THINKGEC: A Cognitive and Pedagogical Framework for LLM-based Grammatical Error Correction

Anonymous ACL submission

Abstract

LLMs are increasingly used for learner-facing writing support, yet grammatical error correction still lacks cognitively aligned training, pedagogically curated data, and interpretable feedback. We present **THINKGEC**, a three-stage framework grounded in Corder's identification–description–explanation paradigm, formalized as *knowledge elicitation* from expert annotations, *knowledge injection* via supervised fine-tuning, and *explanation* through GRPO-guided self-revision. To support this framework, we release **De10K**, an education-oriented German GEC corpus containing 2,899 essays and 14,330 expert-annotated errors across diverse topics and proficiency levels. Experiments demonstrate that THINKGEC substantially outperforms strong baselines, improves precision, mitigates over-correction, and generalizes to held-out semantically driven error types. Further analysis investigates the influence of reasoning trajectories, model scale, and reinforcement design. THINKGEC delivers interpretable, pedagogically aligned rationales, advancing both the accuracy and educational value of LLM-based GEC. Our code and dataset is avaiable at: https://anonymous.4open.science/r/ThinkGEC-04E7

1 Introduction

The rapid advancement of Large Language Models (LLMs) has driven remarkable progress across a wide range of domain-specific text generation tasks, including legal drafting (Guha et al., 2023), medical question answering (Liu et al., 2024), and code generation (Jimenez et al., 2024). In particular, education-oriented applications have gained increasing attention, such as subject-based tutoring (Jiang and Jiang, 2024), automated mathematical problem solving (Shao et al., 2024), and classroom feedback simulation (Zhang et al., 2024). Owing to their multilingual and interactive capabilities, LLMs are now being increasingly adopted to support second language learning, most notably in essay writing and Grammatical Error Correction (GEC) (Liang et al., 2025; Liu et al., 2025a; Li and Lan, 2025), where models assist learners by identifying and revising linguistic errors in their written texts.

Traditional GEC approaches (Zhao and Wang, 2020; Ye et al., 2023; Rothe et al., 2021; Zhou et al., 2023; Zhang et al., 2022b; Fan et al., 2023; Fang et al., 2023) treat correction as a direct end-to-end mapping from erroneous to corrected text. While effective in producing fluent outputs, they overlook explicit linguistic reasoning and constraints such as minimal edits and meaning preservation. Recent LLM-based systems benefit from large-scale pre-training, yet their highly coherent rewriting often leads to overcorrection and obscures the intermediate reasoning steps, offering little interpretability for beginner language learners. Consequently, building learner-oriented GEC systems with off-the-shelf models is non-trivial.

We identify three major limitations that hinder the effective use of LLMs for learner-oriented GEC: 1) **Insufficient High-quality Data**: despite the abundance of multilingual pre-training text corpora, there is a lack of educationally curated datasets that capture learner-specific errors across proficiency levels and instructional contexts. Existing resources rarely include fine-grained annotations that reflect how learners make and correct mistakes. 2) **Cognitively Inconsistent Correction Process**: Current one-shot correction paradigms treat GEC as a direct transformation from error to correction, ignoring the step-by-step reasoning and incremental feedback that characterize human grammatical learning. This lack of cognitive alignment with how humans process and understand errors creates confusion for entry-level learners, who often see *what* was corrected but fail to grasp *why* or *how* the correction was made. 3) **Lack of Interpretable Explanations**: Most existing systems provide fluent rewrites without explicit justifications or linguistic

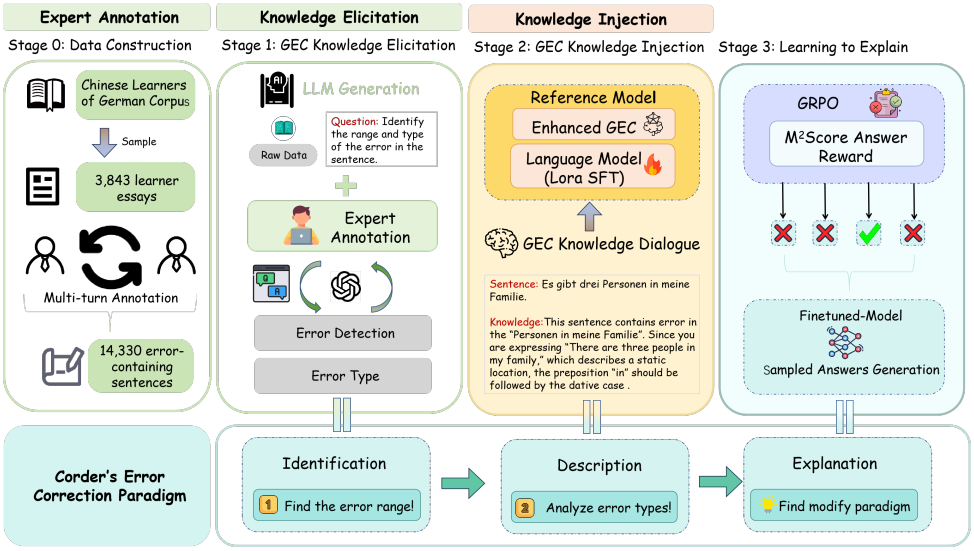

Figure 1: The overall pipeline of THINKGEC, comprising three core components: (1) GEC Knowledge Elicitation (2) GEC Knowledge Injection and (3) Learning to Explain.

explanations, offering limited pedagogical value. For educational scenarios, interpretability is essential, not only for evaluating model reliability but also for supporting learners' awareness and self-correction ability.

To address these limitations, we present THINKGEC, a cognitively aligned, three-stage framework for GEC grounded in Corder's classical error processing theory (Corder, 1975) and second language acquisition principles. Our approach bridges linguistic reasoning with machine learning, modeling human-like correction behavior through sequential stages of **identification, description, and explanation**. To support this framework, we introduce **De10K**, the first linguistically grounded and pedagogically curated corpus for German GEC. Developed over 3 years of interdisciplinary collaboration with pedagogical experts with decades of German education experience, **De10K** contains **2,899** essays and **14,330** manually annotated error instances. The dataset covers a wide range of topics and learner proficiency levels, ensuring both linguistic depth and educational representativeness. Aligned with human cognitive reasoning in GEC, THINKGEC formalizes these three stages through: (1) knowledge elicitation via identification-oriented data enhancement, (2) knowledge injection through supervised fine-tuning (SFT), and (3) iterative self-revision guided by Group Relative Policy Optimization (GRPO). Experimental results demonstrate that our approach not only achieves high accuracy in grammatical categorization but also exhibits strong generalization to out-of-domain (OOD) writing tasks, providing interpretable and pedagogically meaningful correction trajectories.

Our key contributions are concluded as follows:
1) We introduce **De10K**, a linguistically principled and educationally oriented German GEC dataset that provides fine-grained, expert-annotated error representations aligned with real learner contexts.
2) We proposed THINKGEC, a unified GEC framework that bridges human cognitive strategies with machine learning by integrating structured linguistic knowledge into data construction and reasoning-based optimization.
3) Through comprehensive experimental analysis, we not only demonstrate ThinkGEC's high correction accuracy and strong generalization, but also provide an in-depth analysis of the effects of reasoning trajectories, model scale in reinforcement sampling, and the task-specific behavior of GRPO within GEC.

2 Methodology

2.1 Preliminary

Our GEC task aims to automatically detect and correct linguistic errors in texts produced by non-native speakers. Rooted in second language acquisition theory, Corder's seminal framework conceptualizes error processing as a three-stage cognitive sequence: ***identification, description, and explanation*** of learner errors (Corder, 1975). This sequence captures the reasoning process human annotators follow: first *identifying* linguistic anomalies, then *describing* them through classification taxonomies,

and finally *explaining* them by generating appropriate corrections. It not only illustrates the cognitive structure underlying manual correction but also provides an **interpretable analogy** for designing computational GEC systems.

Building on this insight, we formulate GEC as an iterative refinement process guided by structured linguistic knowledge, as illustrated in Figure 1. Our proposed **THINKGEC** framework mirrors Corder's cognitive sequence (Corder, 1975) and decomposes GEC into three corresponding stages:

1) Identification: the recognition of grammatical abnormalities corresponds to **knowledge elicitation**, where LLMs are guided through an annotation-to-reasoning protocol that transforms discrete error labels into explicit linguistic rationales. This stage bridges human error recognition and model understanding by making expert annotation knowledge interpretable and learnable.

2) Description: the categorization of errors corresponds to SFT with **knowledge injection**, aligning model predictions with labeled examples and establishing a structured mapping between error types and corrective semantics. This enables the model to internalize both the *what* and *why* behind each error.

3) Explanation: the generation of justified corrections corresponds to GRPO (DeepSeek-AI et al., 2025), where the model iteratively samples revisions and refines its outputs through composite rewards, mirroring how humans improve grammatical competence through feedback and refinement.

Together, these stages form the foundation of our **THINKGEC** framework, subsequent sections detail how each stage is instantiated.

2.2 Stage 0: Building De10K with Multi-Layered Expert Annotation

To support the identification and knowledge elicitation stages in our THINKGEC framework, we propose **De10K**, the first high-quality GEC dataset with extensive human-annotated German error correction data. Build upon the Chinese Learners of German Corpus (CDLK), **De10K** collects authentic learner writing samples from German learners at various proficiency levels and categorizes errors based on a typology specifically designed for Chinese learners of German.

We selected 3,843 learner essays for in-depth linguistic annotation. After filtering for syntactic and morphosyntactic deviations, we extracted

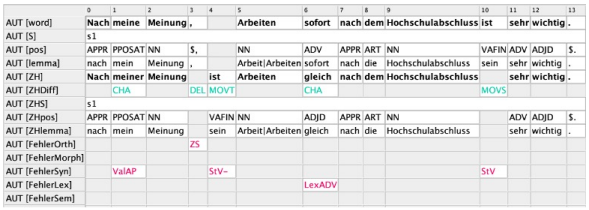

Figure 2: The interface of EXMARaLDA.

14,330 error-containing sentences from 2,899 distinct learner texts. **De10K** offers a linguistically grounded and pedagogically relevant resource for training and evaluating GEC systems focused on German as a Second Language. Key strengths of the dataset include: (1) authentic, topic-diverse writing samples that reflect real learner experiences; (2) broad representativeness across learner proficiency levels, age groups, and educational backgrounds; (3) fine-grained, multi-level annotations, informed by linguistic theory, that provide deep insight into error patterns in L2 German writing.

To develop a learner-specific error typology, we conducted two pilot annotation studies following the framework proposed by Nolda et al. (2023). Starting with an initial framework, we refined the classification through iterative annotation and collaborative analysis. Recurrent error patterns led to the introduction of new categories, including reflexive pronouns in verb valency, prepositions introducing zu-clauses, and prefixes of separable verbs. Categories with low occurrence, such as syllable division, were removed. The lexical choice category was further subdivided by part of speech, and the final system consists of five main categories: Orth, MorSyn, Wortstellung, Valenz, and Syn, together with a *Semantically-driven Error* type, organized into 63 subcategories in total.

Annotations were carried out on learner essays across proficiency levels using EXMARaLDA, as shown in Figure 2. Each text was transformed into a target hypothesis (Zielhypothese) and annotated hierarchically according to the proposed typology. The annotation process adhered to two core principles: **fidelity and consistency**. **Fidelity** ensured minimal intervention, preserving the learner's intended meaning while correcting only what was necessary for grammaticality and orthographic accuracy in standard German (Ellis, 1994). **Consistency** was maintained through standardized labeling protocols and detailed annotation guidelines to ensure uniform treatment of linguistic phenomena across the corpus.

All annotators (instructors and linguistics graduate students) underwent training and calibration. Each essay was independently annotated by two experts, followed by adjudication. Disagreements were resolved in team sessions, with a subset externally validated by senior linguists. Regular calibration meetings ensured sustained quality throughout the six-month annotation process. This rigorous, multi-stage protocol ensures high reliability and linguistic validity, making **De10K** a robust resource for German GEC.

2.3 Stage 1: Formalizing Error Identification through Knowledge Elicitation

The **De10K** corpus provides token-level error annotations with span and type labels derived from a fine-grained, theory-driven taxonomy. While these symbolic annotations precisely locate morpho-syntactic and lexical errors, they lack the **natural-language warrant**, an explanatory reasoning that justifies *why a span constitutes an error*, *what grammatical constraint is violated*, and *how the annotation decision is made*. Without such explicit rationales, LLMs struggle to infer the linguistic principles underlying error identification.

To bridge this gap, we design a knowledge elicitation pipeline that transforms discrete annotation labels into interpretable, pedagogically grounded rationales. Using an off-the-shelf LLM as an **expert-in-the-loop**, we induce reasoning trajectories in two steps:

Step 1: Error Scope Verbalization. Given an erroneous sentence and its gold error span, the model generates a natural-language description of the deviation's position and surface realization, converting positional tags into explicit *scope specifications*.

Step 2: Error Classification Reasoning. Conditioned on the original sentence and the generated scope, the model infers the error type using grammatical constraints (e.g., agreement, subcategorization), producing a *reasoning trajectory* that mirrors human annotation logic.

See Appendix D.1 for instruction templates. Each synthetic example includes detection, scope, and reasoning-based classification, converting implicit expert knowledge into explicit narrative supervision. This enriched corpus forms the basis of Stage 2, where SFT injects these elicited rationales into the model's representation, aligning linguistic identification with human reasoning.

2.4 Stage 2: Learning Error Description through Knowledge Injection

To instill human-aligned reasoning into the LLM, we perform SFT on a dialogue-based synthetic dataset. This process transforms structured linguistic annotations (e.g., error spans, type labels) into naturalistic, multi-turn correction dialogues that explicitly model the diagnostic reasoning chain. By training on these pedagogical interactions, the model internalizes not only *What* to correct, but also *How* and *Why*, thereby acquiring robust syntactic correction capabilities.

Formally, given a pretrained LLM M_θ parameterized by θ, we construct a training dataset:

$$\mathcal{D}_{\text{SFT}} = \left\{ (x'^{(i)},\ r^{(i)}) \right\}_{i=1}^{M},$$

where $x'^{(i)} = p^{(i)} \circ x^{(i)}$ denotes the corrupted or prompt-augmented input, and $r^{(i)}$ is the target reasoning trajectory, consisting of a natural language description of the error scope and error type.

The model is optimized to maximize the conditional likelihood of generating these reasoning steps:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x',\, r)\sim\mathcal{D}_{\text{SFT}}} \left[\log p_\theta(r \mid x') \right], \quad (1)$$

yielding a refined model $M_{\theta'}$, which serves as a strong reference policy for subsequent stages of the pipeline. Through this knowledge-injected SFT phase, M_θ learns to condition its generation not only on the input sentence, but also on structured diagnostic reasoning encoded in p, laying the foundation for advanced correction strategies.

2.5 Stage 3: Learning to Explain via GRPO

Prior work (Liu et al., 2025a; Liang et al., 2025) has demonstrated the efficacy of SFT in enabling models to acquire a robust paradigm, especially in high-resource English GEC settings where large, diverse datasets support error categorization. However, for low-resource languages like German, solely SFT struggles to capture the full spectrum of syntactic error patterns, leading to overfitting and poor generalization.

To overcome these limitations, we involve GRPO to leveragee the syntactic priors embedded in pre-trained LLMs. GRPO's iterative, trial-based learning process, where policies are refined through feedback over multiple correction attempts, **mirrors how humans acquire grammatical competence through repeated practice and iterative refinement**.

Building upon the SFT-initialized policy $M_{\theta'}$, we perform reinforcement learning to further refine the model's correction behavior. At each training step, for each erroneous sentence x and fixed prompt p, we sample a set of G candidate outputs $\{r_1, r_2, \ldots, r_G\}$ from the current policy $p_\theta(\cdot \mid p \circ x)$, where each r_i represents a reasoning trajectory or corrected output. The reward advantage A_i for each output is calculated by normalizing the group rewards $\{r_1, r_2, \ldots, r_G\}$:

$$A_i = \frac{r_i - \mu_r}{\sigma_r}, \tag{2}$$

where $r_i = \text{M}^2\text{Score}(x, y_i)$ represents the reward assigned to the corrected output y_i, and μ_r and σ_r are the mean and standard deviation of the group rewards.

Let $x' = p \circ x$ denote the prompt-augmented or repaired input. Following recent GRPO variants (Yu et al., 2025; Liu et al., 2025b) that employ asymmetric clipping, we adopt an upper-bound clipping strategy combined with a direct KL penalty to stabilize training:

$$\begin{aligned} J_{\text{GRPO}}(\theta'') = \mathbb{E}_{x,\{y_i\}_{i=1}^G} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|y_i|} \cdot \\ \min\Big(\rho_i(\theta'')\, A_i,\ \text{clip}\big(\rho_i(\theta'')\big)\, A_i\Big) \\ -\beta \cdot \mathcal{D}_{KL}\big(p_{\theta''}(\cdot \mid x') \,\|\, p_{\theta'}(\cdot \mid x')\big)\Bigg], \end{aligned} \tag{3}$$

where the likelihood ratio is defined as

$$\rho_i(\theta'') = \frac{p_{\theta''}(y_i \mid x')}{p_{\theta_{\text{old}}}(y_i \mid x')}, \tag{4}$$

and $p_{\theta_{\text{old}}}$ denotes the policy from the previous optimization step (i.e., the old version of θ'').

The clipping function applies asymmetric bounds:

$$\begin{aligned} \text{clip}\big(\rho_i(\theta'')\big) = \text{clip}\big(\rho_i(\theta''),\ 1 - \varepsilon_{\text{low}}, \\ 1 + \varepsilon_{\text{high}}\big). \end{aligned} \tag{5}$$

The KL divergence regularizes the current policy $p_{\theta''}$ to prevent excessive deviation from the SFT-initialized reference model $M_{\theta'}$:

$$\begin{aligned} \mathcal{D}_{KL}\big(p_{\theta''} \,\|\, p_{\theta'}\big) = \mathbb{E}_{y \sim p_{\theta''}(\cdot \mid x')}\big[\log p_{\theta''}(y \mid x') \\ - \log p_{\theta'}(y \mid x')\big]. \end{aligned} \tag{6}$$

By using GRPO, the model learns to refine its correction strategy iteratively, improving its ability to make contextually appropriate corrections while balancing correctness, fluency, and minimal modification.

3 Experiments

3.1 Datasets and Evaluation

We categorize the dataset into six major error types and use five of them for model training, reserving the sixth *Semantically-driven Error* category to evaluate the model's **generalization ability**. The training split contains 12,897 annotated sentences, and evaluation is performed on a held-out test set of 1,433 sentences. For evaluation, we employ the **M^2 Scorer** (Dahlmeier and Ng, 2012) to compute **precision**, **recall**, and $F_{0.5}$, which places higher weight on precision to better reflect correction quality.

3.2 Base Models and Baselines

Base Models. We evaluate our framework using two open-source LLMs: Qwen3 (Yang et al., 2025) and LLaMA3.1 (Grattafiori et al., 2024). Our primary experiments are conducted with their 8B-parameter variants due to computational constraints. To analyze the effect of model scale, we further perform an ablation study using the 4B-parameter variant of Qwen3.

Baselines. We evaluate THINKGEC through controlled comparisons along three dimensions:
1) **Standard SFT**: models are trained directly on error–correction pairs using cross-entropy loss, without any explicit reasoning component.
2) **Optimizer Substitution**: we replace the GRPO optimizer with its variant in THINKGEC pipeline, Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025), which employs adaptive clipping thresholds and dynamic sampling for improved policy stability.
3) **Reasoning Ablation**: we remove the rationale generation step while keeping the optimization and reward setup unchanged, isolating the contribution of explicit reasoning to correction performance.

In addition, we benchmark THINKGEC against state-of-the-art (SOTA) LLMs under identical evaluation protocols to contextualize its overall effectiveness.

3.3 Main Result

The main results are summarized in Table 2. Across all base models and training configurations, the THINKGEC framework consistently outperforms strong baselines in $F_{0.5}$ score, confirming its effectiveness for GEC.

When applied to Qwen3-8B and LLaMA3.1-8B, THINKGEC achieves an average gain of +2.0 $F_{0.5}$ over SFT. These gains are consistent across model

Table 1: Results on the GEC benchmark. All results are obtained through our own implementations under consistent experimental settings. The error categories Orth, MorSyn, Wortstellung, Valenz, and Syn follow the typology defined in data construction. The best-performing models in each category are **bolded**, and this notation is consistently used in all subsequent tables.

Model	Orth			MorSyn			Wortstellung			Valenz			Syn		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Qwen3-8B															
Base	62.80	62.80	62.80	67.83	62.40	66.67	72.41	58.88	69.23	64.23	51.46	61.20	52.68	47.20	51.48
+SFT	66.67	68.00	66.93	71.65	72.80	71.88	75.00	61.68	71.90	**78.31**	76.02	77.84	**64.61**	**62.80**	**64.24**
+THINKGEC	**71.71**	**74.00**	**72.15**	**74.22**	**76.00**	**74.57**	**81.32**	**69.16**	**78.56**	77.40	**80.12**	**77.93**	61.63	60.40	61.38
+THINKGEC(DAPO)	68.85	71.60	69.38	70.31	72.00	70.64	76.14	62.62	72.98	73.29	69.01	72.39	57.44	55.60	57.06
LLaMA3.1-8B															
Base	53.14	64.40	55.06	55.92	68.00	57.98	64.76	63.55	64.52	46.08	54.97	47.62	40.38	50.40	42.06
+SFT	67.94	71.20	68.57	67.94	71.20	68.57	75.28	62.62	72.35	**80.84**	**78.95**	**80.45**	60.54	**63.20**	61.05
+THINKGEC	**71.92**	**74.80**	**72.48**	70.68	**75.20**	71.54	73.03	60.75	70.19	79.52	77.19	79.04	**64.05**	62.00	**63.63**
+THINKGEC(DAPO)	70.37	68.40	69.97	**72.58**	72.00	**72.46**	**80.95**	**63.55**	**76.75**	80.50	74.85	79.31	62.92	60.40	62.40
SOTA Model															
GPT-4o	**68.81**	81.20	70.98	67.83	**77.60**	69.58	68.22	68.22	68.22	62.76	71.93	64.40	50.00	**60.40**	51.78
GPT-o1	63.89	**82.80**	66.95	62.34	76.80	64.78	64.46	72.90	65.99	62.93	**75.44**	**65.09**	44.35	59.60	46.74
DeepSeek-V3	68.99	79.20	**70.82**	**67.63**	75.20	**69.02**	66.67	71.03	67.50	**63.89**	67.25	64.53	50.00	58.40	51.48
DeepSeek-R1	62.46	75.20	64.65	58.16	65.60	59.51	**70.18**	**74.77**	**71.05**	54.82	63.16	56.31	**47.39**	58.00	**49.19**

families, indicating that the benefits of our framework generalize beyond specific LLM architectures. Moreover, THINKGEC surpasses existing SOTA GEC systems, setting a new benchmark for syntactically aware correction.

Table 1 further details performance by grammatical error type. THINKGEC yields clear improvements across all categories, with the largest gains observed for morphologically and lexically localized errors. For more complex cases involving long-range dependencies or structural reordering, the improvement narrows, suggesting that GRPO's effectiveness depends on the syntactic plausibility of sampled candidate corrections. In such settings, generating well-formed yet semantically faithful alternatives remains challenging.

Performance gains primarily stem from higher precision, with a modest reduction in recall. This trade-off aligns with established GEC evaluation preferences, where avoiding false corrections is more desirable than over-editing (Ng et al., 2014). In effect, THINKGEC mitigates the over-correction tendency common in LLMs.

Despite architectural and scale differences among base LLMs, THINKGEC consistently delivers substantial improvements, underscoring its robustness and adaptability to diverse pretrained backbones.

3.4 Out Of Domain Result

Generalization to unseen error types remains a central challenge in GEC due to the open-ended variability of learner language. To evaluate this capability, we test models on a held-out *Semantically-driven Error* category entirely excluded from training, as shown in Table 3.

Table 2: Results on the **De10K** dataset.

Model	P	R	$F_{0.5}$
Qwen3-8B			
Base	63.73	45.67	59.06
+SFT	71.57	60.38	69.02
+THINKGEC	**74.85**	**59.60**	**71.21**
+THINKGEC(DAPO)	71.12	56.27	67.56
LLaMA3.1-8B			
Base	55.03	51.45	54.27
+SFT	72.95	61.40	70.31
+THINKGEC	**75.19**	**61.78**	**72.06**
+THINKGEC(DAPO)	74.95	58.53	70.97
SOTA Model			
GPT-4o	**66.84**	63.99	**66.25**
GPT-o1	63.25	**67.24**	64.01
DeepSeek-V3	66.45	61.09	65.31
DeepSeek-R1	60.97	58.15	60.38

SFT performs poorly under this setting, reflecting its dependence on memorized correction patterns. In contrast, THINKGEC achieves substantially higher $F_{0.5}$ scores, demonstrating stronger reasoning and generalization beyond the training distribution.

These gains arise from its three-stage optimization framework, which promotes linguistic abstraction through iterative comparison and reward-guided refinement. By assessing candidate corrections for grammatical plausibility and contextual coherence, the model learns correction principles that transfer to unseen phenomena. This ability to generalize beyond annotated coverage is essential for real-world deployment, where learner errors are far more diverse than training data. THINKGEC

thus improves both robustness and adaptability, extending its advantages from in-domain to genuinely open-domain GEC.

Table 3: Performance comparison on the *Semantically-driven Error* category.

Model	P	R	$F_{0.5}$
Qwen3-8B			
Base	36.13	38.13	36.51
+SFT	32.41	32.57	32.44
+THINKGEC	39.69	**40.89**	39.93
+w/o-think THINKGEC	**44.93**	36.45	**42.93**
LLaMA3.1-8B			
Base	25.84	38.12	27.62
+SFT	34.97	32.69	34.49
+THINKGEC	**39.80**	**33.34**	**38.32**
SOTA Model			
GPT-4o	**35.98**	**51.36**	**38.27**
GPT-o1	27.46	50.24	30.20

4 Analysis

This section provides a comprehensive analysis of the key factors contributing to the performance of THINKGEC: reasoning trajectories, model scale, and reinforcement optimization. All experiments are conducted under controlled settings to ensure valid comparisons.

4.1 Impact of Reasoning Trajectories

We examine the effect of reasoning trajectories on syntactic error correction under two training paradigms: THINKGEC and standard SFT. Results show that the utility of reasoning is not inherent but depends critically on how it is integrated into the learning framework.

Within THINKGEC, we compare Qwen3-8B models trained with and without reasoning trajectories. As summarized in Table 4, their impact varies with error complexity. For simple surface-level errors such as spelling mistakes, reasoning slightly degrades performance by triggering unnecessary edits, a form of overcorrection arising from excessive logical inference. For moderately complex grammatical phenomena such as verb conjugation or article agreement, reasoning provides consistent gains by directing policy updates toward linguistically valid revisions, suggesting that structured reasoning enhances diagnostic precision once the model possesses sufficient grammatical competence.

However, for high-complexity or strongly context-sensitive errors, reasoning again leads to degraded performance. We attribute this to the generation of incoherent or hallucinated rationales caused by limited German grammatical proficiency, which distorts groupwise reward ranking and reinforces suboptimal behaviors. Hence, reasoning benefits THINKGEC only when the generated rationales are linguistically reliable.

We further assess the same mechanism under the SFT setting using Qwen2.5-7B (Qwen et al., 2025). Two variants are trained: one on direct input–output pairs, and another augmented with reasoning traces distilled from DeepSeek-R1. The latter consistently underperforms, despite containing additional diagnostic signals. This degradation arises from a structural mismatch: SFT minimizes token-level cross-entropy over the entire output sequence. Because erroneous and corrected sentences are highly similar, the cumulative loss from long reasoning traces dominates the final correction loss, causing the model to optimize for fluency rather than correction accuracy.

Overall, **reasoning trajectories are not universally advantageous**. Within THINKGEC, they provide structured feedback for policy refinement; within solely SFT, they act as optimization noise. This contrast underscores that the value of reasoning supervision is inherently framework-dependent and contingent on its integration into the broader learning dynamics.

4.2 Impact of Model Scale

Table 5 summarizes the effect of THINKGEC across different model sizes. The results indicate a strong dependency on model capacity: larger models benefit substantially from reasoning-augmented training, while smaller ones exhibit marginal or even negative gains.

This disparity stems from the *varying quality of generated reasoning trajectories*. Larger models with greater linguistic competence produce coherent and logically grounded rationales, which provide informative signals for group-wise policy refinement. Conversely, smaller models often yield fragmented or factually inconsistent reasoning, introducing noise into the reward computation and destabilizing the optimization process. When such low-quality rationales dominate, the relative ranking signal becomes unreliable, leading to degraded learning dynamics.

Consequently, the effectiveness of THINKGEC hinges on a representational threshold: **larger models with sufficient grammatical awareness are capable produce reliable reasoning signals for**

Table 4: Performance comparison between THINKGEC with/without reasoning trajectories on five error categories.

Model	Orth			MorSyn			Wortstellung			Valenz			Syn		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Qwen3-8B															
think	71.71	74.00	72.15	74.22	76.00	74.57	**81.32**	**69.16**	**78.56**	77.40	**80.12**	**77.93**	61.63	60.40	61.38
w/o-think	**73.91**	**74.80**	**74.09**	**75.59**	**76.80**	**75.83**	75.56	63.55	72.81	**78.98**	72.51	77.60	**62.30**	**60.80**	**61.99**
Qwen2.5-7B															
Base	51.60	58.00	52.77	55.40	61.60	56.53	68.75	61.68	67.21	52.33	52.63	52.39	38.54	44.40	39.59
+SFT	**63.67**	**70.80**	**64.98**	**62.59**	**69.60**	**63.88**	**70.83**	**63.55**	**69.25**	**76.40**	**79.53**	**77.01**	**55.47**	**60.80**	**56.46**
+Reason SFT	57.20	62.00	58.10	58.78	61.60	59.32	62.64	53.27	60.51	63.58	60.23	62.88	46.12	47.60	46.41

Table 5: Performance comparison of THINKGEC across models of varying sizes.

Model	P	R	$F_{0.5}$
Qwen3-4B			
Base	58.42	39.80	53.42
+SFT	71.25	**59.13**	68.44
+THINKGEC	69.52	53.17	65.49
+w/o-think THINKGEC	**73.98**	55.25	**69.28**
Qwen3-8B			
Base	36.13	38.13	36.51
+SFT	32.41	32.57	32.44
+THINKGEC	39.69	**40.89**	39.93
+w/o-think THINKGEC	**44.93**	36.45	**42.93**

reinforcement optimization. Below this threshold, the framework risks amplifying spurious correlations rather than improving syntactic accuracy.

4.3 Impact of GRPO Components

Table 6: Ablation study of GRPO components on **De10K** dataset.

Model	P	R	$F_{0.5}$
Qwen3-8B			
+THINKGEC	**74.85**	**59.60**	**71.21**
+THINKGEC(DAPO)	71.12	56.27	67.56
LLaMA3.1-8B			
+THINKGEC	**75.19**	**61.78**	**72.06**
+THINKGEC(DAPO)	74.95	58.53	70.97

We further examine the contribution of individual components in GRPO and its variant DAPO, within the THINKGEC framework. Surprisingly, as shown in Table 6, DAPO consistently underperforms GRPO, indicating that its additional modifications, though effective in long-sequence reasoning, do not transfer well to GEC. Three factors account for this discrepancy.

First, both algorithms employ higher clipping thresholds (`clip-high`) to prevent policy entropy collapse, a common issue in GEC where repetitive corrections reduce exploration. This mechanism preserves sampling diversity and remains beneficial in our implementation.

Second, token-level policy gradient optimization in DAPO offers limited gains for short-sequence editing tasks. Since GEC involves minimal token changes, intermediate gradient updates contribute little beyond sequence-level optimization.

Third, DAPO's dynamic sampling excludes extreme reward cases (e.g., 0 or 1), yet such samples are informative in GEC: perfect corrections reinforce valid editing patterns, while failed attempts reveal systematic grammatical weaknesses. Removing them weakens the learning signal and impairs policy refinement.

Further analysis are provided in Appendix D.3. Collectively, these findings underscore **that algorithmic advances in reinforcement learning must be carefully contextualized**, mechanisms beneficial for general text generation may be suboptimal for localized, low-entropy tasks such as GEC.

5 Conclusion

Inspired by Corder's theory of error analysis (Corder, 1975), we propose THINKGEC, a three-stage cognitively-inspired framework for grammatical error correction. Stage one distils fine-grained linguistic annotations into natural-language principles; stage two injects these principles to enhance syntactic–semantic accuracy; stage three optimises the correction trajectory with GRPO. Across architectures and metrics, THINKGEC consistently outperforms both baseline and supervised-fine-tuned Qwen3-8B and LLaMA3.1-8B, yielding higher $F_{0.5}$, exact-match, and lower edit-distance scores. By integrating symbolic knowledge into neural generation, THINKGEC advances the state of GEC.

Limitations

Due to constraints in computational resources and time, our experiments were limited to lightweight large language models (specifically 4B and 8B parameter variants), focusing on the application of

the THINKGEC framework to German grammar error correction. As such, the generalizability of the approach across larger model scales, additional NLP tasks, and other languages remains to be fully explored. We leave the extension to a broader range of models, tasks, and linguistic domains to future work.

Ethical Considerations

The data collection protocol for **De10K** was reviewed and approved by the Institutional Review Board (IRB) of School of Foreign Languages, Zhejiang University, ensuring compliance with ethical standards for research involving human participants. All learner writing samples were collected with informed consent, including explicit permission for linguistic analysis and non-commercial research use. Personally identifiable information was fully removed or anonymized prior to annotation.

Annotations were performed by members of the research team as part of their academic responsibilities, without additional financial compensation. The **De10K** dataset is released under the Apache License 2.0, promoting open and responsible research use. For detailed information on participants instruction, please refer to: https://anonymous.4open.science/r/ThinkGEC-04E7.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Stephen P Corder. 1975. Error analysis, interlanguage and second language acquisition. *Language teaching*, 8(4):201–218.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Rod Ellis. 1994. *The study of second language acquisition*. Oxford University.

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. *Preprint*, arXiv:2307.13923.

Tao Fang, Jinpeng Hu, Derek F. Wong, Xiang Wan, Lidia S. Chao, and Tsung-Hui Chang. 2023. Improving grammatical error correction with multimodal feature integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9328–9344, Toronto, Canada. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

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

Zhoumingju Jiang and Mengjun Jiang. 2024. Beyond answers: Large language model-powered tutoring system in physics education for deep learning and precise understanding. *Preprint*, arXiv:2406.10934.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. Swe-bench: Can language models resolve real-world github issues? *Preprint*, arXiv:2310.06770.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Jiquan Li, Junliang Guo, Yongxin Zhu, Xin Sheng, Deqiang Jiang, Bo Ren, and Linli Xu. 2022. Sequence-to-action: Grammatical error correction with action guided sequence generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10974–10982.

Xinyuan Li and Yunshi Lan. 2025. Large language models are good annotators for type-aware data augmentation in grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 199–213, Abu Dhabi, UAE. Association for Computational Linguistics.

Jiehao Liang, Haihui Yang, Shiping Gao, and Xiaojun Quan. 2025. Edit-wise preference optimization for grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3401–3414, Abu Dhabi, UAE. Association for Computational Linguistics.

Fenglin Liu, Zheng Li, Hongjian Zhou, Qingyu Yin, Jingfeng Yang, Xianfeng Tang, Chen Luo, Ming Zeng, Haoming Jiang, Yifan Gao, Priyanka Nigam, Sreyashi Nag, Bing Yin, Yining Hua, Xuan Zhou, Omid Rohanian, Anshul Thakur, Lei Clifton, and David A. Clifton. 2024. Large language models in the clinic: A comprehensive benchmark. *Preprint*, arXiv:2405.00716.

Xinpeng Liu, Bing Xu, Muyun Yang, Hailong Cao, Conghui Zhu, Tiejun Zhao, and Wenpeng Lu. 2025a. A chain-of-task framework for instruction tuning of LLMs based on Chinese grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8623–8639, Abu Dhabi, UAE. Association for Computational Linguistics.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Andreas Nolda, Laura Auteri, Natascia Barrale, Arianna Di Bella, and Sabine Hoffmann. 2023. Fehlerannotation und fehleranalyse am beispiel des deutsch-ungarischen lernerkorpus dulko. *Jahrbuch für internationale Germanistik*, 10:747–755.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51, Melbourne, Australia. Association for Computational Linguistics.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297. ACM.

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.

Yingying Wang, Cunliang Kong, Liner Yang, Yijun Wang, Xiaorong Lu, Renfen Hu, Shan He, Zhenghao Liu, Yun Chen, Erhong Yang, and Maosong Sun. 2021. Yaclc: A chinese learner corpus with multidimensional annotation. *Preprint*, arXiv:2112.15043.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Konstantin Yakovlev, Alexander Podolskiy, Andrey Bout, Sergey Nikolenko, and Irina Piontkovskaya. 2023. GEC-DePenD: Non-autoregressive grammatical error correction with decoupled permutation and decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1546–1558, Toronto, Canada. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jingheng Ye, Yinghui Li, Yangning Li, and Hai-Tao Zheng. 2023. MixEdit: Revisiting data augmentation and beyond for grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10161–10175, Singapore. Association for Computational Linguistics.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022a. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. Association for Computational Linguistics.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022b. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhanxin Hao, Jianxiao Jiang, Jie Cao, Huiqin Liu, Zhiyuan Liu, Lei Hou, and Juanzi Li. 2024. Simulating classroom education with llm-empowered agents. *Preprint*, arXiv:2406.19226.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part II*, page 439–445, Berlin, Heidelberg. Springer-Verlag.

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1226–1233.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *Preprint*, arXiv:2403.13372.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. Improving Seq2Seq grammatical error correction via decoding interventions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7393–7405, Singapore. Association for Computational Linguistics.

A Related Work

A.1 GEC Dataset

Research on GEC datasets has largely focused on English. Early resources such as NUCLE (Dahlmeier et al., 2013), drawn from student essays at the National University of Singapore and informed by TOEFL iBT annotations, laid the foundation for the field. JFLEG (Napoles et al., 2017) further advanced evaluation with native-speaker fluency edits, while the W&I corpus (Bryant et al., 2019) offers large-scale, systematically annotated data from non-native learners worldwide.

Beyond English, several Chinese GEC (CGEC) datasets have emerged. NLPCC (Zhao et al., 2018) was among the first benchmarks for non-native Chinese. CGED (Rao et al., 2018), YACLC (Wang et al., 2021), and MuCGEC (Zhang et al., 2022a) have since expanded the landscape with fine-grained annotations, high-quality filtering, and multi-dialect integration, forming key public resources for the CGEC community. In contrast, FCGEC (Xu et al., 2022) provides manually annotated, fluent texts written by native speakers, serving as a reference for grammaticality and fluency modeling. Together, these datasets support more comprehensive and context-aware CGEC research.

These datasets have played a pivotal role in advancing the development and evaluation of GEC systems.In contrast to the relatively rich resources available for English and, to a lesser extent, Chinese, low-resource languages such as German face significant challenges in GEC research due to the scarcity of annotated datasets.

A.2 Traditional GEC Methods

Traditional GEC methods can be broadly categorized into two paradigms: Seq2Edit and Seq2Seq.

The Seq2Edit approach frames GEC as a sequence labeling task, where models predict fine-grained edit operations for each token in the input sentence—such as deletion, insertion, or substitution (Awasthi et al., 2019; Stahlberg and Kumar, 2020; Li et al., 2022; Yakovlev et al., 2023). A landmark in this line of work is GECToR (Omelianchuk et al., 2020), which significantly advanced the field by introducing task-specific, morpheme-level transformations in addition to standard word-level edits.

In contrast, the Seq2Seq paradigm treats GEC as a monolingual machine translation task, employing an encoder-decoder architecture to generate fully corrected sentences from erroneous inputs (Zhao and Wang, 2020; Ye et al., 2023; Rothe et al., 2021; Zhou et al., 2023). Unlike Seq2Edit, which focuses on local token-level modifications, Seq2Seq aims to produce globally coherent outputs through end-to-end generation. More recent advancements in this framework (Zhang et al., 2022b; Fan et al., 2023; Fang et al., 2023) further improve performance by incorporating auxiliary information such as syntactic structures or explicit error detection signals.

B Dataset

We conduct some statistics on our corpus, including the length distribution of sentences, the nums of error types in each category.

Classify Guideline

Sentence Length Distribution We analyze the distribution of sentence lengths in terms of character count. As illustrated in Figure 3, the corpus exhibits a wide but manageable range: the shortest sentence contains 4 characters, while the longest spans 56 characters. This bounded variation suggests that the corpus is well-suited for integration with a variety of pre-trained language models—particularly those with constrained input length—without requiring aggressive truncation or segmentation strategies.

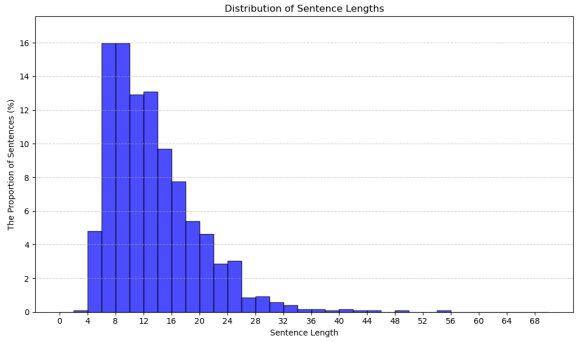

Figure 3: The length distribution of sentences in the whole corpus.

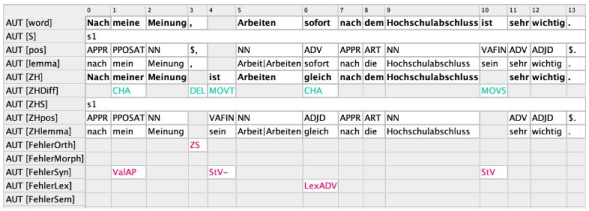

Figure 4: The interface of EXMARaLDA.

Error Type Distribution Table 7 presents the distribution of five major grammatical error categories across the training and test sets. The observed distribution is notably imbalanced, with certain error types (e.g., Orth, Syn) occurring significantly more frequently than others (e.g., Wortstellung, Valenz). This imbalance reflects authentic patterns in second-language acquisition, where learners exhibit non-uniform error tendencies influenced by linguistic transfer and developmental stages. Such characteristics underscore the realistic challenge posed by the corpus, making it a valuable benchmark for robust grammar error correction systems.

	Orth	MorSyn	Wortstellung	Valenz	Syn
• Train	4242	2472	1364	2949	4211
• Test	483	258	163	341	529

Table 7: Distribution of linguistic annotations across training and test sets.

B.1 Annotation Procedure

Error annotation was carried out using the EXMARaLDA (Figure 4) annotation environment. The process comprised two interrelated tasks: (i) correction of learner errors in the original text to produce a linguistically well-formed target hypoth-

Table 8: Hyperparameter settings in our experiments.

Hyperparameter	**De10K**		
Backbone	Qwen3-8B	LLaMA3.1-8B	Qwen3-4B
Batch size (SFT)	128	128	128
Batch size (GRPO)	32	32	64
Max Epochs (SFT)	5	5	5
Max Epochs (GRPO)	5	5	5
Max Length	4096	4096	4096
Learning Rate (SFT)	1e-5	1e-5	1e-5
Learning Rate (GRPO)	1e-6	1e-6	1e-6
Learning Rate Scheduler	Cosine	Cosine	Cosine
Optimizer	AdamW	AdamW	AdamW
Weight Decay	0.0	0.0	0.0
Warmup Ratio	0.1	0.1	0.1
LoRA	target modules = all linears; lora rank = 32; lora alpha = 64		
$\varepsilon_{\text{high}}$	0.28	0.28	0.28
ε_{low}	0.18	0.18	0.18
β	1e-3	1e-3	1e-3
Role out	8	8	8

esis (Zielhypothese), and (ii) classification and hierarchical annotation of all identified errors according to the proposed error typology.

The annotation campaign spanned six months and followed a three-stage protocol to ensure reliability and validity. In Stage 1, each essay was independently annotated by two experts: a graduate student specializing in German linguistics and an experienced university-level German instructor. Following independent annotation, the pair engaged in adjudication sessions to reconcile discrepancies and produce a consensus version. In Stage 2, unresolved disagreements were escalated to a broader annotation team for deliberation until consensus was achieved. Finally, in Stage 3, a random subset of annotations underwent external validation by senior scholars in German linguistics to verify inter-annotator consistency and typological fidelity.

To sustain annotation quality and conceptual alignment throughout the project, the team convened regular calibration meetings—weekly during the initial two months and biweekly thereafter—focusing on edge cases, category boundary disputes, and evolving annotation guidelines.

C Extra Metric

Exact Match (EM), Minimum Edit Distance (MED), and Error Correction Rate (ECR) were adopted as evaluation metrics to assess the similarity between multi-granularity algorithmic annotations and human reference annotations. EM measures the proportion of instances in which the generated output sequence exactly matches the target reference, thereby reflecting precision at the token-level. While EM is a stringent and interpretable metric for short, well-formed sequences, it is overly sensitive to minor discrepancies and fails to account for partial correctness—particularly in tasks involving semantic coherence and long-range dependencies.

MED, on the other hand, quantifies the dissimilarity between two sequences by computing the minimum number of edit operations—specifically insertions, deletions, and substitutions—required to transform the generated sequence into the reference. This metric provides a fine-grained, gradient assessment of output quality and is robust to superficial mismatches that do not affect meaning.

However, in the context of semantic-aware GEC over long-form texts,EM has limitations. EM tends to yield spuriously low scores even when most errors are correctly addressed, as a single deviation (e.g., paraphrasing, synonym substitution, or acceptable reordering) results in a complete mismatch. This makes EM poorly correlated with actual correction effectiveness in semantically complex or structurally flexible contexts.

To address this limitation and better evaluate

model performance on long-text correction, we introduce ECR, defined as the ratio of successfully corrected errors to the total number of annotated errors in the input. As presented in Table 9, the THINKGEC framework consistently outperforms SFT across all linguistic dimensions, demonstrating its effectiveness in GEC.

Table 9: Quantitative evaluation of model performance on sentence-level and article-level correction tasks. EM: Exact Match ratio (%); MED: Minimum Edit Distance (lower is better); ECR: Error Correction Rate. Results are averaged over the test set.

Model	De10K		Sem	
	EM ↑	MED ↓	ECR ↑	MED ↓
Qwen3-8B				
Base	0.3454	6.2829	0.1467	17.0657
+SFT	0.4713	5.0836	0.1143	26.9487
+THINKGEC	0.5037	4.5877	0.1474	7.9484
LLaMA3.1-8B				
Base	0.2931	10.1478	0.1028	17.0639
+SFT	0.4806	4.8142	0.0987	15.4256
+THINKGEC	0.4848	4.3784	0.1268	9.3725

D Experiment Detail

D.1 Instruction Templates

Table 10: Instruction templates for the three stages of the German GEC pipeline.

Stage	Instruction Template
Knowledge Elicitation I	You are a German language expert. Detect the grammatical error span in the input text. {Expert annotation}
Knowledge Elicitation II	You are a German language expert. Identify the grammatical error type within the detected span. {Expert annotation}
Knowledge Injection	You are a German language expert. Identify the error span and classify its type. `input:\n{Source}\n\n response:\n{Target}`
Finding Correct Paradigm	You are a German language expert. Correct the sentence into grammatically well-formed output. `input:\n{Source}\n\n response:\n{Target}`

Table 10 displays the instruction templates for the three stages of THINKGEC. Some templates include an input field (for providing source text) and a response field (for specifying target text).

Figure 5: The response length changes during GRPO training process.

D.2 Implementation Details

Our implementation is built upon the open-source frameworks `LLaMA-Factory` (Zheng et al., 2024) and `verl` (Sheng et al., 2025), leveraging their support for efficient reinforcement learning and parameter-efficient fine-tuning. Due to practical constraints in training time and computational resources, we adopt Low-Rank Adaptation (LoRA) (Hu et al., 2021) during the Knowledge Injection phase, as opposed to full-parameter fine-tuning. This approach significantly reduces trainable parameter count by introducing low-rank decomposition matrices into the transformer attention modules, thereby improving training efficiency while preserving model capacity.

We optimize the training process using the AdamW optimizer (Kingma and Ba, 2017) with a cosine annealing learning rate schedule, which provides stable convergence and mitigates overfitting. All hyperparameters are detailed in Table 8. Experiments are conducted on a cluster equipped with 4 NVIDIA A800 80GB GPUs.

D.3 GRPO Components

During the reinforcement learning phase with GRPO and DAPO in Qwen3-8B, we observed notable dynamics in several training indicators, including generated response length, policy entropy, and the effective number of sampled batches.

Figure 5 illustrates the evolution of response length during the GRPO-based training phase. The generated outputs exhibit a gradual decrease in length over training steps, eventually converging to a stable and linguistically appropriate range. Notably, no pathological expansion or excessively long generations are observed.

This controlled length dynamics suggests that the policy naturally learns to produce concise corrections without requiring explicit length regu-

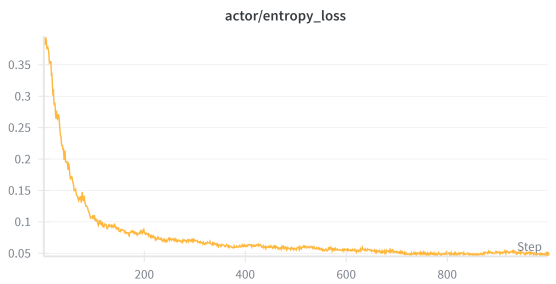

Figure 6: The entropy changes during GRPO training process

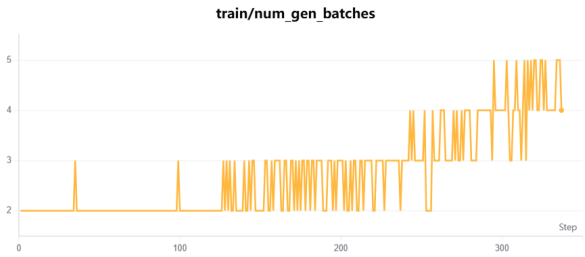

Figure 7: The change of generation batches during GRPO training process

larization. Combined with the absence of training instability, this indicates that token-level optimization—while beneficial in some generation tasks—does not provide necessary advantages in the grammatical error correction setting, where edits are typically minimal and localized.

Figure 6 illustrates the policy entropy dynamics during GRPO training. The entropy decreases rapidly in the initial stages and continues to decline until it approaches zero, indicating severe entropy collapse. This behavior reflects a sharp reduction in output diversity, as the policy becomes increasingly deterministic and overconfident in its token predictions.

In the context of grammatical error correction, such collapse is particularly problematic. It limits the model's ability to explore alternative corrections, which is crucial for handling diverse learner errors. The observed trend highlights the necessity of stabilization mechanisms, such as higher clipping thresholds, to regulate policy updates and maintain sufficient exploration throughout training.

Figure 7 presents the number of batches required to collect dynamic samples with non-extreme rewards during DAPO training. As training progresses, an increasing number of batches are needed to obtain valid samples that fall within the target reward range, reflecting a growing inefficiency in sample filtering. This trend implies a substantial rise in computational overhead for subsequent training stages. Moreover, the aggressive exclusion of samples with reward scores of 0 or 1 leads to the discard of potentially informative instances, including both systematically incorrect outputs that reveal model weaknesses and perfectly corrected instances that reinforce accurate editing patterns. Consequently, this sampling strategy not only increases training cost but also weakens the learning signal by removing critical error and correction exemplars from the update process.

In contrast, we find that retaining only the higher clipping threshold, while reverting to sequence-level updates and standard sampling, yields more stable training and consistently better performance on the GEC task. Therefore, our THINKGEC framework adopts this simplified variant, leveraging the proven benefits of elevated clipping to mitigate entropy collapse without introducing destabilizing components.