kakkar, Manish Bansal, Surender Kumar, Chinmay Sharma, Himanshu Malhotra, and Praneet Mehta. 2022. Learning-to-spell: Weak supervision based query correction in e-commerce search with small strong labels. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3431–3440.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Hang Yan, Fei Yang, Zhe Li, Hujun Bao, and Xipeng Qiu. 2024a. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *Science China Information Sciences*, 67(5):1–13.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Qwen Team. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

Boshi Wang, Xiang Yue, and Huan Sun. 2023. Can chatgpt defend its belief in truth? evaluating llm reasoning via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11865–11881.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems*, 37:66383–66409.

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. Fcgec: Fine-grained corpus for chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1900–1918.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. *Preprint*, arXiv:2010.11934.

Dezhi Ye, Bowen Tian, Jiabin Fan, Jie Liu, Tianhua Zhou, Xiang Chen, Mingming Li, and Jin Ma. 2023. Improving query correction using pre-train language model in search engines. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2999–3008.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Kepu Zhang, Zhongxiang Sun, Weijie Yu, Xiaoxue Zang, Kai Zheng, Yang Song, Han Li, and Jun Xu. 2025a. Qe-rag: A robust retrieval-augmented generation benchmark for query entry errors. *arXiv preprint arXiv:2504.04062*.

Kepu Zhang, Zhongxiang Sun, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Jun Xu. 2025b. Trigger3: Refining query correction via adaptive model selector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 13260–13268.

A Appendix

A.1 Prompts

This section presents the detailed prompt templates used for training and evaluation of the three reasoning formats: **Reasoning-Answer**, **Answer-Reasoning**, and our proposed **SandwichR**. Each prompt is designed to instruct the model to follow a specific output structure while performing query correction. In these templates, we use colored tags to indicate different components: the `<reasoning>...</reasoning>` tags indicate the reasoning trace, the `<answer>...</answer>` tags indicate the corrected query output, and the `[original query]` placeholder represents the actual input query. The prompt structure explicitly defines the output format, ensuring the model generates responses in the desired sequence.

Reasoning-Answer:
You are a Chinese text error correction tool that can detect and correct errors in the text. Please check the errors in the following text, correct them, modify only the erroneous parts while keeping the original sentence structure as much as possible, provide your reasoning process, and output the corrected version. Please strictly use the following format for your reply: **<reasoning>** (briefly analyze the location, type, and basis of the error) **</reasoning>** \n **<answer>** (output the corrected full text) **</answer>**. **[original query]**

Answer-Reasoning:
You are a Chinese text error correction tool that can detect and correct errors in the text. Please check the errors in the following text, correct them, modify only the erroneous parts while keeping the original sentence structure as much as possible, first output the corrected version, and then provide your reasoning process. Please strictly use the following format for your reply: **<answer>** (output the corrected full text) **</answer>** \n **<reasoning>** (briefly analyze the location, type, and basis of the error) **</reasoning>**. **[original query]**

SandwichR :
You are a Chinese text error correction tool that can detect and correct errors in the text. Please check the errors in the following text, correct them, modify only the erroneous parts while keeping the original sentence structure as much as possible, first output the corrected version, then provide your reasoning process, and finally output the corrected version again. Please strictly use the following format for your reply: **<answer>** (first output the corrected full text) **</answer>** \n **<reasoning>** (briefly analyze the location, type, and basis of the error) **</reasoning>** \n **<answer>** (output the corrected full text again) **</answer>**. **[original query]**

A.2 More Implementation Details

During SFT training, we performed LoRA (Hu et al.) fine-tuning. The LoRA hyperparameters were set to r=8 and alpha=16. Training was performed on a curated dataset of 1,000 samples with a batch size of 32 and a learning rate of 5e-5. The model was trained for 30 epochs, and the checkpoint that achieved the best performance on the validation set was selected. This optimal checkpoint served both as the final model for the SFT stage evaluation and as the initial policy model for the subsequent GRPO training. During GRPO training, we conducted full-parameter fine-tuning which uti-

Table 6: Performance Comparison: Single Mixed-Domain Model trained on combined data vs. Three Domain-Specific Models trained separately on each dataset using the same two-stage training strategy.

Training Strategy	E-commerce		Video		Medical	
	$F_{0.5}$	Acc	$F_{0.5}$	Acc	$F_{0.5}$	Acc
Mixed-Domain (Single Model)	0.209	0.201	**0.343**	**0.334**	**0.425**	**0.407**
Domain-Specific (Three Models)	**0.221**	**0.213**	0.325	0.307	0.396	0.375

lized a dataset of 200 samples with a batch size of 8. We employed a learning rate of 1e-5 and 20 epochs. with a cosine decay schedule, which decreased to a final value of 1e-6. The maximum completion length was 256 tokens. Models were trained for 20 epochs. and the checkpoint that performed best on the validation set was selected for final evaluation.

A.3 Task Merging

We conducted a task merging experiment to evaluate the generalization capability of our SandwichR method. A single unified SandwichR model was trained on the combined datasets from all three domains (total 3,000 SFT and 600 GRPO samples), using the same training strategy, and compared against three domain-specific models trained separately on each individual dataset.

Results as shown in Table 6 demonstrate a clear cross-domain transfer effect: the unified model outperformed the domain-specific models on the Video and Medical datasets. However, its performance on the E-commerce dataset was slightly lower than that of the model trained solely on E-commerce data. We attribute this pattern to two factors: (1) the inherent difficulty and lower baseline accuracy of the E-commerce task (as shown in Table 1 of the main paper), and (2) during joint training, the model likely prioritized learning patterns from the relatively simpler Video and Medical tasks, which exhibit more regular error patterns and clearer correction boundaries.