

# Improving Low-resource Grammatical Error Correction Via Automatic Linguistic Neurons Probing and Optimizing

Zihao Yu Ying Li\* Zhengtao Yu Shengxiang Gao Cunli Mao  
Yuxin Huang Hua Lai Wenjun Wang

Yunnan Provincial Key Laboratory of Artificial Intelligence Faculty of Information Engineering and Automation Kunming University of Science and Technology, China  
1141327579@qq.com yingli\_hlt@foxmail.com ztyu@hotmail.com  
gaoshengxiang.yn@foxmail.com maocunli@163.com huangyuxin2004@163.com  
405904235@qq.com 175360805@qq.com

## Abstract

Large Language Models (LLMs) exhibit strong language understanding and instruction-following abilities, making them promising backbones for grammatical error correction (GEC). However, prompt-only correction for low-resource languages still suffers from the over-correction challenge, since LLMs cannot fully understand their internal grammar. To address this issue, we propose an automatic linguistic neuron probing and optimizing approach to improve the low-resource GEC with only a few neuron activations. Concretely, we first utilize multiple low-resource GEC corpora to observe the changes in neurons, thus obtaining a linguistically specific probe matrix to ensure whether the neuron is activated. Then, we construct an extra weight matrix to determine the activation intensity of each neuron. Finally, we combine the probe matrix and the weight matrix to obtain an activation matrix, which is injected into selected model layers during inference to guide the frozen LLM toward more controllable GEC. Experiments on four benchmark datasets show that our approach consistently improves GEC performance, indicating that lightweight sparse activation rescaling is extremely useful for low-resource grammatical knowledge understanding. Detailed analysis demonstrates that our approach can further provide interpretable neuron-level control without large-scale parameter fine-tuning.

## 1 Introduction

Low-resource grammatical error correction (GEC) refers to correcting grammatical, spelling, and lexical errors across low-resource languages while preserving the intended meaning of the input text. As shown in Figure 1, the GEC model accurately identifies and corrects tense errors and word order errors in a Vietnamese input sentence. Currently,

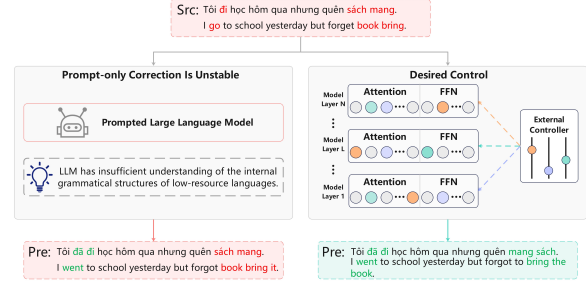


Figure 1: An example of prompt-only low-resource GEC, where the prompted LLM corrects part of the erroneous sentence but still makes unstable edits and leaves some errors unresolved. The contents of the dashed boxes indicate the contrast between prompt-only correction and controlled correction.

GEC results are widely involved in natural language processing applications, i.e., writing assistance, machine translation post-editing, and multilingual text generation (Omelianchuk et al., 2020; Chollampatt et al., 2020; Xue et al., 2021; NLLB Team et al., 2022).

Recently, researchers have increasingly focused on using Large Language Models (LLMs) for GEC, since LLMs possess strong language understanding ability and can follow natural language instructions to correct common grammatical, lexical, and fluency errors. Existing LLM-based GEC methods can be broadly divided into two categories: prompt engineering and efficient parameter fine-tuning. Prompt engineering offers a more flexible paradigm for LLM-based GEC. Instead of training a separate correction model, a general-purpose LLM can be instructed to correct errors under zero-shot or few-shot settings. By designing task instructions, demonstrations, and output constraints, prompting methods can guide LLMs to perform conservative or fluency-oriented correction (Loem et al., 2023; Katinskaia and Yangarber, 2024; Zeng et al., 2024). However, prompt-only correction is highly sensitive to instruction wording, decod-

\*Corresponding author.

ing behavior, and language-specific pre-training patterns, which may over-correct, especially in low-resource settings where the model has insufficient ability to understand these languages. Efficient parameter fine-tuning further adapts LLMs to GEC with lower training cost than full fine-tuning. Methods such as adapters, prefix tuning, prompt tuning, LoRA, and QLoRA introduce lightweight trainable parameters while keeping most backbone parameters unchanged (Houlsby et al., 2019; Li and Liang, 2021; Lester et al., 2021; Hu et al., 2022; Dettmers et al., 2023). These methods improve task adaptation and reduce computational overhead, making them suitable for low-resource GEC scenarios. Nevertheless, they mainly optimize external modules or low-rank parameter updates, without explicitly identifying which internal neurons are responsible for grammatical correction. Therefore, controlling the internal linguistic neurons of LLMs for more stable low-resource GEC remains an important challenge.

Although existing methods have achieved remarkable progress in high-resource GEC settings, the proportion of low-resource language data in LLM pre-training limits the models’ ability to capture their internal grammatical structures. As illustrated in Figure 1, simply injecting correction instructions into an LLM can correct the word “go” into “went”, but it may still leave the word order errors. Therefore, how to promote the adaptation ability of LLMs to low-resource languages through controllable parameter or activation updates remains a major challenge. Recent studies on neuron probing, model editing, and activation steering suggest that transformer components can encode factual, linguistic, or behavioral information and can be modified locally (Geva et al., 2021; Dai et al., 2022; Gurnee et al., 2023; Meng et al., 2022, 2023; Turner et al., 2023; Zou et al., 2023). Inspired by these studies, we explore whether low-resource GEC can be improved by detecting and optimizing linguistic neurons inside LLMs.

Following this motivation, we propose automatic linguistic neuron probing and optimizing, a sparse linguistic neuron control framework for low-resource GEC. First, we feed erroneous and corrected sentence pairs into the same frozen LLM and contrast their internal representations, combining the difference between the corrected representation and the incorrect representation with the gradient signal in the GEC objective function to locate neurons that are sensitive to grammati-

cal correction. Then, we estimate the correction importance of these neurons with the GEC objective and learn lightweight external weights while keeping the backbone LLM frozen, so that each selected neuron obtains a correction-specific intervention strength. Finally, the learned controller is injected into selected model layers during inference to activate corrected-form neurons and suppress error-associated neurons dynamically. Experiments on benchmark datasets demonstrate that our method consistently improves both the original LLM and its LoRA-finetuned variant, suggesting that sparse neuron-level control can serve as an effective lightweight enhancement for low-resource GEC. Further analysis shows that our method is sensitive to neuron selection ratio, target module choice, and intervention direction. The learned masks reveal structured layer-wise and module-wise linguistic neuron distributions.

## 2 Related Works

### Low-resource grammatical error correction.

Low-resource grammatical error correction aims to correct grammatical, spelling, lexical, and fluency errors across low-resource languages while preserving the original meaning. Early neural approaches formulate GEC as source-to-target generation or edit-based tagging, where erroneous sentences are rewritten or transformed into corrected forms (Yuan and Briscoe, 2016; Junczys-Dowmunt et al., 2018; Omelianchuk et al., 2020). Because annotated correction corpora are expensive to construct, prior work expands training resources through artificial error generation, pseudo-parallel data, minimally augmented examples, and back-translation (Grundkiewicz and Junczys-Dowmunt, 2019; Kiyono et al., 2019; Koyama et al., 2021). Recent multilingual pre-training and translation models further support cross-lingual correction with shared representations and multilingual generation ability (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021; NLLB Team et al., 2022; Rothe et al., 2021; Luhtaru et al., 2024). These studies show that GEC can be transferred across languages, but they mainly rely on external data construction or task-specific architectures, leaving the internal grammar-related behavior of LLMs less explored.

**LLMs for low-resource GEC.** The emergence of LLMs shifts GEC from training specialized correction models to prompting or adapting general-

purpose models. Recent studies show that LLMs can produce fluent corrections under zero-shot, few-shot, and instruction-based settings, but they also suffer from over-correction, format instability, and semantic drift (Loem et al., 2023; Katin-skaia and Yangarber, 2024; Zeng et al., 2024). LLMs have also been explored as GEC evaluators (Kobayashi et al., 2024; Seminck et al., 2025). Meanwhile, language-family or parameter-efficient adaptation can improve transfer without updating the whole model (Chronopoulou et al., 2023; Houlsby et al., 2019; Li and Liang, 2021; Lester et al., 2021; Hu et al., 2022; Dettmers et al., 2023). However, these methods still rely on prompt design or external trainable parameters and do not explicitly reveal which internal components drive GEC across languages.

**Neuron probing and sparse activation optimization.** Recent work suggests that transformer components can encode factual, linguistic, and behavioral information in localized neurons, layers, or activation directions (Geva et al., 2021, 2022; Dai et al., 2022; Gurnee et al., 2023; Bills et al., 2023). Model editing and activation steering further show that targeted internal interventions can modify model behavior without full parameter updates (Meng et al., 2022, 2023; Turner et al., 2023; Zou et al., 2023). In parallel, sparse and parameter-efficient tuning methods exploit layer-wise or module-wise importance to reduce adaptation cost (Yao et al., 2024; Qing et al., 2024; Chekalina et al., 2024). Inspired by these works, we focus on low-resource GEC and probe linguistic neurons by contrasting erroneous and corrected sentence activations. Different from PEFT methods that add trainable modules to the backbone, our framework keeps the LLM frozen, learns sparse external mask weights from the GEC objective, and dynamically activates or suppresses selected neurons during correction.

### 3 Our Approach

Motivated by the instability of prompt-only correction, we propose a sparse linguistic neuron control framework for low-resource GEC. It identifies linguistic neurons from a frozen LLM’s internal activations and dynamically activates or suppresses them during correction. Specifically, we feed erroneous and corrected sentences into the LLM to record contrastive representations, then compute their difference and combine it with gradient analy-

sis to construct a probe matrix to detect which neurons are activated. With the LLM frozen, we construct a weight matrix to determine the activation intensity of each neuron via GEC loss between predicted and gold corrections. Finally, we combine the probe matrix and weight matrix into an activation matrix, which is injected into selected layers for optimization to guide final correction. Figure 2 illustrates the two phases of our framework: linguistic neurons probing and analysis, and linguistic neurons dynamic activation and optimization.

#### 3.1 Linguistic Neurons Probing and Analysis

**Linguistic Neurons Probing.** Prompt-only LLM correction is often unstable in low-resource GEC: the model may under-correct, over-correct, or unintentionally change the original meaning. To identify neurons associated with GEC, we feed each erroneous sentence  $x$  and its corrected counterpart  $y$  into the same frozen LLM under the same instruction template. This paired design controls for the prompt condition and allows the representation difference between  $x$  and  $y$  to highlight correction-related activation patterns.

Let  $F_\theta$  denote the frozen LLM. For layer  $\ell$ , submodule  $s$ , neuron dimension  $j$ , and token position  $t$ , we denote the internal representation as  $h_{\ell,s,j}^{(t)}$ . We probe both the attention and MLP projection modules:

$$s \in \{q, k, v, o, \text{gate}, \text{up}, \text{down}\}, \quad (1)$$

where  $q, k, v, o$  denote attention projections, and gate, up, and down denote MLP projections.

For an input sentence  $z \in \{x, y\}$ , we aggregate token-level representations into a sentence-level neuron response:

$$A_{\ell,s,j}(z) = \frac{1}{|z|} \sum_{t=1}^{|z|} h_{\ell,s,j}^{(t)}(z). \quad (2)$$

We then compute the corrected-minus-erroneous representation difference over the probing subset  $\mathcal{D}_p$ :

$$D_{\ell,s,j} = \frac{1}{|\mathcal{D}_p|} \sum_{(x,y) \in \mathcal{D}_p} [A_{\ell,s,j}(y) - A_{\ell,s,j}(x)]. \quad (3)$$

A positive value of  $D_{\ell,s,j}$  indicates that the neuron is more active for corrected sentences, while a negative value indicates stronger activation for erroneous sentences. We refer to them as positive and negative correction-related neurons, respectively.

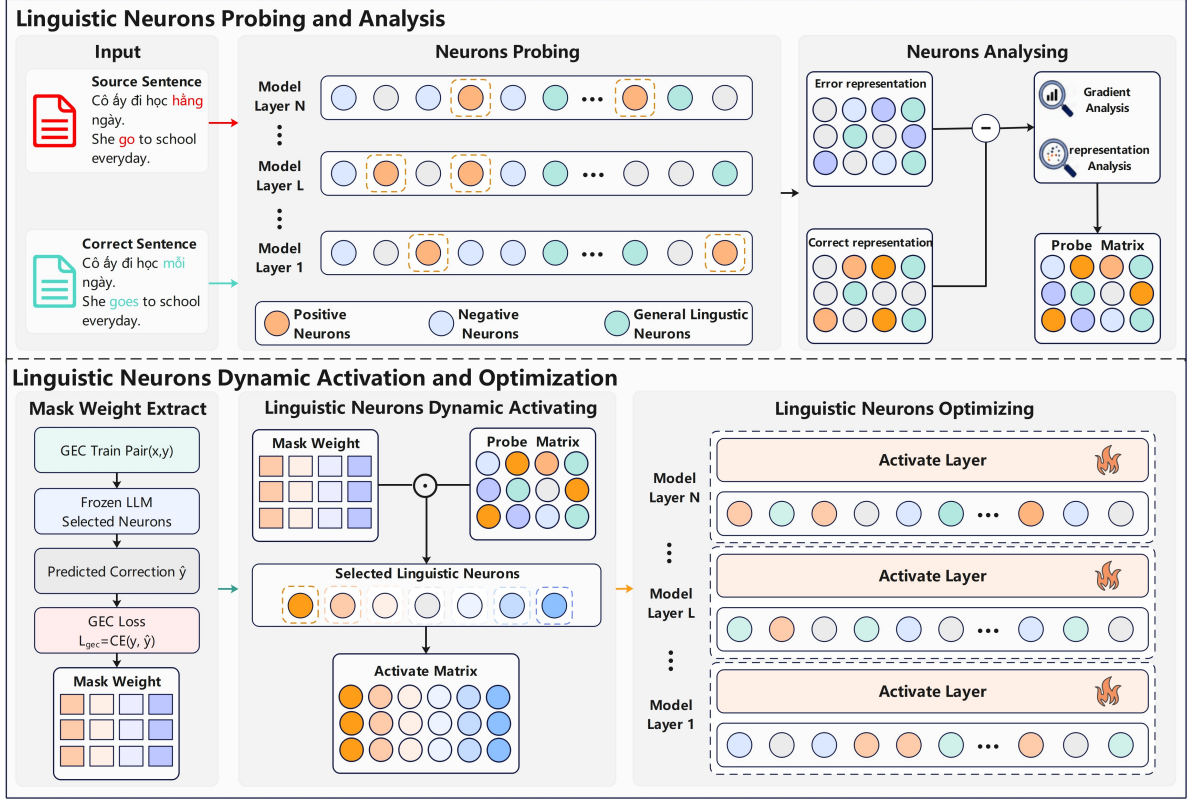


Figure 2: The overall framework of our approach. The upper part illustrates linguistic neurons probing and analysis, where erroneous and corrected sentences are fed into the frozen LLM to obtain error representations, correct representations, gradient signals, and the probe matrix. The lower part illustrates linguistic neurons’ dynamic activation and optimization, where mask weights are extracted from the GEC loss and combined with the probe matrix to produce the activation matrix for layer-wise neuron intervention.

**Linguistic Neurons Analysing.** Representation difference alone only shows contrastiveness, but does not indicate whether a neuron is useful for correction. Therefore, we further estimate each neuron’s sensitivity to the GEC objective. For a training pair  $(x, y)$ , the GEC loss is:

$$\mathcal{L}_{\text{gec}} = - \sum_t \log P(y_t | x, y_{<t}). \quad (4)$$

The backbone parameters are not updated in this step. We only use gradients to estimate neuron sensitivity:

$$G_{\ell,s,j} = \|\nabla_{\theta_{\ell,s,j}} \mathcal{L}_{\text{gec}}\|_1, \quad (5)$$

where  $\theta_{\ell,s,j}$  denotes the parameter slice corresponding to the output dimension  $j$  of submodule  $s$  in layer  $\ell$ .

We select linguistic neurons using a coverage-based strategy. For each candidate neuron,  $|D_{\ell,s,j}|$  measures the relative representation difference between corrected and erroneous sentences. We sort

all candidates by this score and select the smallest  $k$  satisfying

$$\frac{\sum_{i=1}^k |D|_{(i)}}{\sum_{i=1}^N |D|_{(i)}} \geq \tau, \quad (6)$$

where  $|D|_{(i)}$  is the  $i$ -th largest score and  $N$  is the number of candidate neurons. Here,  $\tau$  controls the cumulative contrastive score covered by the selected neurons. The resulting  $k$  is further constrained by

$$\lceil r_{\min} N \rceil \leq k \leq \lfloor r_{\max} N \rfloor, \quad (7)$$

which prevents overly sparse or dense masks. Thus,  $\tau$  determines selection according to representation differences, while  $r_{\min}$  and  $r_{\max}$  provide practical bounds on the number of selected neurons.

The selected neuron set is denoted by  $\mathcal{S}$ , and the probe matrix is defined as

$$P_{\ell,s,j} = \begin{cases} \text{sign}(D_{\ell,s,j}) G_{\ell,s,j}, & (\ell, s, j) \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Here,  $\text{sign}(D)$  determines whether a neuron should be enhanced or suppressed, while  $G$  measures the sensitivity of the GEC loss to that neuron. We use  $|D|$  only as a relative selection score rather than an intervention strength, because its raw magnitude can vary with layer scale, module type, sequence length, and tokenization. Thus,  $P$  records three pieces of information: selected neuron positions, intervention directions, and gradient-based correction sensitivity. Positive entries correspond to corrected-form neurons to be enhanced, while negative entries correspond to error-associated neurons to be suppressed.

### 3.2 Linguistic Neurons Dynamic Activation and Optimization

**Mask Weight Extraction.** After obtaining the probe matrix, we learn how strongly each selected linguistic neuron should be modified. Given a training pair  $(x, y)$ , the frozen LLM produces a predicted correction  $\hat{y}$  under the current neuron mask, and the loss is computed as:

$$\mathcal{L}_{\text{gec}} = \text{CE}(y, \hat{y}). \quad (9)$$

Only the external mask weight matrix  $W$  is optimized:

$$W \leftarrow W - \eta \nabla_W \mathcal{L}_{\text{gec}}. \quad (10)$$

Here, each element in  $W$  corresponds to a selected linguistic neuron. This step learns correction-specific intervention strengths while keeping the backbone LLM frozen.

**Linguistic Neurons Dynamic Activating.** The dynamic activating module combines the probe matrix  $P$  and the learned mask weights  $W$ . The activation matrix is computed as:

$$M = P \odot W, \quad (11)$$

where  $\odot$  denotes element-wise multiplication. For each selected neuron, we rescale its representation as:

$$\tilde{h}_{\ell,s,j}^{(t)} = h_{\ell,s,j}^{(t)} \cdot \max(0, 1 + \alpha M_{\ell,s,j}). \quad (12)$$

Here,  $\alpha$  controls the global intervention strength. Positive values in  $M$  enhance corrected-form neurons, negative values suppress error-associated neurons, and zero values leave neurons unchanged.

**Linguistic Neurons Optimizing.** Finally, we inject the activation matrix into selected model layers through layer-wise activation hooks. For each layer and submodule, only selected dimensions are

modified, while all other dimensions remain unchanged:

$$\tilde{\mathbf{h}}_{\ell,s}^{(t)} = \mathbf{h}_{\ell,s}^{(t)} \odot (1 + \mathbf{m}_{\ell,s}), \quad (13)$$

where  $\mathbf{m}_{\ell,s}$  is derived from  $M$  and is zero on unselected dimensions. The modified representations are then passed through the remaining transformer computation to generate the final corrected sentence. Since the backbone LLM remains frozen, we perform sparse activation-level optimization rather than parameter-level fine-tuning, providing an interpretable and lightweight control interface for low-resource GEC.

| Dataset    | Train   | Test  | All     |
|------------|---------|-------|---------|
| Vietnamese | 50,000  | 2,265 | 52,265  |
| Burmese    | 50,000  | 5,000 | 55,000  |
| Lao        | 50,000  | 4,250 | 54,250  |
| Khmer      | 50,000  | 2,800 | 52,800  |
| MuCGEC     | 250,000 | 6,000 | 256,000 |
| FCGEC      | 36,341  | 3,000 | 39,341  |

Table 1: Dataset statistics by sentence number.

## 4 Experiments

### 4.1 Experimental Setups

**Datasets.** We conduct experiments on four low-resource languages: Lao, Khmer, Vietnamese, and Burmese. Since publicly available parallel GEC corpora are scarce for these languages, we use ASR-generated GEC corpora as training data. These corpora contain naturally occurring recognition-induced errors, including grammatical, lexical, spelling, and phonological errors, making them a practical proxy for low-resource GEC evaluation. The dataset statistics are reported in Table 1. In addition, we include Chinese as a higher-resource comparison setting. Specifically, we sample 250,000 training sentences from MuCGEC and evaluate the model with the standard MuCGEC/ChERRANT protocol. Subsequently, as the official evaluation channel of MuCGEC has been closed, we additionally engaged FCGEC to evaluate our method.

**Evaluation.** We use  $F_{0.5}$  as the main evaluation metric for all languages (Ng et al., 2013). Following standard GEC evaluation practice,  $F_{0.5}$  assigns more weight to precision than recall, as unnecessary edits may alter sentence meaning or reduce fluency. For Chinese, we conduct evaluation

| Model              | vi           |              |              | my           |              |              | lao          |             |              | kh           |              |              | avg.         |              |              |
|--------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                    | P            | R            | $F_{0.5}$    | P            | R            | $F_{0.5}$    | P            | R           | $F_{0.5}$    | P            | R            | $F_{0.5}$    | P            | R            | $F_{0.5}$    |
| <b>Llama3.1-8B</b> |              |              |              |              |              |              |              |             |              |              |              |              |              |              |              |
| Base               | 19.66        | 22.99        | 20.25        | 4.43         | 4.36         | 4.42         | 1.63         | 2.33        | 1.73         | 1.34         | 1.55         | 1.38         | 6.76         | 7.81         | 6.95         |
| Base*              | 28.33        | <b>24.30</b> | 27.42        | 6.94         | 1.83         | 4.46         | 12.66        | <b>7.95</b> | <b>11.32</b> | 6.27         | 7.91         | 6.54         | 13.55        | 10.50        | 12.44        |
| LoRA               | 26.05        | 12.13        | 21.19        | 13.24        | <b>12.40</b> | 13.06        | <b>15.17</b> | 4.25        | 10.02        | 13.12        | 10.80        | 12.58        | 16.89        | 9.90         | 14.21        |
| LoRA*              | <b>37.41</b> | 23.93        | <b>33.63</b> | <b>20.76</b> | 10.48        | <b>17.36</b> | 11.60        | 7.11        | 10.30        | <b>15.92</b> | <b>13.68</b> | <b>15.42</b> | <b>21.42</b> | <b>13.80</b> | <b>19.18</b> |
| <b>Qwen3-8B</b>    |              |              |              |              |              |              |              |             |              |              |              |              |              |              |              |
| Base               | 39.32        | 5.08         | 16.74        | 2.78         | 2.24         | 2.65         | 0.73         | 0.96        | 0.77         | 0.93         | 1.58         | 1.02         | 10.94        | 2.46         | 5.29         |
| Base*              | 16.52        | 28.24        | 18.01        | 4.77         | 4.48         | 4.71         | 9.43         | 2.92        | 6.52         | 13.34        | 4.75         | 9.80         | 11.02        | 10.10        | 9.76         |
| LoRA               | 34.83        | <b>29.21</b> | 33.54        | 11.99        | <b>12.38</b> | 12.06        | 8.97         | <b>4.87</b> | 7.68         | 9.03         | <b>8.58</b>  | 8.94         | 16.20        | <b>13.76</b> | 15.55        |
| LoRA*              | <b>43.93</b> | 25.08        | <b>38.19</b> | <b>19.61</b> | 11.93        | <b>17.38</b> | <b>10.51</b> | 4.31        | <b>8.16</b>  | <b>19.31</b> | 7.66         | <b>14.80</b> | <b>23.34</b> | 12.24        | <b>19.63</b> |
| <b>Sailor2-8B</b>  |              |              |              |              |              |              |              |             |              |              |              |              |              |              |              |
| Base               | 11.46        | 27.19        | 12.96        | 8.99         | 7.97         | 8.79         | 4.55         | 0.69        | 2.14         | 2.74         | 0.31         | 1.06         | 6.94         | 9.04         | 6.24         |
| Base*              | 30.24        | <b>32.97</b> | 30.75        | 13.69        | 6.93         | 11.45        | 7.83         | 7.85        | 7.83         | 14.27        | 6.99         | 11.81        | 16.51        | 13.69        | 15.46        |
| LoRA               | 53.87        | 29.95        | 46.16        | 29.15        | 21.85        | 27.33        | <b>15.51</b> | 4.78        | 10.70        | 19.80        | 14.19        | 18.35        | 29.58        | 17.69        | 25.64        |
| LoRA*              | <b>54.83</b> | 30.52        | <b>47.30</b> | <b>32.55</b> | <b>24.52</b> | <b>30.55</b> | 14.48        | <b>8.66</b> | <b>12.76</b> | <b>21.24</b> | <b>16.95</b> | <b>20.22</b> | <b>30.77</b> | <b>20.16</b> | <b>27.71</b> |

Table 2: Main results of four languages on the test dataset. The superscript \* denotes applying our approach to the corresponding model.

on the MuCGEC and FCGEC test sets. We use the ChERRANT scorer for MuCGEC sets (Zhang et al., 2022) and the official FCGEC evaluation script for FCGEC sets (Xu et al., 2022).

**Hyper-parameter choices.** For LoRA fine-tuning, we use AdamW with a learning rate of  $2 \times 10^{-5}$ , cosine decay, 500 warm-up steps, a gradient accumulation step of 8, and a maximum sequence length of 512. For activation-mask construction, we set  $\tau = 0.85$  and  $r_{\min} = 0.01$ , and select  $r_{\max}$  from  $\{0.05, 0.1, 0.2, 0.3\}$  on the development set. With the backbone frozen, we optimize only the selected mask weights using the GEC loss and select  $\alpha$  from  $\{0.1, 0.25, 0.5, 1.0\}$  based on development-set performance.

**Baseline Settings.** We evaluate representative open-source multilingual LLMs, including Qwen3-8B, LLaMA3.1-8B, and Sailor2-8B, under both Base and LoRA-finetuned settings. The Base setting evaluates the prompt-only correction ability of the original model, while the LoRA setting evaluates the effect of parameter-efficient supervised fine-tuning. We further apply our method to both settings, denoted as Base\* and LoRA\*, to examine whether sparse neuron-level control can serve as a lightweight post-hoc enhancement for low-resource GEC. For analysis experiments, we use the LoRA fine-tuned Sailor2-8B model as the base model. This setting allows us to conduct controlled comparisons of target modules, intervention directions, activation matrices, and efficiency under the same language and model configuration. For extension experiments, we evaluate our method on LoRA fine-tuned Qwen3-0.6B to examine its gen-

eralization to smaller-scale LLMs and on LoRA fine-tuned Qwen3-8B with Chinese to verify its effectiveness in a high-resource language setting.

## 4.2 Experimental Results

Table 2 reports the main results across four languages and three LLM backbones. Overall, our approach consistently improves both the original models and their LoRA fine-tuned variants. In the Base setting, applying our method increases the average  $F_{0.5}$  from 6.95 to 12.44 on Llama3.1-8B, from 5.29 to 9.76 on Qwen3-8B, and from 6.24 to 15.46 on Sailor2-8B. This shows that it can activate useful linguistic neurons in frozen LLMs and improve GEC without updating model parameters. In the LoRA setting, our method further improves the average  $F_{0.5}$  from 14.21 to 19.18, 15.55 to 19.63, and 25.64 to 27.71 on the three backbones, respectively, indicating that it can serve as an effective lightweight enhancement after parameter-efficient fine-tuning. Across languages, the gains are especially clear on Vietnamese, Khmer, and Burmese, where our approach improves correction precision and the final  $F_{0.5}$  score under most settings. Since  $F_{0.5}$  emphasizes precision, these results suggest that it does not merely encourage more edits but provides more controlled correction behavior by enhancing corrected-form linguistic neurons and suppressing error-associated ones.

## 4.3 Ablation Study

We first study the effect on different target modules, including attention submodules, MLP submodules, and all selected submodules. As shown in Fig-

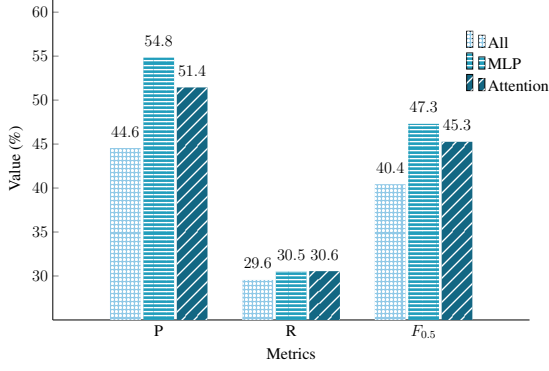


Figure 3: Comparison of P, R, and  $F_{0.5}$  scores across different target modules.

ure 3, the MLP setting achieves the best performance, with an  $F_{0.5}$  score of 47.30. It outperforms the attention-only setting by 2.04 points and the all-module setting by 6.86 points. This result indicates that linguistic neurons useful for GEC are more concentrated in MLP representations. Although attention modules also provide useful correction signals, all submodules may introduce redundant or noisy interventions, weakening the precision of neuron-level control.

#### 4.4 Effect of Activation Matrix

##### 4.4.1 Positive / Negative Neuron Analysis

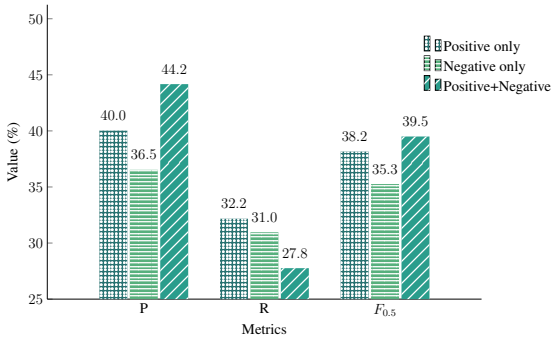


Figure 4: Effect of positive and negative neurons.

We further analyze the role of different intervention directions in the activation matrix. Specifically, we compare three settings: using only positive linguistic neurons, using only negative linguistic neurons, and using both types of neurons. As shown in Figure 4, combining positive and negative linguistic neurons achieves the best  $F_{0.5}$  score of 39.50. Positive neurons alone obtain higher recall, suggesting that corrected-form neurons help the model make more edits. Negative

neurons alone perform worse, indicating that suppressing error-associated neurons is insufficient by itself. The combined setting achieves the best precision and overall  $F_{0.5}$ , showing that activating corrected-form neurons and suppressing error-associated neurons provide complementary benefits for low-resource GEC.

##### 4.4.2 Visualization of the Activation Matrix

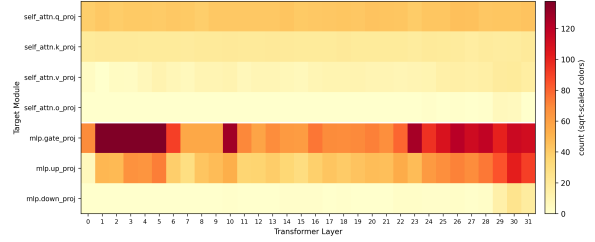


Figure 5: Layer-wise and module-wise visualization of the learned activation matrix on Vietnamese using Sailor2-8B-LoRA.

To better understand the internal behavior of our method, we visualize the learned activation matrix on Vietnamese with Sailor2-8B-LoRA. As shown in Figure 5, the selected linguistic neurons are not uniformly distributed across layers and modules. Instead, they are mainly concentrated in MLP submodules, especially the `mlp.gate_proj` projection. The `mlp.up_proj` module also shows stronger activation density in later Transformer layers, while `mlp.down_proj` and most attention projections contain fewer selected neurons. Among attention modules, `self_attn.q_proj` has relatively higher density than key, value, and output projections, but its overall magnitude is still lower than that of MLP projections. This structured distribution is consistent with the target-module ablation results, where applying the learned mask to MLP submodules achieves stronger correction performance than using attention-only or all-module intervention. It suggests that low-resource GEC behavior is more strongly associated with MLP representations, which are likely to encode lexical, syntactic, and correction-oriented features.

##### 4.4.3 Effect on Different Edit Types

Table 3 shows the error-type changes before and after mask injection on Vietnamese with Sailor2-8B-LoRA. Overall, applying the learned mask increases true positives by 17 and reduces false positives and false negatives by 15 and 17, respectively, indicating that the optimized activation matrix improves both edit detection and edit precision. The

| Type    | Changes in error counts |             |             | Selected neurons |          |
|---------|-------------------------|-------------|-------------|------------------|----------|
|         | $\Delta$ TP             | $\Delta$ FP | $\Delta$ FN | Positive         | Negative |
| M       | +1                      | +1          | -1          | -                | -        |
| R       | +5                      | -6          | -5          | -                | -        |
| S       | +11                     | -9          | -11         | -                | -        |
| W       | 0                       | -1          | 0           | -                | -        |
| Overall | +17                     | -15         | -17         | 35,294           | 31,061   |

Table 3: Changes in error-type distribution. M, R, S, and W denote missing, redundant, substitution, and word-order edits, respectively.  $\Delta$ TP: correct edit;  $\Delta$ FP: spurious edit;  $\Delta$ FN: missing edit.

largest gain comes from substitution errors, where TP increases by 11 while FP and FN decrease by 9 and 11, suggesting that it is especially effective for lexical and grammatical replacement decisions. These results corroborate our approach: positive neurons strengthen corrected-form representations, while negative neurons suppress error-related or over-editing signals.

#### 4.5 Efficiency Analysis

| Comparison axis          | LoRA                      | Activation Mask                           |
|--------------------------|---------------------------|-------------------------------------------|
| Trainable parameters     | 46.14M                    | 66.36K                                    |
| Trainable object         | Low-rank adapter matrices | One scalar per selected linguistic neuron |
| Updated model components | Adapter weights           | only external mask values                 |
| Training data used       | 50000 SFT examples        | 10000 train and 1000 dev                  |
| Correction quality       | $F_{0.5}=0.4616$          | $F_{0.5}=0.4730$                          |
| Artifact size            | 176.06 MiB adapter        | 1.71 MiB mask                             |

Table 4: Efficiency comparison between LoRA fine-tuning and the learned mask on Vietnamese Sailor2-8B-LoRA.

Table 4 compares the activation mask with LoRA fine-tuning on Vietnamese using Sailor2-8B. Activation mask optimizes only one scalar for each selected linguistic neuron, reducing the number of trainable parameters from 46.14M to 66.36K. The saved artifact is also much smaller, decreasing from a 176.06 MiB LoRA adapter to a 1.71 MiB activation mask. Despite this large reduction in trainable parameters and storage cost, the activation mask achieves a slightly higher  $F_{0.5}$  score than LoRA in this setting. These results show that our method provides a lightweight post-hoc control mechanism, where correction behavior can be improved by optimizing sparse external mask values while keeping the backbone model unchanged.

#### 4.6 Case Study

As shown in Table 5, the Base model leaves all three misspelled words unchanged, indicating that prompt-only can not sufficiently activate its internal knowledge of Vietnamese grammatical patterns. LoRA fine-tuning successfully corrects the original errors, but introduces a new spelling error by changing the second occurrence of “người” to “ngúoi”. This suggests that fine-tuning improves correction ability but may still produce unstable or unnecessary edits. In contrast, our method corrects all three errors without modifying the remaining correct content, which demonstrates that neuron-level intervention can activate the internal grammatical knowledge of the model, thereby alleviating the incomplete and unstable correction behavior.

#### 4.7 Effect on different scale of LLM

To examine whether the benefit of our method changes with model capability and language-specific knowledge, we conduct additional experiments on Qwen3-0.6B across four languages. As shown in Table 6, our method improves  $F_{0.5}$  in all Base and LoRA settings. The average  $F_{0.5}$  increases from 2.53 to 4.59 for the Base model and from 11.93 to 13.64 after LoRA fine-tuning. The absolute average gain is slightly smaller after LoRA adaptation (+1.71 versus +2.06), suggesting that additional language-specific knowledge may partially reduce the intervention benefit. However, this pattern is not uniform across languages: the LoRA gains are larger in Lao and Burmese. Therefore, the benefit does not monotonically disappear as model knowledge increases; instead, its magnitude depends on the language and the backbone’s existing correction ability.

### 5 Conclusion

In this paper, we propose an automatic linguistic neuron probing and optimization framework to address the challenge of low-resource GEC. Concretely, we first identify correction-sensitive linguistic neurons by observing neuron changes across low-resource GEC corpora, producing a linguistic probe matrix that determines whether a neuron is activated. Then, we construct an external weight matrix to control the activation intensity of each selected neuron, optimized directly via the GEC loss while keeping the backbone LLM fully frozen. Finally, we combine the

| Output           |                                                                                                                                                                                                  |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Source           | Họ dẫn <b>người</b> xuyên Việt <b>đen</b> gặp người dẫn <b>trường</b> trình.<br><i>They took the [invalid form] traveling across Vietnam [invalid form] meet the [incorrect form] presenter.</i> |
| Target           | Họ dẫn <b>người</b> xuyên Việt <b>đen</b> gặp người dẫn <b>chương</b> trình.<br><i>They took the person traveling across Vietnam to meet the host.</i>                                           |
| Base             | Họ dẫn <b>người</b> xuyên Việt <b>đen</b> gặp người dẫn <b>trường</b> trình.<br><i>The Base model leaves all three errors unchanged.</i>                                                         |
| LoRA fine-tuning | Họ dẫn <b>người</b> xuyên Việt <b>đen</b> gặp <b>người</b> dẫn <b>chương</b> trình.<br><i>LoRA corrects the original errors but introduces a new spelling error in người.</i>                    |
| Ours             | Họ dẫn <b>người</b> xuyên Việt <b>đen</b> gặp người dẫn <b>chương</b> trình.<br><i>Our method corrects all three errors without introducing unnecessary edits.</i>                               |

Table 5: A Vietnamese correction case produced by different models.

| Model | vi           | my           | lo          | kh          | Avg.         |
|-------|--------------|--------------|-------------|-------------|--------------|
| Base  | 8.10         | 0.26         | 0.09        | 1.66        | 2.53         |
| Base* | <b>15.42</b> | <b>0.31</b>  | <b>0.11</b> | <b>2.51</b> | <b>4.59</b>  |
| LoRA  | 25.10        | 9.09         | 5.39        | 8.13        | 11.93        |
| LoRA* | <b>26.75</b> | <b>10.78</b> | <b>7.88</b> | <b>9.16</b> | <b>13.64</b> |

Table 6:  $F_{0.5}$  results on Qwen3-0.6B. The superscript \* denotes the application of our method.

probe matrix and the weight matrix to obtain an activation matrix, which is injected during inference to enhance correction-favored neurons and suppress error-associated neurons. Experiments on four benchmarks show that our method consistently improves both original LLMs and LoRA fine-tuned models. Further analysis reveals that the learned masks produce interpretable layer-wise and module-wise neuron distributions and offer an efficient control mechanism with far fewer trainable parameters than conventional fine-tuning. These findings demonstrate that sparse linguistic neuron control can enhance stability and controllability in low-resource GEC, pointing to a promising direction for future research.

## Limitations

This work has several limitations. First, although the activation mask improves both original LLMs and LoRA-finetuned models, our experiments are still limited to a small set of languages and model families. More experiments on typologically diverse languages and stronger multilingual LLMs are needed to further verify its generality. Second, the activation mask relies on erroneous-corrected sentence pairs to construct contrastive representations and optimize external mask weights. Therefore, its effectiveness may be affected by the qual-

ity and coverage of available GEC data, especially for languages with noisy or limited annotations. Finally, our evaluation relies primarily on standard automatic metrics (e.g.,  $F_{0.5}$ ), and future work should incorporate human evaluation to better assess the fluency and semantic fidelity of the corrections.

## Acknowledgments

We thank all reviewers for their helpful comments. This work is financially supported by the National Natural Science Foundation of China (62666042, 62306129, U21B2027, 62366027, 62266028, 62466030), Yunnan Fundamental Research Projects (202401CF070121, 202401BC070021, 202301AS070047, 202601AT070095), Yunnan Provincial Major Science and Technology Special Plan Projects (202602AD080015, 202502AD080016, 202502AD080014, 202402AG050007, 202103AA080015, 202202AD080003, 202203AA080004), Kunming University of Science and Technology “Double First-rate” Construction Joint Projects (202301BE070001-027, 202201BE070001-021), Yunnan High and New Technology Industry Project (201606), Science and Technology Projects of Yunnan Universities Serving Key Industries (FWCY-ZD2025006), and the First Batch of Projects for the 2025 Intergovernmental International Science, Technology and Innovation Cooperation (2025YFE0121700).

## References

Steven Bills, Nick Cammarata, Dan Mossing, Hunter Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. [Lan-](#)

- guage models can explain neurons in language models. *OpenAI*.
- Yayu Cao, Tianxiang Wang, Lvxiaowei Xu, Zhenyao Wang, and Ming Cai. 2025. [CxGGEC: Construction-guided grammatical error correction](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6143–6156. Association for Computational Linguistics.
- Viktoriia A. Chekalina, Anna Rudenko, Gleb Mezentsev, Aleksandr Mikhalev, Alexander Panchenko, and Ivan Oseledets. 2024. [SparseGrad: A selective method for efficient fine-tuning of MLP layers](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14929–14939, Miami, Florida, USA. Association for Computational Linguistics.
- Shamil Chollampatt, Raymond Hendy Susanto, Liling Tan, and Ewa Szymanska. 2020. [Can automatic post-editing improve NMT?](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2736–2746, Online. Association for Computational Linguistics.
- Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2023. [Language-family adapters for low-resource multilingual neural machine translation](#). In *Proceedings of the Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023)*, pages 59–72, Dubrovnik, Croatia. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. [Knowledge neurons in pretrained transformers](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 8493–8502. Association for Computational Linguistics.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [QLoRA: Efficient finetuning of quantized LLMs](#). In *Advances in Neural Information Processing Systems*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186. Association for Computational Linguistics.
- Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. [Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45. Association for Computational Linguistics.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. [Transformer feed-forward layers are key-value memories](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495. Association for Computational Linguistics.
- Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. Minimally-augmented grammatical error correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 357–363, Hong Kong, China. Association for Computational Linguistics.
- Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. [Finding neurons in a haystack: Case studies with sparse probing](#). *Transactions on Machine Learning Research*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. [Parameter-efficient transfer learning for NLP](#). In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching neural grammatical error correction as a low-resource machine translation task](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 595–606. Association for Computational Linguistics.
- Anisia Katinskaia and Roman Yangarber. 2024. [GPT-3.5 for grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843, Torino, Italia. ELRA and ICCL.
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An empirical study of incorporating pseudo data into grammatical error correction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 1236–1242. Association for Computational Linguistics.

- Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. [Large language models are state-of-the-art evaluator for grammatical error correction](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.
- Aomi Koyama, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021. [Comparison of grammatical error correction using back-translation models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–135, Online. Association for Computational Linguistics.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. Association for Computational Linguistics.
- Wei Li and Houfeng Wang. 2024. [Detection-correction structure via general language model for grammatical error correction](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1748–1763. Association for Computational Linguistics.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4582–4597. Association for Computational Linguistics.
- Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. [Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 205–219, Toronto, Canada. Association for Computational Linguistics.
- Agnes Luhtaru, Elizaveta Korotkova, and Mark Fishel. 2024. [No error left behind: Multilingual grammatical error correction with pre-trained translation models](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1209–1222, St. Julian’s, Malta. Association for Computational Linguistics.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in GPT](#). In *Advances in Neural Information Processing Systems*.
- Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023. [Mass-editing memory in a transformer](#). In *International Conference on Learning Representations*.
- Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. [The CoNLL-2013 shared task on grammatical error correction](#). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Hefernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. [No language left behind: Scaling human-centered machine translation](#). *arXiv preprint arXiv:2207.04672*.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzshanskyi. 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. Association for Computational Linguistics.
- Peijun Qing, Chongyang Gao, Yefan Zhou, Xingjian Diao, Yaoqing Yang, and Soroush Vosoughi. 2024. [AlphaLoRA: Assigning LoRA experts based on layer training quality](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20511–20523, Miami, Florida, USA. 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.
- Olga Seminck, Yoann Dupont, Mathieu Dehouck, Qi Wang, Noé Durandard, and Margo Novikov. 2025. [Lattice @MultiGEC-2025: A spiteful multilingual language error correction system using LLaMA](#). In *Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning*, pages 34–41, Tallinn, Estonia. University of Tartu Library.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. [Activation addition: Steering language models without optimization](#). *arXiv preprint arXiv:2308.10248*.
- 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. Association for Computational Linguistics.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua,

and Colin Raffel. 2021. [mT5: A massively multi-lingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–498. Association for Computational Linguistics.

Kai Yao, Penglei Gao, Lichun Li, Yuan Zhao, Xiaofeng Wang, Wei Wang, and Jianke Zhu. 2024. [Layer-wise importance matters: Less memory for better performance in parameter-efficient fine-tuning of large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1977–1992, Miami, Florida, USA. Association for Computational Linguistics.

Zheng Yuan and Ted Briscoe. 2016. [Grammatical error correction using neural machine translation](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386. Association for Computational Linguistics.

Min Zeng, Jiexin Kuang, Mengyang Qiu, Jayoung Song, and Jungyeul Park. 2024. [Evaluating prompting strategies for grammatical error correction based on language proficiency](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6426–6430, Torino, Italia. ELRA and ICCL.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. [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.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

## A Appendix

### A.1 Effect on high-resource language

We further examine whether our method remains effective beyond low-resource settings by evaluating it on the Chinese FCGEC dataset with Qwen3-8B. As shown in Table 7, compared with previously published Chinese GEC systems, our method

achieves an  $F_{0.5}$  score of 56.47, outperforming DoGoGLM and CxGEC by 5.51 and 3.63 points, respectively. It also obtains higher precision and recall than both systems. These results demonstrate that our method is not limited to low-resource languages but can also activate the internal grammatical knowledge within the high-resource language.

| Model                              | P            | R            | $F_{0.5}$    |
|------------------------------------|--------------|--------------|--------------|
| <i>DoGoGLM</i> (Li and Wang, 2024) | 55.75        | 37.91        | 50.96        |
| <i>CxGEC</i> (Cao et al., 2025)    | 55.90        | 35.92        | 52.84        |
| <i>Ours</i>                        | <b>61.54</b> | <b>42.47</b> | <b>56.47</b> |

Table 7: Comparison with previously published systems on the public FCGEC test set.

We also investigate whether our method benefits when trained with more data. We use 50K, 150K, and 250K training examples and all settings are evaluated on the same MuCGEC set.

| Scale | Method | P     | R     | $F_{0.5}$    | $\Delta F_{0.5}$ |
|-------|--------|-------|-------|--------------|------------------|
| 50K   | LoRA   | 23.64 | 12.04 | 19.85        | –                |
|       | LoRA*  | 25.47 | 14.28 | <b>22.02</b> | <b>+2.17</b>     |
| 150K  | LoRA   | 28.35 | 16.94 | 24.99        | –                |
|       | LoRA*  | 28.61 | 16.87 | <b>25.12</b> | <b>+0.13</b>     |
| 250K  | LoRA   | 29.08 | 15.21 | 24.59        | –                |
|       | LoRA*  | 29.42 | 16.94 | <b>25.64</b> | <b>+1.05</b>     |

Table 8: The superscript \* denotes the application of our method.

As shown in Table 8, our method consistently improves  $F_{0.5}$  across all training scales. The improvement is largest with 50K examples (+2.17), while smaller gains are observed with 150K (+0.13) and 250K (+1.05) examples. Although the benefit becomes less pronounced as more training data are introduced, it does not disappear and is not strictly monotonic with data size. These results demonstrate that our method remains effective in high-resource languages and generalizes across different training-data scales.