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 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, comprising *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 model scale and reinforcement design, crucially revealing the ***complexity-dependent efficacy*** of reasoning trajectories, which benefits structural repairs but proves redundant for surface-level errors. THINKGEC delivers interpretable, pedagogically aligned rationales, advancing both the accuracy and educational value of LLM-based GEC. Our code and dataset are available 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 have evolved through distinct paradigms, primarily dominated by sequence labeling methods like GECToR (Omelianchuk et al., 2020) and encoder-decoder Seq2Seq models (Zhao and Wang, 2020; Ye et al., 2023; Rothe et al., 2021; Zhou et al., 2023; Zhang et al., 2022b; Fang et al., 2023). These systems typically 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, such as GrammarGPT (Fan et al., 2023) and task-decomposition frameworks like CoTask (Liu et al., 2025a) and ExCGEC (Ye et al., 2025) benefit from large-scale pretraining. Yet, despite introducing alignment mechanisms (Liang et al., 2025), 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 Fine-grained Learner 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) **Absence of Interpretable Feedback**: Most existing systems provide fluent rewrites without explicit justifications or linguistic 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 propose 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) We demonstrate THINKGEC's superior performance and analyze model scale and reinforcement design, crucially revealing the complexity-dependent efficacy of reasoning for structural versus surface-level errors.

2 Related Work

A comprehensive review of GEC datasets, traditional methodologies, and LLM-based approaches is detailed in Appendix B.

GEC Datasets. High-quality datasets are foundational to GEC. While English and Chinese benefit from large-scale benchmarks like BEA-19 (Bryant et al., 2019) and MuCGEC (Zhang et al., 2022a), resources for German remain scarce. Existing learner corpora such as Falko (Reznicek et al., 2012) and Merlin (Boyd et al., 2014) primarily focus on surface-level error tags. Crucially, they lack the fine-grained, pedagogically oriented annotations required to support diagnostic reasoning in LLMs.

Methodologies. Traditional GEC approaches evolved through Seq2Edit (Omelianchuk et al., 2020) and Seq2Seq (Rothe et al., 2021) paradigms. Recently, LLMs have introduced a new paradigm leveraging instruction tuning (Fan et al., 2023) and task-decomposition strategies (Liu et al., 2025a). Despite these advancements, most methods still treat correction and reasoning as separate or implicit processes, often leading to over-correction. Unlike reinforcement learning approaches that rely on large-scale preference pairs (Liang et al., 2025), THINKGEC explicitly models the cognitive workflow of identification, description, and explanation, optimizing it via GRPO without requiring massive parallel corpora.

3 Methodology

3.1 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

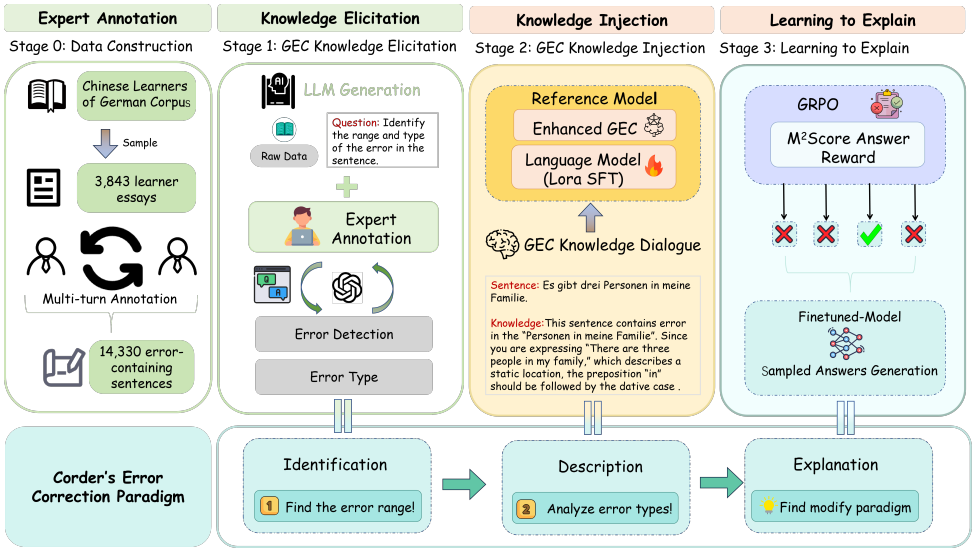

Figure 1: The overall pipeline of THINKGEC, comprising three core components: (1) GEC Knowledge Elicitation (2) GEC Knowledge Injection and (3) Learning to Explain.

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.

Unlike traditional end-to-end models that treat correction as a black-box mapping, our formulation explicitly models the cognitive diagnostic process of a human tutor, ensuring that corrections are not just statistically probable but linguistically justified. Together, these stages form the foundation of our **THINKGEC** framework, and subsequent sections detail how each stage is instantiated.

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

To support the identification and knowledge elicitation stages in our THINKGEC framework, we propose **De10K**. Distinct from existing corpora like Falko (Reznicek et al., 2012) and Merlin (Boyd et al., 2014) that primarily focus on surface-level correctness, De10K stands as the **first German GEC benchmark featuring a multi-layered, pedagogically oriented error typology**. Build upon the Chinese Learners of German Corpus (CDLK)(Li and Wu, 2024), **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. Its fine-grained annotation scheme (spanning 63 sub-categories) provides the precise structural signals requisite for inducing diagnostic reasoning in LLMs, offering value for broader L2 German acquisition research.

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

Error Category	Train	Test	Total	Dist. (%)
Orth	4,242	483	4,725	27.8%
MorSyn	2,472	258	2,730	16.0%
Wortstellung	1,364	163	1,527	9.0%
Valenz	2,949	341	3,290	19.3%
Syn	4,211	529	4,740	27.9%

Table 1: Distribution of Error Types Across Training and Test Sets.

grounds; (3) fine-grained, multi-level annotations, informed by linguistic theory, that provide deep insight into error patterns in L2 German writing.

Table 1 presents the statistics of De10K across different proficiency levels and error categories. The distribution reveals a realistic imbalance, with structural errors constituting a significant portion, thereby posing a greater challenge for syntactic reasoning compared to simple orthographic corrections.

Further details on the dataset and annotation process are provided in Appendix D.

3.3 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 F.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.

3.4 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. Distinct from standard SFT method (Fan et al., 2023) that optimizes solely for the final rewritten text, our paradigm enforces process supervision, compelling the model to internalize the syntactic rules governing the error before executing the edit.

Formally, we construct a training dataset pairing prompt-augmented inputs with target reasoning trajectories consisting of natural language descriptions of the error scope and type. The model is optimized to maximize the conditional likelihood of generating these reasoning steps (formal objectives are detailed in Appendix C), 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, the model learns to condition its generation not only on the input sentence but also on structured diagnostic reasoning, laying the foundation for advanced correction strategies.

3.5 Stage 3: Learning to Explain via GRPO

Prior work (Liu et al., 2025a; Liang et al., 2025) has demonstrated the efficacy of SFT in establishing a basic correction paradigm, particularly in high-resource English settings. However, for low-resource languages like German, relying solely on SFT is insufficient. It restricts the model to passive imitation of limited references, often resulting in overfitting rather than robust syntactic generalization. While reinforcement learning could mitigate this, popular preference optimization methods (e.g., DPO, EPO (Liang et al., 2025)) depend heavily on large-scale, high-quality paired annotations, which are unavailable in the scarce German GEC landscape.

To overcome these limitations, we employ GRPO to leverage the syntactic priors embedded in pre-trained LLMs. Unlike SFT's rigid mimicry or DPO's reliance on extensive paired data, GRPO optimizes policies by contrasting multiple self-generated outputs. This 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 candidate outputs from the current policy, where each output represents a reasoning trajectory or corrected text. The reward advantage for each output is calculated by normalizing the rewards within the sampled group. Specifically, the reward is assigned to the corrected output using the $\mathbf{M}^2$ **Scorer** (Dahlmeier and Ng, 2012), which evaluates **precision**, **recall**, and the $F_{0.5}$**-score** between the original and corrected text pairs.

Following recent GRPO variants (Yu et al., 2025; Liu et al., 2025b), we adopt an optimization objective that integrates an upper-bound clipping strategy with a direct KL penalty to stabilize training (formal definition provided in Appendix C). This objective applies asymmetric clipping bounds to the likelihood ratio to restrict drastic policy updates, while the KL divergence penalty prevents the current policy from deviating excessively from the SFT reference model $M_{\theta'}$.

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.

4 Experiments

4.1 Datasets and Evaluation

We conduct experiments on two datasets to verify both specific efficacy on German and broad generalization across languages.

German De10K. We use the **De10K** corpus constructed in this work, categorizing errors into six major types. We utilize five types for training and reserve the *Semantically-driven Error* category to evaluate the model's **generalization ability** on unseen error patterns. The training split contains 12,897 annotated sentences, and evaluation is performed on a held-out test set of 1,433 sentences.

English Generalization Benchmark. To assess the framework's adaptability to other languages under similar data constraints, we simulate a low-resource setting using the Low Resource Track of the BEA-2019 (Bryant et al., 2019) dataset for training. Performance is reported on the standard CoNLL-2014 (Ng et al., 2014) and BEA-2019 test sets.

Evaluation Metrics. We employ a composite protocol to capture both precision and semantic quality: **(1) $\mathbf{M}^2$ Scorer**: We compute Precision, Recall, and $F_{0.5}$ using the standard MaxMatch algorithm. We prioritize $F_{0.5}$ to better reflect correction quality by penalizing false positives. **(2) LLM-based Judge**: Recognizing that reference-based metrics may penalize valid alternative corrections, we adopt the Semantic-incorporated Evaluation (SEE) framework (Li et al., 2025). Unlike rigid string matching, SEE leverages an LLM to assess semantic correctness, ensuring that reasonable edits absent from gold data are fairly evaluated.

4.2 Base Models and Baselines

Base Models. We evaluate our framework on Qwen3 (Yang et al., 2025) and LLaMA3.1 (Grattafiori et al., 2024). Experiments primarily utilize their 8B variants, with an ablation on Qwen3-4B to analyze scaling effects.

Baselines. To rigorously evaluate THINKGEC, we benchmark against a diverse set of systems ranging from traditional architectures to strong proprietary LLMs. For traditional methods, we include GECToR (Omelianchuk et al., 2020), a robust encoder-based sequence tagging model. For LLM-based systems, we compare against strong proprietary models and open-source instruction-tuning baselines, including Standard SFT and Chain-of-Task (Liu et al., 2025a), a recent framework that also decomposes GEC into sub-tasks. To dissect our framework's contributions, we further examine internal variants such as *Reasoning Ablation* and *Optimizer Substitution*. Regarding the latter, we validate our reinforcement strategy by comparing GRPO against DAPO (Yu et al., 2025). Notably, we do not employ preference-based optimization methods like DPO or EPO (Liang et al., 2025), as standard GEC corpora lack the high-quality chosen-rejected pairs required for these algorithms.

4.3 Main Result

The main results are summarized in Table 3.

Overall Performance. Across all base models and training configurations, the THINKGEC framework consistently outperforms strong baselines

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	$F_{0.5}$
Traditional Method															
GECToR	61.11	26.40	48.39	51.79	23.20	41.55	50.98	24.30	41.80	79.56	63.74	75.80	66.67	37.60	57.74
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
+CoTask	67.32	69.20	67.68	68.55	68.00	68.44	71.26	57.94	68.13	**78.62**	73.10	77.45	**65.98**	**63.60**	**65.49**
+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
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
+CoTask	68.90	70.00	69.12	**71.07**	68.80	70.61	**76.74**	61.68	**73.17**	75.29	74.85	75.21	61.37	57.20	60.49
+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**
Leading LLMs															
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
Gemini3-Flash	59.26	**83.20**	62.88	64.15	**81.60**	67.02	62.18	69.16	63.46	**64.90**	**78.95**	**67.30**	44.94	**64.00**	47.79

Table 2: Performance comparison on the GEC benchmark under avg@5, with error categories following our typology. Best scores are **bolded**; this evaluation protocol and formatting are maintained in all subsequent tables.

in $F_{0.5}$ score, confirming its effectiveness for GEC. Notably, it surpasses traditional methods like GECToR and recent multi-stage instruction tuning frameworks such as CoTask, verifying the superiority of our specific reasoning-enhanced design. 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 families, indicating that the benefits of our framework generalize beyond specific LLM architectures. Moreover, THINKGEC surpasses larger proprietary systems, setting a new benchmark for syntactically aware correction. Despite architectural and scale differences among base LLMs, THINKGEC consistently delivers substantial improvements, underscoring its robustness and adaptability to diverse pretrained backbones.

Fine-grained Analysis. To understand the source of improvements, Table 2 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.

Correction Tendency and Semantic Fidelity. 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. Moreover, this observation is further contextualized by the SEE results on the right of Table 3. Under this semantic-aware metric, THINKGEC not only maintains its lead over SFT but also demonstrates improvements in both Precision and Recall. This contrasts with the M^2 Scorer findings, suggesting that the reasoning-guided model successfully generates valid corrections that are semantically faithful, even if they deviate from the rigid gold reference.

Model	M^2 Scorer			SEE		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Traditional Method						
GECToR	68.03	31.30	55.10	38.82	26.12	35.38
Qwen3-8B						
Base	63.73	45.67	59.06	56.03	51.77	55.13
+SFT	71.57	60.38	69.02	65.76	63.43	65.28
+CoTask	72.10	57.55	68.63	64.38	61.06	63.69
+THINKGEC	**74.85**	**59.60**	**71.21**	**69.52**	**67.57**	**69.12**
LLaMA3.1-8B						
Base	55.03	51.45	54.27	54.58	53.38	54.33
+SFT	72.95	61.40	70.31	63.84	63.84	63.84
+CoTask	73.06	56.88	69.13	63.68	59.38	62.77
+THINKGEC	**75.19**	**61.78**	**72.06**	**66.99**	**63.94**	**66.36**
Leading LLMs						
GPT-4o	**66.84**	63.99	**66.25**	-	-	-
GPT-o1	63.25	67.24	64.01	-	-	-
Gemini3-Flash	62.26	**69.19**	63.53	-	-	-

Table 3: Main Results on the **De10K** dataset Across Different Approaches.

4.4 Generalization Capabilities

We evaluate the generalization capability of THINKGEC from two perspectives: robustness to unseen error patterns within German, and adaptability to English under low-resource constraints.

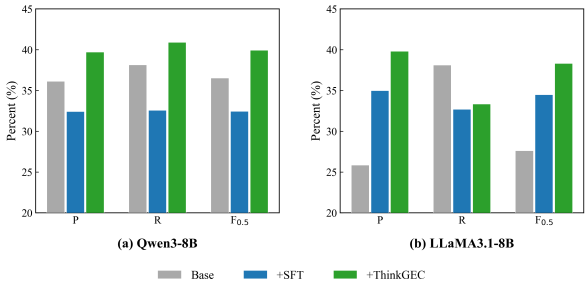

Figure 2: Performance comparison on the *Semantically-driven Error* category.

Generalization to Unseen Error Types. 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 Figure 2. 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.

Model	CoNLL-14			BEA-19		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
Traditional Method						
GECToR	56.11	16.82	38.25	65.61	31.64	54.01
Qwen3-8B						
Base	59.50	50.78	57.52	58.87	64.88	59.98
+SFT	64.83	51.41	61.62	70.10	66.09	69.26
+THINKGEC	**68.33**	**51.91**	**64.27**	**75.01**	**66.91**	**73.24**

Table 4: Cross-lingual results on English benchmarks. THINKGEC achieves superior Precision and $F_{0.5}$ score.

Cross-Lingual Generalization. Given that most real-world GEC scenarios lack massive parallel corpora, we assessed the framework's adaptability by simulating a low-resource setting on English GEC benchmarks. As shown in Table 4, while traditional methods struggle in data-scarce scenarios, THINKGEC consistently outperforms the SFT baseline across both CoNLL-14 and BEA-2019 benchmarks. This improvement, particularly in precision, confirms that the *Identification-Description-Explanation* workflow is language-agnostic. By enforcing explicit reasoning, the framework effectively mitigates over-correction, demonstrating robust generalization to languages with different grammatical rules.

5 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.

5.1 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 5, their impact varies with error complexity. For simple surface-level errors such as Orthography and Morphology, reasoning slightly degrades performance. This aligns with our observation that explicit reasoning for obvious errors triggers unnecessary edits, a form of over-correction arising from excessive logical inference. However, for high-complexity errors involving structural reordering, most notably *Wortstellung*, reasoning provides substantial gains. This contrast suggests that while reasoning trajectories may introduce redundancy for local corrections, they are indispensable for resolving long-range dependencies and structural constraints.

To further verify the quality of these trajectories, we conducted an expert evaluation as shown in Table 6. The results confirm that the generated rationales possess concrete pedagogical value rather than being mere hallucinations or meaningless repetitions. Specifically, the high Correlation scores demonstrate that the model's reasoning is faithful to its corrections, effectively avoiding the disconnect often seen in hallucinated content. Furthermore, the Interpretability scores indicate that the model successfully articulates linguistic rules to justify its edits. Even for abstract categories like *Valenz* and *Wortstellung*, the model provides coherent explanations that help learners understand the underlying

Model	De10k (Overall)			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	$F_{0.5}$	P	R	$F_{0.5}$
Qwen3-8B																		
Base	63.73	45.67	59.06	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	71.57	60.38	69.02	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**
+Rational SFT	71.56	**61.35**	69.25	68.34	70.80	68.82	69.23	72.00	69.77	76.22	65.75	73.87	77.68	77.51	77.65	63.10	62.60	62.99
+THINKGEC	73.01	60.62	70.14	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 w/o rational	**74.85**	59.60	**71.21**	**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

Table 5: **Comprehensive Ablation Study on Reasoning Trajectories.** We compare model performance across the full De10K dataset and specific error categories.

Error Type	Interpretability	Correlation
Orth	3.8	4.6
MorphSyn	3.8	4.4
Wortstellung	3.3	4.1
Valenz	3.1	4.3
Syn	3.9	4.6

Table 6: Human Evaluation of Rationales (5-point scale)

logic behind the correction, validating the educational utility of the THINKGEC framework.

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.

5.2 Impact of Model Scale

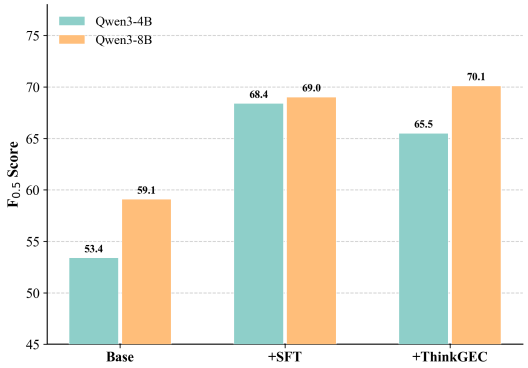

Figure 3: Performance comparison of THINKGEC across models of varying sizes.

Figure 3 illustrates 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 fraDgmented 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 of producing reliable reasoning signals for reinforcement optimization.** Below this threshold, the framework risks amplifying spurious correlations rather than improving syntactic accuracy.

5.3 Impact of RL Components

We further investigate the choice of reinforcement learning algorithms by benchmarking GRPO against its variant, DAPO. Contrary to observations in long-text generation, our experiments reveal that GRPO consistently outperforms DAPO on the De10K dataset. This discrepancy stems from the nature of GEC: the token-level policy gradient optimization in DAPO offers limited gains for short-sequence editing tasks, whereas GRPO's group-based outcome supervision proves more robust for localized corrections. A detailed ablation study, including the analysis of clipping thresholds and convergence dynamics, is provided in Appendix F.3.

6 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 variants with 4B and 8B parameters, focusing on the application of the THINKGEC framework to German grammatical error correction. The **De10K** corpus used in this work consists of essays written by Chinese learners of German as a second language. This restricts the representativeness of learner errors to a single L1 background, potentially limiting the applicability of our approach to speakers of other native languages. As such, the generalizability of the method across larger model scales, additional NLP tasks, and diverse linguistic domains remains to be fully explored. We leave the extension to a broader range of models, tasks, and learner populations 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

Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. The merlin corpus: Learner language and the cefr.

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.

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.

Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan Huang, and Min Zhang. 2023. TemplateGEC: Improving grammatical error correction with detection template. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6878–6892, Toronto, Canada. Association for Computational Linguistics.

Yinghui Li, Shang Qin, Jingheng Ye, Haojing Huang, Yangning Li, Shu-Yu Guo, Libo Qin, Xuming Hu, Wenhao Jiang, Hai-Tao Zheng, and Philip S. Yu. 2025. Rethinking the roles of large language models in Chinese grammatical error correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 553–567, Vienna, Austria. Association for Computational Linguistics.

Yuan Li and Zekun Wu. 2024. Chinesisches deutschlerner-korpus (cdlk). ein umfangreiches korpus mit mehrebenen-annotation und multidimensionalen metadaten. In Marc Kupietz and Thomas Schmidt, editors, *Neue Entwicklungen in der Korpuslandschaft der Germanistik. Beiträge zur IDS-Methodenmesse 2022*, number 11 in Korpuslinguistik und interdisziplinäre Perspektiven auf Sprache | Corpus Linguistics and Interdisciplinary Perspectives on Language | CLIP, pages 223 – 236. Leibniz-Institut für Deutsche Sprache (IDS).

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.

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.

Marc Reznicek, Anke Lüdeling, Cedric Krummes, Franziska Schwantuschke, Maik Walter, Karin Schmidt, and Hagen Hirschmann. 2012. *Das Falko-Handbuch. Korpusaufbau und Annotationen. Version 2.01*.

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.

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.

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.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 180–189.

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.

Jingheng Ye, Shang Qin, Yinghui Li, Xuxin Cheng, Libo Qin, Hai-Tao Zheng, Ying Shen, Peng Xing, Zishan Xu, Guo Cheng, and Wenhao Jiang. 2025. Excgec: A benchmark for edit-wise explainable chinese grammatical error correction. *Preprint*, arXiv:2407.00924.

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.

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 Use of AI Writing Assistants

In accordance with the conference policy on AI writing assistance, we declare that generative AI tools were used solely for the purpose of grammatical polishing, rephrasing for clarity, and stylistic refinement of the manuscript.

B Related Work

B.1 GEC Datasets

High-quality annotated datasets are the cornerstone of GEC research. The field has been predominantly shaped by English benchmarks, ranging from the foundational FCE (Yannakoudakis et al., 2011) and NUCLE (Dahlmeier et al., 2013) to large-scale shared tasks like CoNLL-14 (Ng et al., 2014) and BEA-19 (Bryant et al., 2019). Beyond English, Chinese GEC has seen progress with datasets such as MuCGEC (Zhang et al., 2022a), and FCGEC (Xu et al., 2022), which introduced multi-reference and fine-grained evaluation standards.

In contrast, resources for German remain scarce. Current research relies heavily on learner corpora like Falko (Reznicek et al., 2012) and Merlin (Boyd et al., 2014). While providing basic error tags,

these datasets primarily facilitate surface-level correction. Crucially, they lack the fine-grained, pedagogically oriented annotations required to support diagnostic reasoning in LLMs, a gap that limits the development of explainable, pedagogically grounded GEC systems for German learners.

B.2 Traditional GEC Methods

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

The **Seq2Edit** framework treats GEC as a sequence labeling task, where models like GECToR(Omelianchuk et al., 2020) predict iterative edits to correct errors. Recently, TemplateGEC (Li et al., 2023) introduced template-enhanced generation to better handle structural constraints. While efficient for local errors, these models often struggle with complex syntactic reordering common in German.

In contrast, the **Seq2Seq** paradigm frames GEC as monolingual translation, employing encoder-decoder architectures to generate corrected sentences end-to-end (Rothe et al., 2021; Ye et al., 2023). Recent advancements have further optimized this framework by incorporating auxiliary syntactic features or error detection signals to enhance performance (Zhang et al., 2022b; Fang et al., 2023). However, these models are inherently data-intensive, relying heavily on massive parallel corpora to learn mapping patterns. This dependency poses a significant bottleneck for low-resource languages like German, where large-scale annotated training data is scarce.

B.3 LLM-based GEC Approaches

LLMs have introduced a new paradigm for GEC. They leverage pre-training and instruction tuning to achieve superior performance. Early works focused on SFT to adapt open-source models for correction tasks. For example, GrammarGPT (Fan et al., 2023) maps erroneous sentences to corrected forms.

To move beyond simple rewriting, recent research has integrated *explicit reasoning* and *alignment* mechanisms. For instance, the Chain-of-Task (CoTask) framework (Liu et al., 2025a) mitigates over-correction by decomposing the process into sequential sub-tasks, while ExCGEC (Ye et al., 2025) explores explainable GEC benchmarks to enhance interpretability. On the alignment front, reinforcement learning has been employed to capture human editing preferences; notably, Edit-Wise Preference Optimization (EPO) (Liang et al., 2025) refines generation granularity to the token level.

Despite these advancements, most approaches still treat correction and reasoning as separate or implicit processes. In contrast, THINKGEC explicitly formalizes the cognitive workflow. It includes *Identification, Description, and Explanation*. We optimize this process via GRPO. This approach bridges the gap between pedagogical reasoning and high-precision correction.

C Formal Optimization Objectives

In this section, we provide the detailed mathematical formulations for the supervised fine-tuning (SFT) and reinforcement learning (GRPO) stages of the THINKGEC framework.

C.1 Stage 2: SFT Objective

Given a pretrained LLM M_θ parameterized by θ, we construct a dialogue-based training dataset $\mathcal{D}_{\text{SFT}} = \{(x'^{(i)}, t^{(i)})\}_{i=1}^{M}$. Here, $x'^{(i)} = p \circ x^{(i)}$ denotes the input sentence $x^{(i)}$ augmented with a specific instruction prompt p, and $t^{(i)}$ represents the target reasoning trajectory (comprising error scope and type description).

The model is optimized to maximize the conditional likelihood of generating the target reasoning chain t given the input x'. The loss function is defined as the standard cross-entropy loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x',t)\sim\mathcal{D}_{\text{SFT}}}\Big[\log p_\theta(t \mid x')\Big]. \quad (1)$$

Minimizing this loss yields the fine-tuned model $M_{\theta'}$, which serves as the reference policy (π_{ref}) for the subsequent reinforcement learning stage.

C.2 Stage 3: GRPO Objective

To further refine the model's ability to generate high-quality corrections and rationales, we employ Group Relative Policy Optimization (GRPO). Unlike standard PPO which requires a separate value network, GRPO estimates the baseline from group averages.

For each prompt-augmented input x', we sample a group of G outputs $\{y_1, y_2, \ldots, y_G\}$ from the current policy $\pi_\theta(\cdot|x')$. We then compute the reward r_i for each output y_i using the M^2 Scorer (checking $F_{0.5}$ against the gold reference).

Advantage Computation. To reduce variance, the advantage A_i for the i-th output is calculated by normalizing the rewards within the group:

$$A_i = \frac{r_i - \mu_r}{\sigma_r}, \quad (2)$$

where μ_r and σ_r are the mean and standard deviation of the rewards $\{r_1, \ldots, r_G\}$ within the sampled group.

Optimization Objective. The GRPO objective incorporates a clipping mechanism to stabilize training and a KL-divergence penalty to prevents the model from deviating too far from the SFT reference policy. The objective is formulated as:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{x' \sim \mathcal{D}, \{y_i\} \sim \pi_\theta} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \Bigg(\min \Big(\rho_i A_i, \text{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon) A_i \Big) - \beta \cdot \mathbb{D}_{\text{KL}} \big(\pi_\theta(\cdot|x') \parallel \pi_{\text{ref}}(\cdot|x') \big) \Bigg) \Bigg], \tag{3}$$

where:

- $\rho_i = \frac{\pi_\theta(y_i|x')}{\pi_{\text{old}}(y_i|x')}$ is the probability ratio between the current and old policies.
- ε is the clipping parameter (e.g., 0.2) that constrains the policy update step.
- β is the coefficient controlling the strength of the KL-divergence penalty.

D Dataset and Annotation Details

D.1 Dataset

We conduct a statistical analysis of our corpus, including sentence length distribution, and present illustrative examples.

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

Illustrative Examples To illustrate the types of errors present in the **De10K** corpus, we present a selection of representative examples along with their corrected versions and error categories in Table 8.

D.2 Error Typology Development

To develop a learner-specific error typology, we conducted two pilot annotation studies following the framework proposed by Nolda et al. (2023).

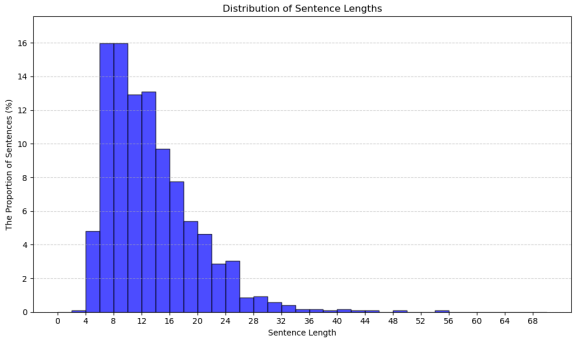

Figure 4: The length distribution of sentences in the whole corpus.

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.

D.3 Annotation Environment and Principles

Error annotation was carried out using the EXMARaLDA annotation environment (Figure 5). The process comprised two interrelated tasks: (i) correction of learner errors in the original text to produce a linguistically well-formed target hypothesis (Zielhypothese), and (ii) classification and hierarchical annotation of all identified errors according to the proposed error typology.

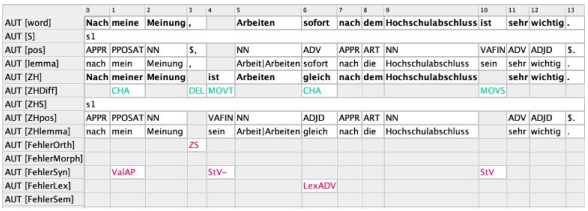

Figure 5: The interface of EXMARaLDA used for De10K annotation.

Crucially, the annotation process adhered to two core principles to ensure standard-compliant yet learner-centric corrections:

- **Fidelity**: Ensuring minimal intervention, preserving the learner's intended meaning while correcting only what was necessary for gram-

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

Table 7: Hyperparameter settings in our experiments.

Origin	Target	Type
In meiner Familian, Vater ist sehr wichtig.	In meiner Familie ist mein Vater sehr wichtig.	Orth,Wortstellung
Er hat schwarz Haare und schwarz Augen.	Er hat schwarze Haare und schwarze Augen.	MorSyn
Er ist einen lehrer.	Er ist Lehrer.	Orth,Valenz
Er mag chillen und lernen sehr.	Er mag Chillen und Lernen sehr.	Valenz
Sie mag tanzen und auch lernen.	Sie mag Tanzen und auch Lernen.	Valenz
Sie ist eine lehrerin.	Sie ist Lehrerin.	Orth,Valenz
Meine Oma kocht für uns. Opa hilft sie.	Meine Oma kocht für uns, mein Opa hilft ihr.	Orth,MorSyn
Oh, heiße Anna.	Oh, heiße Anna.	Orth
Ich finde, Deutschlernen ist sehr wichtig.	Ich finde, dass Deutschlernen ist sehr wichtig.	Syn
Das ist meine Familian.	Das ist meine Familie.	Orth

Table 8: Illustrative examples from the **De10K** corpus showing original learner sentences, corrected versions, and corresponding error types.

maticality and orthographic accuracy in standard German (Ellis, 1994).

- **Consistency**: Maintained through standardized labeling protocols and detailed annotation guidelines to ensure uniform treatment of linguistic phenomena across the corpus.

D.4 Workflow and Quality Assurance

The annotation campaign spanned six months and followed a rigorous three-stage protocol to ensure reliability and validity:

Stage 1: Paired Annotation. 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.

Stage 2: Team Deliberation. Unresolved disagreements were escalated to a broader annotation team for deliberation until consensus was achieved.

Stage 3: External Validation. Finally, a random subset of annotations underwent external validation by senior scholars in German linguistics to verify inter-annotator consistency and typological fidelity.

We maintained annotation quality and consistency through regular calibration meetings, held weekly during the first two months and biweekly thereafter. These sessions prioritized the discussion of edge cases, category boundaries, and necessary updates to the guidelines.

E 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.

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

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.

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

F Experiment Detail

F.1 Instruction Templates

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).

F.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 7.

Experiments are conducted on a cluster equipped with 4 NVIDIA A800 80GB GPUs.

F.3 Detailed Analysis of RL Components

In this section, we provide a deeper analysis of the contribution of individual components in GRPO and its variant DAPO within the THINKGEC framework.

Performance Comparison. As illustrated in Figure 6, DAPO consistently underperforms GRPO. This finding indicates that DAPO's additional modifications, though effective in long-sequence reasoning, do not transfer well to the specific constraints of GEC.

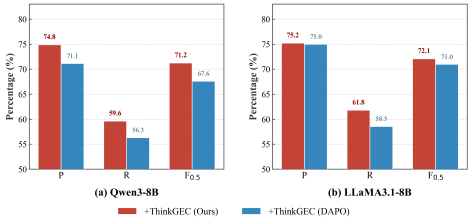

Figure 6: Ablation study of GRPO components on the **De10K** dataset. GRPO shows superior efficiency compared to DAPO in the GEC setting.

Analysis of Discrepancy. We identify two primary factors accounting for this performance gap:

- **Clipping Thresholds:** 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.
- **Gradient Optimization Level:** The token-level policy gradient optimization in DAPO offers limited gains for short-sequence editing tasks. Since GEC typically involves minimal token changes compared to the input, intermediate gradient updates contribute little beyond sequence-level optimization.

Collectively, these findings underscore that **algorithmic advances in reinforcement learning must be carefully contextualized**; mechanisms beneficial for general open-ended text generation may be suboptimal for localized, low-entropy tasks such as GEC.

F.4 GRPO Dynamics

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.

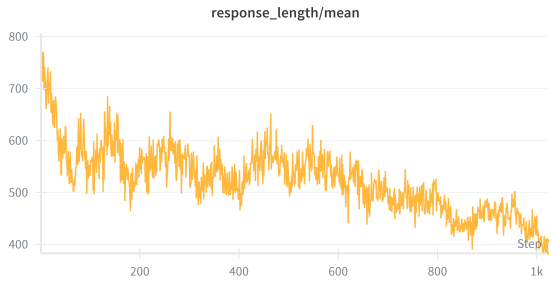

Figure 7: The response length changes during GRPO training process.

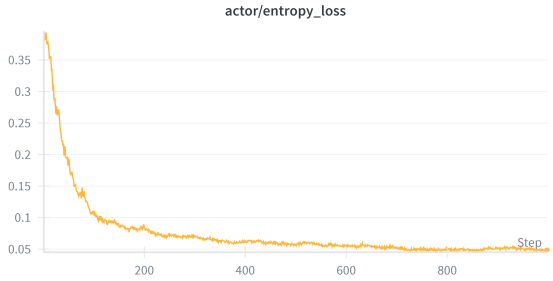

Figure 8: The entropy changes during GRPO training process

Figure 7 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 regularization. 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 8 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 correc-

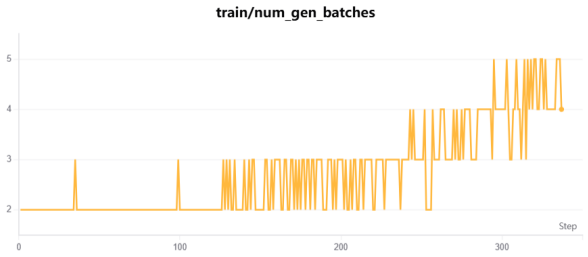

Figure 9: The change of generation batches during GRPO training process

tions, 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 9 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.