Investigating Neurons and Heads in Transformer-based LLMs for Typographical Errors

Anonymous ACL submission

Abstract

This paper investigates how LLMs encode inputs with typos. We hypothesize that specific neurons and attention heads recognize typos and fix them internally using local and global contexts. We introduce a method to identify **typo neurons** and **typo heads** that work actively when inputs contain typos. Our experimental results suggest the following: 1) LLMs can fix typos with local contexts when the typo neurons in either the early or late layers are activated, even if those in the other are not. 2) Typo neurons in the middle layers are the core of typo-fixing with global contexts. 3) Typo heads fix typos by widely considering the context not focusing on specific tokens. 4) Typo neurons and typo heads work not only for typo-fixing but also for understanding general contexts.[1]

1 Introduction

Inputs for large language models (LLMs) sometimes contain typographical errors (typos) (Zheng and Saparov, 2023; Wang et al., 2024a; Zhu et al., 2023). LLMs often make correct answers on inputs with typos (Wang et al., 2024a), which implies that LLMs can "fix" typos to recover the intended meaning. However, LLMs sometimes imperfectly fix the meaning against typos, which might "damage" the performance of LLMs on downstream tasks (Zhuo et al., 2023; Wang et al., 2023; Zhu et al., 2023; Edman et al., 2024). To reduce the impact of typos on LLMs, it is essential to understand both their robustness against typos and the reasons for performance degradation caused by typos more deeply.

Existing studies have primarily focused on the surface-level exhibition of performance degradation due to typos (Wang et al., 2023; Zhu et al., 2023) and methods for improving robustness against typos (Zheng and Saparov, 2023; Zhuo et al., 2023; Almagro et al., 2023). Few studies have investigated how typos affect LLM's inner workings (Kaplan et al., 2024; García-Carrasco et al., 2024b). However, previous work focused on cases where the input contains only a few subwords and a typo. Therefore, they examined typo-fixing working with only local contexts. In contrast, the performance of typo correction can be improved by observing longer (global) contexts (Li et al., 2020; Ji et al., 2021). This implies that LLMs might see global contexts when handling typo inputs.

Based on these previous works, we hypothesize that LLMs with the Transformer-based decoder also fix typos along two axes: typo-fixing with local contexts, which focuses on nearby subwords, and typo-fixing with global contexts, which understands longer contextual information. To verify this hypothesis, we investigated neurons (**typo neurons**) and attention heads (**typo heads**) in LLMs. First, we investigated the inner workings against typos in contextualized words using a word identification task (§3). Then, we propose a method to identify typo neurons (§4) and typo heads (§5). Subsequently, we analyze the differences in their behavior between cases where the model is damaged by typos and cases or not.

We conducted experiments using various LLMs to investigate the inner workings when inputs contain typos. Our findings suggest the following:

- LLMs can fix typos when the typo neurons in either the early or late layers, both of which focus on local contexts, are activated, even if those in the other are not.
- Typo neurons in the middle layers are responsible for typo-fixing considering global contexts, regardless of the models.
- Typo heads fix typos using the local and global contexts, not focusing on specific tokens.
- Typo neurons and typo heads not only fix typos but also understand general grammatical or morphological features.

[1] We upload our code including creating the dataset to the supplementary material. We will release this after accepted.

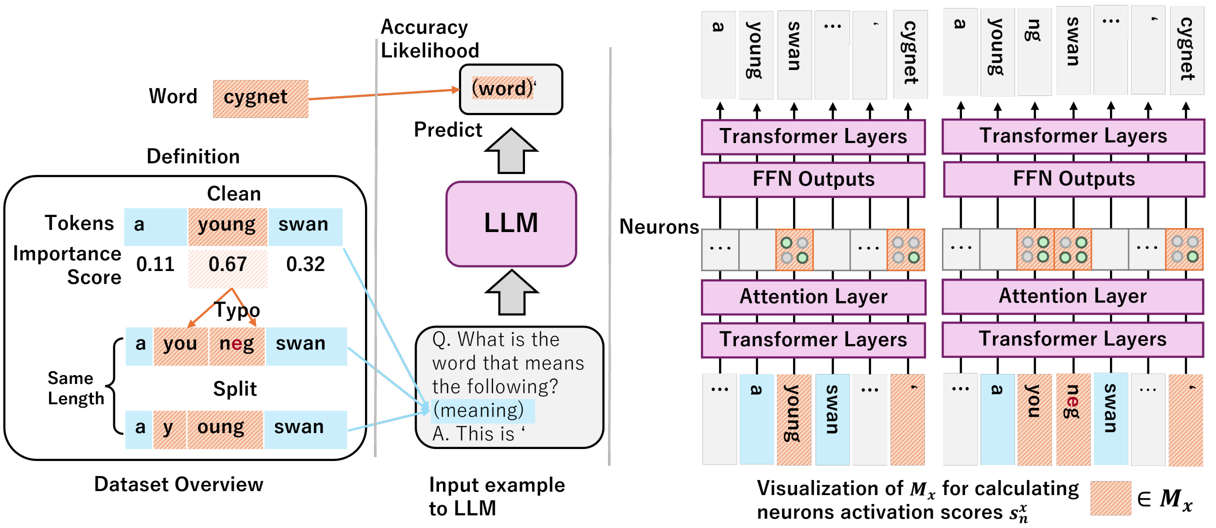

Figure 1: The dataset overview (left), an overview of an input example to LLM (middle), and the visualization of M_x for calculating neurons activation score s_n^x (right).

2 Related work

2.1 Analysis of LLMs against Typos

Typos are mistakes in writing or typing letters, categorized into insertion, deletion, substitution, and reordering (Gao et al., 2018). Research on the robustness of LLMs regards typos as a perturbation. Typos change the token sequence obtained through the tokenization process. Changing the token sequence potentially leads to a different output, even if the sentence is the same (Tsuji et al., 2024). Most existing LLM studies about typos focus on measuring the robustness against perturbed inputs (Wang et al., 2021, 2023; Zhu et al., 2023; Edman et al., 2024) or modifying the architecture or prompts to improve robustness (Zhuo et al., 2023; Zheng and Saparov, 2023; Almagro et al., 2023). Chai et al. (2024) reported that the larger models are more robust to typos. Before the LLM era, researchers corrected typos using specific models for typo-correction (Li et al., 2020; Ji et al., 2021).

2.2 LLM's Interpretability

The feed-forward network (FFN) layer in the Transformer (Vaswani, 2017) has two linear layers separated by an activation function. Recent studies regard the output of the activation function as "neurons" that store knowledge (Geva et al., 2021). It has been reported that some neurons promote specific tasks (Wang et al., 2022, 2024c), knowledge (Dai et al., 2022; Bau et al., 2019; Gurnee et al., 2024), and behaviors (Hiraoka and Inui, 2024; Wang et al., 2024b; Chen et al., 2024).

Some attention heads also respond to specific knowledge (Gould et al., 2024; Voita et al., 2019; García-Carrasco et al., 2024b) or behaviors (McDougall et al., 2024; Crosbie and Shutova, 2024). Additionally, some heads are responsible for merging multiple subwords of a word (Correia et al., 2019; Ferrando and Voita, 2024).

There are various methods to investigate LLM's interpretability. Some measure contributions to the residual stream (García-Carrasco et al., 2024a; Hanna et al., 2024), while others observe intermediate predictions (nostalgebraist, 2020; Kaplan et al., 2024), graph the inference process (Ferrando and Voita, 2024), or directly observe activations (Wang et al., 2022; Hiraoka and Inui, 2024; Wang et al., 2024c). We hypothesize that typo neurons are a type of skill neurons. Therefore we use the direct activation observation method, following previous studies on skill neurons (Wang et al., 2022; Hiraoka and Inui, 2024). Mosbach et al. (2024) concludes that understanding the inner workings is important to improve the model performance.

Lad et al. (2024) divides LLMs into four stages. The early layers convert token-level representations into entity-level representations with local contexts as *Detokenization*. The early middle layers build representations with global contexts as *Feature Engineering*. The late middle layers, convert current representations into next token representations as *Prediction Ensembling*. Finally, the late layers remove the noise and refine the distribution of the next token as *Residual Sharpening*. Elhage et al. (2022) reports that the late layers perform the opposite function of the early layers' *Detokenization*, converting entity-level representations into token-level representations as *Retokenization*.

Kaplan et al. (2024) reveals which layers are responsible for typo-fixing. However, they only focused on isolated words as inputs by layer-level observation. We focus on neurons and heads and experiment with global contexts.

3 Preliminary

We created a dataset that LLMs can solve without typos (§3.2). Then, we applied typos to the dataset (§3.3) and conducted a preliminary experiment to observe accuracy when inputs include typos (§3.4). Next, we identify typo neurons and reveal their specific roles (§4). Similarly, we conduct analogous experiments for attention heads (§4).

3.1 Models

We used Google's Gemma 2 (Team et al., 2024) with 2B, 9B, and 27B parameters, Meta's Llama 3.2 (AI@Meta, 2024) with 1B and 3B parameters, Meta's Llama 3.1 with 8B parameters, and Qwen's Qwen 2.5 (Yang et al., 2024) with 3B, 7B, 14B, 32B parameters; Gemma 2 27B and Qwen 2.5 32B were loaded in bfloat16, while the others were loaded in float32[2]. We conducted all experiments using greedy generation.

3.2 Clean Datasets without Typos

We used a word identification task in which LLMs infer a single word from a given definition. Since typo-fixing relies on vocabulary knowledge, it is crucial to use a task that directly reflects the LLMs' vocabulary knowledge, such as word identification. Moreover, we avoided tasks requiring complex reasoning, such as NLI, as variations in sample difficulty could hinder a clear observation of typo-related phenomena.

For instance, we feed the definition of the word as input, like "*a young swan*", to an LLM, and then the model is expected to output the corresponding word "*cygnet*". Following Greco et al. (2024), we extracted 62,643 word-definition pairs from WordNet (Fellbaum, 2005)[3]. We created the dataset with these pairs. We designed a prompt so that LLMs can solve this task by predicting tokens following outputs, as shown in the middle part of Figure 1.

For our analysis, we need a dataset composed of samples that LLMs can correctly answer when the samples do not include typos. Therefore, we extracted the top 5,000 or 1,000 word-definition pairs after sorting the samples by descending order of likelihood for the correct words[4]. Note that we created a unique dataset for each model.

3.3 Generating Inputs with Typos

3.3.1 Typo Dataset

To focus on text with typos, we generated inputs with typos from the definition part of the clean dataset created in §3.2. We selected the top t most important tokens depending on their importance scores on the word identification task. Then, we injected a random single letter or digit into each selected token as a typo. The importance scores were calculated with the method used in Wang et al. (2023); Li et al. (2019), with the smallest models among ones that share the same tokenizer (e.g., Gemma 2 2B for Gemma 2 or Llama 3.2 1B for Llama 3 family). Specifically, we obtained the importance scores by performing back-propagation while predicting words from their definitions. This process assigns higher gradients to tokens that are important to predict the correct answer. For example, consider the sentence "*a young swan*" with $t = 2$ and the top two most important words are "*young*" and "*swan*." In this case, we inject random letters such as "*e*" and "*5*" into random positions[5] of each word, which results in "*a youn**e**g s**5**wan*."

3.3.2 Split Dataset

We often obtain a different number of subwords when tokenizing typo inputs compared to clean inputs. For instance, the tokenizer encodes the word "young" into a single token, but it tokenizes the typo version "youneg" into two tokens (e.g., "you / neg"). When comparing the inner workings when LLMs encode the clean inputs and the typo inputs, the difference in the token length might prevent appropriate analysis[6].

To divide typo-related inner workings into the factor corresponding to typos and the one to tokenization difference, we created the "split dataset" in addition to the "typo dataset" mentioned in

[2]We described our computing environment in Appendix A.

[3]WordNet via NLTK (Bird and Loper, 2004) ver.3.9.1.

[4]Due to Llama 3.2 1B's worse performance, we could not extract 5,000 pairs for the Llama 3 family. Therefore, we extract 1,000 pairs for the Llama 3 family.

[5]We exclude the positions before the spaces to avoid the situation where a typo would appear at the end of the previous token rather than within the target token.

[6]Kaplan et al. (2024) reported that there are inner workings to fix the original token from differently tokenized subwords. We need to exclude the effect of this factor to deeply focus on the typo-related inner workings.

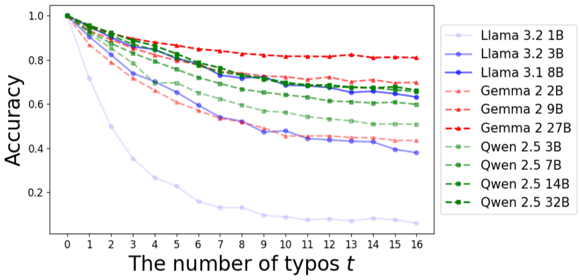

Figure 2: Accuracy on the word identification task with different numbers of typos t.

§3.3.1. The split dataset contains samples tokenized into the same number of tokens as the one with typos. For example, when the typo dataset has a sample whose tokenized sequence is "a / you / neg / swan", an example of counterparts in the split dataset is "a / y / oung / swan" whose length is equivalent to the one of the typo version. We can obtain the various tokenization candidates using the tokenizer and we randomly selected one candidate with the same length as the typo input. This process is shown in Figure 1 (left).

3.4 Preliminary Experiment

To examine the impact of typos on the model performance, we applied typos to t tokens ($1 \leq t \leq 16$) and analyzed the change in accuracy[7].

Figure 2 shows the results. The accuracy of $t = 0$ indicates the performance of the clean data. Since the clean data consists of samples that each model can answer correctly, the accuracy for all models is 1.0. The larger models maintain higher accuracy than the smaller ones even with many typos. This supports the existing work reporting that larger models have robustness against typos (Chai et al., 2024). This result also indicates that the robustness of larger models against typos is insufficient, resulting in a performance drop. We conclude that typos damage performance, but larger LLMs have some robustness against typos, which motivates us to investigate the typo-related inner workings. Furthermore, this leads us to a deep analysis of the differences in robustness against typos among models for further improvement.

4 Typo Neurons

Some FFN layers have been found to combine multiple tokens into a single representation vector (Kaplan et al., 2024; Elhage et al., 2022; Lad et al., 2024). Additionally, it has been reported that certain neurons within LLMs function as "skill neurons" with specific roles (Wang et al., 2022). In this section, we investigate the existence of typo neurons, a particular type of skill neuron that is responsible for recognizing and fixing typos.

4.1 Method to Identify Typo Neurons

Following the approach of Hiraoka and Inui (2024), we compare the activation values of neurons between clean inputs and typo inputs to identify neurons that specifically respond to typos. Let $x \in X$ be a sample of the dataset, where x is a sequence of $|x|$ tokens: $x = w_1, ..., w_m, ..., w_{|x|}$. Each sample comprises the prompt (e.g., "*Q. What is ... A. This is* ") and the answer (e.g., "*cygnet*").

The activation value s_n^X of a neuron n when feeding a dataset X is defined as the following:

$$s_n^X = \frac{1}{|X|}\sum_{x \in X}\left(\frac{1}{|M_x|}\sum_{m \in M_x} f(x_1^m, n)\right), \quad (1)$$

where $|X|$ is the number of samples in the dataset. $f(x_1^m, n)$ is a function calculating the activation value of the neuron n corresponding to w_m when the LLM reads the input $x_1^m = w_1, ..., w_m$. M_x is a set of indices that indicates the token positions, and $|M_x|$ is the number of indices. We define M_x as the indices comprising the answer word tokens and t important words.

For example, in Figure 1, M_x for the clean input is composed of "young" and the apostrophe before "cygnet", while M_x for the typo input is composed of "you", "neg", and the apostrophe and for the split input is "y", "oung", and the apostrophe. In the figure, tokens comprising M_x are indicated with an orange background.

We obtain the responsibility of neurons specialized to the typo inputs separated from clean and split inputs with the following score Δ_n:

$$\Delta_n = s_n^{X_{\text{typo}}} - \max\left(s_n^{X_{\text{clean}}}, s_n^{X_{\text{split}}}\right), \quad (2)$$

where X_{typo}, X_{clean}, and X_{split} are the typo, clean, and the split datasets, respectively.

A larger Δ_n indicates the neuron n that responds specifically to typos but not clean inputs or split inputs. Among the neurons, the top K neurons based on Δ_n scores are identified as typo neurons.

[7] We showed the examples damaged by a typo in Appendix C

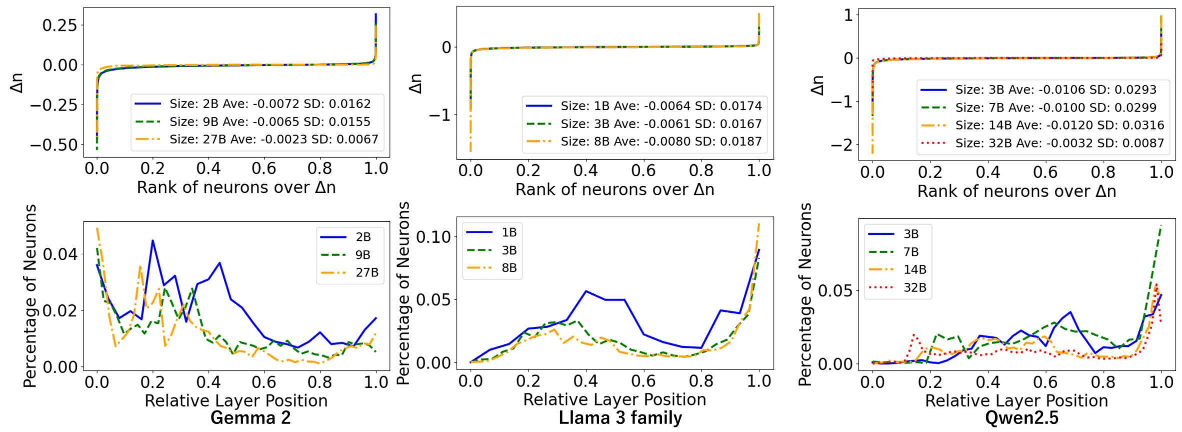

Figure 3: Distribution of Δ_n (upper) and percentage of typo neurons per layer (lower) with $t = 1$. The left figures are for Gemma 2, the center figures are for Llama 3 family and the right figures are for Qwen 2.5.

4.2 Experimental Results

We investigate typo neurons with the method introduced in §4.1. We used the number of typos $t = 1$. Appendix D describes the results for $t = 16$[8].

Figure 3 shows the distribution of Δ_n and the distribution of the typo neurons in each layer. We extracted the top 0.5% of neurons with the highest Δ_n as the typo neurons. The average (Ave) and standard deviation (SD) in Figure 3 indicate that a few neurons have significantly larger scores than others, similar to knowledge and skill neurons (Dai et al., 2022; Wang et al., 2022).

For the distribution of neurons, Llama 3 family and Qwen 2.5 have many typo neurons in the late layers(i.e., from 0.8 to 1.0). In contrast, Gemma 2 models have many typo neurons in the early layers (i.e., from 0.0 to 0.2).

According to Lad et al. (2024), the late layers perform *Residual Sharpening*, which removes noise from representations. Considering typos as noise, it is natural that many typo neurons are in the late layers. Besides, Elhage et al. (2022) reports that the early layers are responsible for *Detokenization* that converts token representations into coherent entities (e.g., words), while the late layers perform *Retokenization* that converts them back into token representations. These suggest that Gemma 2 fixes typos as *Detokenization*, while LLaMA 3 family and Qwen 2.5 fix typos as *Retokenization*. Since both processes use local contexts, we can see the variety of the balance in responsibility between the early and late layers. As shown in Appendix D, with many typos, typo neurons in the late layers of Gemma 2 models also increased. This indicates that the distribution of responsibility between the early and late layers is adaptable.

In the middle layers (i.e., 0.2-0.8), all models have many typo neurons. This suggests that these layers play a common role in typo-fixing across models. Since the early middle layers create representations depending on global contexts with attention heads as *Feature Engineering* and the late middle layers convert current representations to next token representations as *Prediction Ensembling* (Lad et al., 2024), typo-fixing in these layers seem to focus on recognition of global contexts in contrast to the early and late layers.

[8] We investigated the consistency of typo neurons in the Appendix F. We observed consistency in the result. Therefore, we expect the same results for the intermediate number.

4.3 Discussion

While the experimental results in §4.2 suggest the existence of typo neurons, their impact has not been clarified. Then, in this section, we investigate their specific impact, focusing primarily on Gemma 2.

4.3.1 Neuron ablation

We expect typo neurons to work typo-fixing. Therefore, ablating them should result in a remarkable decrease in performance for typo inputs while not affecting the performance for clean inputs.

We test this hypothesis by conducting ablation experiments on typo neurons and randomly selected neurons of Gemma 2 models. Appendix E discusses the results of the ablation study for other models. From a dataset of 5,000 samples, 100 randomly selected samples were used to identify typo neurons. Then, we evaluate the performance of the word identification task using the remaining 4,900 samples by deactivating the identified neurons.

	Clean Dataset	Typo Dataset
Gemma 2 2B	1.00	0.86
⊖ Random Neurons	0.98	0.87
⊖ Typo Neurons	0.84	0.73
Gemma 2 9B	1.00	0.93
⊖ Random Neurons	0.99	0.96
⊖ Typo Neurons	0.93	0.90
Gemma 2 27B	1.00	0.95
⊖ Random Neurons	0.98	0.94
⊖ Typo Neurons	0.96	0.91

Table 1: Accuracy of the word identification task with neuron ablation on clean and typo datasets. "⊖ Random/Typo Neurons" indicates the performance by ablating random and typo neurons, respectively.

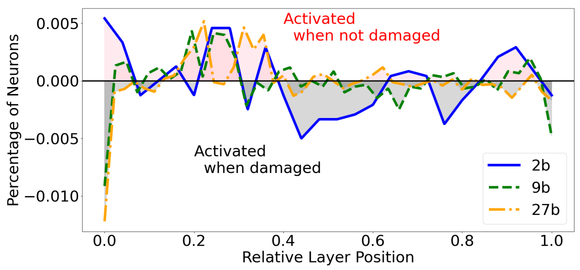

Figure 4: Distribution of typo neurons per layer for samples damaged or not. Values above the black line indicate many typo neurons activated when the LLMs predicted correct words.

Following §4.2, we identified the top 0.5% of neurons as typo neurons. We also randomly selected 0.5% of neurons as a baseline. Deactivation was performed by setting the output values of the neurons to zero. The experiments were conducted for the clean inputs and the typo inputs with $t = 1$.

Table 1 shows the experimental results. For typo inputs, performance remained largely unchanged when random neurons were ablated, regardless of the model. However, performance decreased when typo neurons were ablated. This suggests that a small number of typo neurons play an important role in typo-fixing for typo inputs. For clean datasets, the ablation of typo neurons also resulted in a larger performance decrease than the random neuron ablation. This indicates that typo neurons may not exclusively act on typos but could also play a crucial role in processing general grammatical or morphological features. We can see similar results with the other models (Appendix E).

4.3.2 Neurons for Typo-fixing

The experiments in §4.2 sought typo neurons by comparing clean and typo inputs without considering whether the LLMs could correctly solve the task with typo inputs. This section focuses on the difference in typo neurons between cases where the LLMs answer with typos correctly and incorrectly.

From the dataset of 5,000 samples, we extracted 100 samples where typos did not damage the inferences and the correct word was predicted. Similarly, we extracted another 100 samples where typos damaged the inferences and led to incorrect word prediction. We compared differences in the activation of typo neurons in these two groups. We conducted this experiment with $t = 1$ and compared the difference in the layer distribution of the typo neurons that have the top 0.5% Δ_n.

Figure 4 shows the result. In the 9B and 27B models, the number of typo neurons in the early layers increases when incorrect inferences are predicted. This suggests that some neurons in the early layers might play other roles than typo-related phenomena, and activation of those neurons prevents correct recognition of typos. In the 2B model, when the model fails to fix typos, typo neurons in the middle-middle layers are activated. This suggests that the strong activations observed in the middle-middle layers of Gemma 2 2B in §4.2 are due to neurons damaged by typos rather than contributing to typo-fixing. Across all models, more typo neurons in the early middle layer (e.g., 0.2-0.4) were activated when typos did not damage inferences. This indicates the importance of typo neurons in the early middle layers.

5 Typo Heads

5.1 Method to Identify Typo Heads

Typo-fixing may not solely depend on neurons but subword merging by attention heads (Correia et al., 2019; Ferrando and Voita, 2024) and is based on understanding local and global contexts. We assume two types of such heads for typo inputs: 1) the one focusing on important tokens and 2) the one widely attending contexts.

In this section, we investigate the attention heads specialized to typo inputs. Herein, we calculated the KL divergence between a uniform distribution and the rows of attention maps by considering them as a probability distribution. The KL divergence increases monotonically with the number of tokens, which can result in higher values for typo inputs or split inputs, as they often have more tokens than clean inputs. We alleviate this problem by normalizing the KL divergence with the maximum score

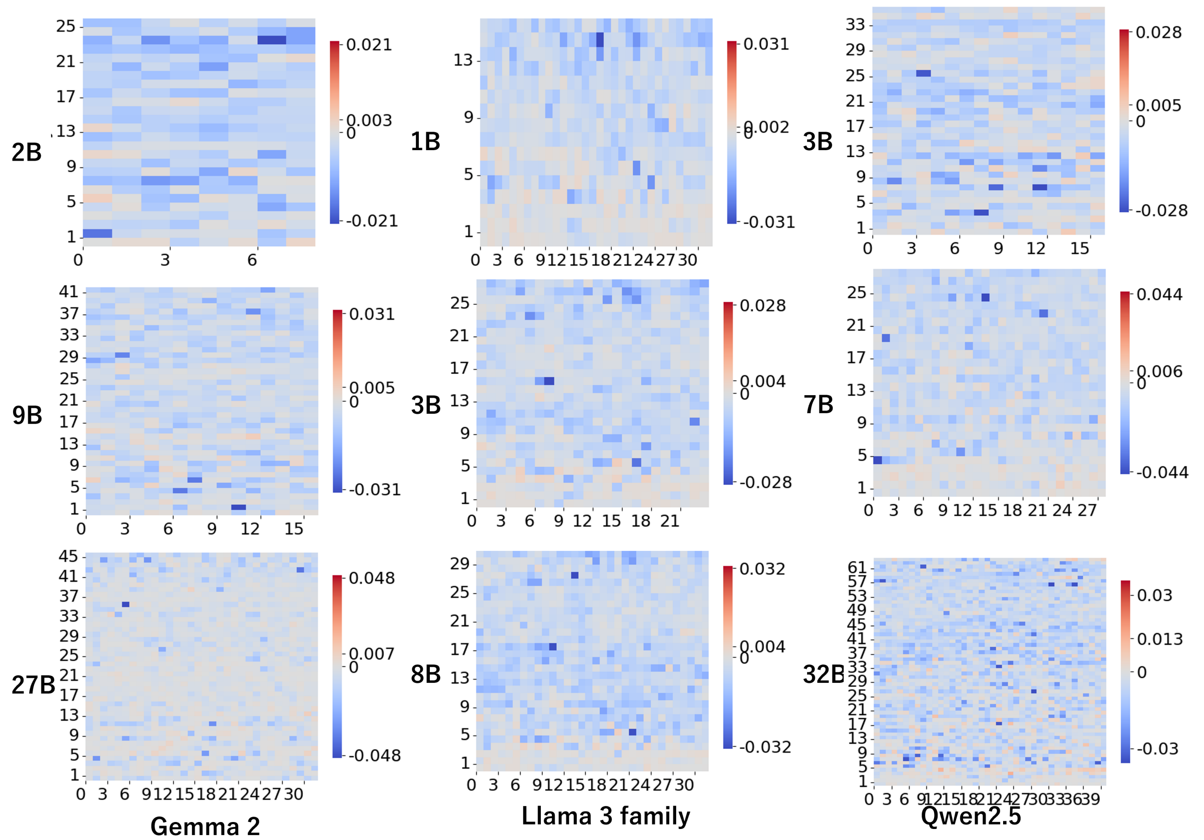

Figure 5: Distribution of Δ_h for each model with $t = 1$. The heat map colors are centered around 0, and the tick mark closest to 0 on the positive side of the heat bar represents the maximum Δ_h. The left figures are for Gemma 2, the center figures are for Llama 3 family and the right figures are for Qwen 2.5.

	Gemma 2			Llama 3.2		Llama 3.1	Qwen 2.5			
	2B	9B	27B	1B	3B	8B	3B	7B	14B	32B
Average	-0.0045	-0.0042	-0.0032	-0.0040	-0.0039	-0.0049	-0.0043	-0.0053	-0.0047	-0.0050
SD	0.0038	0.0041	0.0049	0.0045	0.0040	0.0044	0.0046	0.0056	0.0052	0.0057

Table 2: The average and standard deviation (SD) of Δ_h.

$\log_2 m$, defined as follows:

$$s_h^X = \frac{1}{|X|} \sum_{x \in X} \left(\sum_m \left(\frac{D_{\mathrm{KL}}(P_{x,m,h} || U_m)}{\log_2 m} \right) \right), \quad (3)$$

where $D_{\mathrm{KL}}(\cdot)$ is the function that returns the KL divergence, U_m is a uniform distribution over m elements. $P_{x,m,h}$ is the m-th row of the attention map output by head h for the token sequence x. In decoder models, attention scores for the m-th token and each token from the first to the m-th token sum to 1. Unlike neurons, for the calculation of typo head identification, we did not narrow down the tokens to calculate and used all tokens in prompts.

Similar to Eq. (2) in neurons, the responsibility score of the heads to the typos is defined as follows:

$$\Delta_h = s_h^{X_{\mathrm{typo}}} - \max\left(s_h^{X_{\mathrm{clean}}}, s_h^{X_{\mathrm{split}}}\right), \quad (4)$$

where X_{typo}, X_{clean}, and X_{split} are the typo, clean, and split datasets, respectively. A large absolute value of Δ_h indicates that the head behaves differently between typo and clean inputs. Specifically, a large positive Δ_h indicates the head that focuses on specific tokens for typo-fixing, while a large negative Δ_h indicates the head that widely attends contexts for it. We identified the top J heads with the highest absolute value of Δ_h as typo heads.

5.2 Experimental Results

We used the number of typos $t = 1$. Appendices G and I discuss other settings[9]. As shown in Figure 5, the differences between the maximum and absolute minimum scores are approximately 10 times in

[9]We investigated the consistency of typo heads in Appendix H. We observed consistency in the result. Therefore, we expect the same results for the intermediate number.

	Clean Dataset	Typo Dataset
Gemma 2 2B	1.00	0.86
$\ominus$ Random Heads	0.87	0.80
$\ominus$ Typo Heads	0.81	0.75
Gemma 2 9B	1.00	0.93
$\ominus$ Random Heads	0.80	0.76
$\ominus$ Typo Heads	0.89	0.81
Gemma 2 27B	1.00	0.95
$\ominus$ Random Heads	0.35	0.33
$\ominus$ Typo Heads	0.69	0.67

Table 3: Accuracy of the word identification task with head ablation on clean and typo datasets. "$\ominus$ Random Heads" and "$\ominus$ Typo Heads" indicate the performance by ablating random and typo heads, respectively.

all models. The average and standard deviation in Table 2 also indicate that few heads near the minimum Δ_h are distinctive. These results suggest that heads recognize and fix typos by observing the wider context, not by focusing on specific tokens.

As the model size increases, the proportion of heads with Δh close to zero increases. This contrasts with the results in §4.2, where model differences contributed to the difference in the distribution of typo neurons. However, we can see a similar trend between the distributions of typo neurons and typo heads in very early layers ($\sim 10\%$ layers from the first layer). For instance, Gemma 2 has some heads with large Δh in these layers while the Llama3 family and Qwen 2.5 do not. This trend among models is similar to the one in the distribution of typo neurons (see Figure 3).

5.3 Discussion

In this section, we investigate the specific impact and behavior of typo heads, focusing primarily on Gemma 2 similar to §4.3.

5.3.1 Head Ablation

Following the approach in §4.3.1, we identified typo heads in Gemma 2 using 100 randomly selected samples of the dataset. Then, we ablated these identified typo heads and measured the accuracy on the remaining 4,900 samples. Since the total number of heads is smaller than neurons, we identified the top 1.5% of heads as typo heads (e.g., $J = 3, 10, 22$ for 2B, 9B, 27B, respectively). We also randomly selected 1.5% of heads as a baseline. We performed ablation by setting all attention scores of the selected heads to 0. The experiments were conducted for the clean inputs and the typo inputs with $t = 1$. We described the results of the ablation study for other models in Appendix J.

Table 3 shows the experimental result. In the 9B and 27B models, the ablation of random heads damages the performance in both clean and typo datasets compared to the typo heads, while the ablation of typo heads also degrades the performance to some degree. This suggests that many heads, including those not normally needed for solving that task, cooperate to fix typos, and that no specific few heads are responsible for fixing typos. This is a different result from our hypothesis that only a few heads are responsible for fixing typos. In contrast, for the 2B model, which has fewer heads, the ablation of typo heads resulted in a greater decrease in accuracy than the ablation of random heads. This suggests that when the number of heads and parameters is limited, a few specific heads fix typos, as we hypothesized.

In summary, many heads fix typos in the larger model, while a few specific heads fix the typos in the smaller model. Additionally, since the ablation of typo heads also reduces accuracy on clean datasets, typo heads may play a role in processing general contextual information like typo neurons.

6 Conclusion

This paper investigated how the neurons and heads of Transformer-based LLMs respond to typo inputs. Experimental results show that LLMs can fix typos with local contexts when the typo neurons in either the early or late layers are activated even if those in the other are not. While they fix typos by recognizing local contexts, typo neurons in the middle layer are responsible for the core of typo-fixing with global contexts. Typo heads fix typos using the context widely rather than focusing on specific tokens because many heads have negative Δh. Additionally, many heads fix typos in the larger model, while a few specific heads fix the typos in the smaller model.

Our findings indicate that Transformer-based LLMs fix typos with not only local but also global contexts, which suggests that improving typo robustness requires approaches that emphasize recognition of both local and global contexts. The results of the ablation study show that typo-fixing is related to general grammatical or morphological recognition, suggesting that methods for improving typo robustness may also enhance general contextual recognition performance. These findings also suggest that aiming at improving general contextual recognition could contribute to typo robustness.

Limitation

This work focuses on the investigation of typo-related inner workings. We believe our findings will help develop applications to alleviate the performance decrease caused by typo inputs. However, the discussion of a concrete method for this application is out of the scope of this paper. Our analysis was limited to Gemma 2, Llama 3 family, and Qwen 2.5 models and examined models with sizes up to 32B. Larger models or LLMs with different architectures may have different properties. For hyperparameters, our experiments were performed only at $t \in \{1, 16\}$. Furthermore, our experiments focused on a specific task, and models may show different properties in a wider variety of tasks. We ran all experiments only once, although there was randomness in applying typos and conducting some experiments. For typo neurons, models were observed to have either more typo neurons in the early layers or more in the late layers. This may be due to differences in training methods or datasets. However, the true reason remains unclear. Additionally, our method mostly detected neurons and heads that respond to inputs with typos. However, it cannot distinguish between those that contribute to typo-fixing and those that are damaged by typos. Our head ablation method did not always work well for models other than Gemma 2. Therefore, it remains unclear whether the same trends can be reliably observed in other models. We do not use logit Lens (nostalgebraist, 2020), because we may overlook some types of typo-fixing with it. Specifically, the model can solve the task by ignoring typos. This recovers the intended meaning without explicitly "reconstructing" the original words. While investigating the difference between ignoring typos and restoring original words is important, we consider it next-stage research following the identification of typo neurons and heads.

References

AI@Meta. 2024. Llama 3 model card.

Mario Almagro, Emilio Almazán, Diego Ortego, and David Jiménez. 2023. Lea: Improving sentence similarity robustness to typos using lexical attention bias. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 36–46.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. 2024. Tokenization falling short: On subword robustness in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1582–1599, Miami, Florida, USA. Association for Computational Linguistics.

Jianhui Chen, Xiaozhi Wang, Zijun Yao, Yushi Bai, Lei Hou, and Juanzi Li. 2024. Finding safety neurons in large language models. *arXiv preprint arXiv:2406.14144*.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184, Hong Kong, China. Association for Computational Linguistics.

Joy Crosbie and Ekaterina Shutova. 2024. Induction heads as an essential mechanism for pattern matching in in-context learning. *arXiv preprint arXiv:2407.07011*.

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 (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' understanding of their tokens. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3017–3026, Miami, Florida, USA. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. 2022. Softmax linear units. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/solu/index.html.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Keith Brown, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier.

Javier Ferrando and Elena Voita. 2024. Information flow routes: Automatically interpreting language models at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17432–17445, Miami, Florida, USA. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Jorge García-Carrasco, Alejandro Maté, and Juan Trujillo. 2024a. Detecting and understanding vulnerabilities in language models via mechanistic interpretability. *arXiv preprint arXiv:2407.19842*.

Jorge García-Carrasco, Alejandro Maté, and Juan Carlos Trujillo. 2024b. How does gpt-2 predict acronyms? extracting and understanding a circuit via mechanistic interpretability. In *International Conference on Artificial Intelligence and Statistics*, pages 3322–3330. PMLR.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2024. Successor heads: Recurring, interpretable attention heads in the wild. In *The Twelfth International Conference on Learning Representations*.

Candida Maria Greco, Lucio La Cava, and Andrea Tagarelli. 2024. Talking the talk does not entail walking the walk: On the limits of large language models in lexical entailment recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14991–15011, Miami, Florida, USA. Association for Computational Linguistics.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. 2024. Universal neurons in gpt2 language models. *arXiv preprint arXiv:2401.12181*.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2024. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36.

Tatsuya Hiraoka and Kentaro Inui. 2024. Repetition neurons: How do language models produce repetitions? *arXiv preprint arXiv:2410.13497*.

Tuo Ji, Hang Yan, and Xipeng Qiu. 2021. SpellBERT: A lightweight pretrained model for Chinese spelling check. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3544–3551, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. 2024. From tokens to words: On the inner lexicon of llms. *arXiv preprint arXiv:2410.05864*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Vedang Lad, Wes Gurnee, and Max Tegmark. 2024. The remarkable robustness of LLMs: Stages of inference? In *ICML 2024 Workshop on Mechanistic Interpretability*.

J Li, S Ji, T Du, B Li, and T Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*.

Xiangci Li, Hairong Liu, and Liang Huang. 2020. Context-aware stand-alone neural spelling correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 407–414, Online. Association for Computational Linguistics.

Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2024. Copy suppression: Comprehensively understanding a motif in language model attention heads. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 337–363, Miami, Florida, US. Association for Computational Linguistics.

Marius Mosbach, Vagrant Gautam, Tomás Vergara Browne, Dietrich Klakow, and Mor Geva. 2024. From insights to actions: The impact of interpretability and analysis research on NLP. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3078–3105, Miami, Florida, USA. Association for Computational Linguistics.

nostalgebraist. 2020. interpreting gpt: the logit lens. Accessed on Nov 27, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Kohei Tsuji, Tatsuya Hiraoka, Yuchang Cheng, and Tomoya Iwakura. 2024. Subregweigh: Effective and efficient annotation weighing with subword regularization. *arXiv preprint arXiv:2409.06216*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *Advances in Neural Information Processing Systems*, 36.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. In *Advances in Neural Information Processing Systems*.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, et al. 2024a. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *Data Engineering*, page 48.

Weichuan Wang, Zhaoyi Li, Defu Lian, Chen Ma, Linqi Song, and Ying Wei. 2024b. Mitigating the language mismatch and repetition issues in LLM-based machine translation via model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15681–15700, Miami, Florida, USA. Association for Computational Linguistics.

Weixuan Wang, Barry Haddow, Minghao Wu, Wei Peng, and Alexandra Birch. 2024c. Sharing matters: Analysing neurons across languages and tasks in llms. *arXiv preprint arXiv:2406.09265*.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding skill neurons in pre-trained transformer-based language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11132–11152, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Hongyi Zheng and Abulhair Saparov. 2023. Noisy exemplars make large language models more robust: A domain-agnostic behavioral analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4560–4568, Singapore. Association for Computational Linguistics.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, et al. 2023. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, pages 57–68.

Terry Yue Zhuo, Zhuang Li, Yujin Huang, Fatemeh Shiri, Weiqing Wang, Gholamreza Haffari, and Yuan-Fang Li. 2023. On robustness of prompt-based semantic parsing with large pre-trained language model: An empirical study on codex. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1090–1102, Dubrovnik, Croatia. Association for Computational Linguistics.

A Computing Environment

We used NVIDIA A100 40GB$\times$2 for Gemma 2 and Llama 3.1 8B, NVIDIA A100 80GB$\times$1 for Qwen 2.5, and NVIDIA RTX 3060$\times$1 for Llama 3.1 1B and 3B.

B Models Using the Same Tokenizer

Since LLMs using the same tokenizer share their vocabulary, the impact of typos could be similar. To compare LLMs using the same tokenizer under similar settings, we constructed datasets for such models so that they contain as many identical samples as possible.

C Example of Damaged Outputs

We reported accuracy degradation in samples with typos. Here, we show specific examples where a single typo damaged Gemma 2 9B, leading to incorrect predictions.

Table 4 shows that typos can lead to various types of errors. First, in some cases, the output itself contains a typo, as seen in "Palaemon" becoming "Palaeomon." Additionally, we observed cases such as "gruel" becoming "porridge," where the model repeated a word that was originally in the input definition. We can also observe various other types of cases.

D Typo Neurons for Many Typos

In §4.2, we reported the results for $t = 1$. Here, we describe the behavior of typo neurons with $t = 16$, where many typos are introduced. Since we are comparing $t = 1$, which contains a minimal number of typos, with $t = 16$, which has an unrealistically high number of typos, it is expected that the behavior for real-world typos would fall somewhere between them.

Figure 6 (upper) shows that the maximum value of Δ_n increases across all models. This indicates that typo neurons respond more strongly as the number of typos increases. Since the average and standard deviation remain close to zero, it suggests that even in such environments, most neurons activate similarly to those with clean input.

For the Llama 3 family and Qwen 2.5, the proportion of typo neurons in the late layers increases further, while there are few typo neurons in other layers. However, We extracted only the top 0.5% of neurons with the highest Δ_n as the typo neurons. Therefore, even if neurons in other layers are activated similarly to those in $t = 1$, a significant increase in typo neuron activation in the late layers could cause a ranking inversion of Δ_n. This leads to the possibility that some activated neurons are not extracted as the typo neurons.

To address this, we redefine typo neurons by extracting neurons with Δ_n values greater than the minimum Δ_n of the typo neurons in $t = 1$ for each model. In other words, we extracted neurons that activate equally to or greater than the typo neurons in $t = 1$ as typo neurons. Figure 7 shows the layer-wise distribution of typo neurons under this new criterion. This shows that while typo neurons increase in the late layers of Llama 3 family and Qwen 2.5, they also increase significantly in the middle layers. For Gemma 2, the typo neurons in the early layers decrease, while those in the late layers increase even in Figure 7. This suggests that both the early and late layers are responsible for recognizing local contexts and the balance of responsibility between them can shift.

The number of typo neurons in Qwen 2.5 32B and Gemma 2 27B does not increase compared to the case of $t = 1$ in §4.2, while the number of typo neurons in most other models significantly increases in Figure 7. This suggests that typo neurons in larger models can fix typos regardless of the number of typos.

E Neuron Ablation for Other Models

In §4.3.1, we reported the results for Gemma 2. Here, we examined the ablation study for typo neurons in the Llama 3 family and Qwen 2.5.

Table 5 shows that the results of the ablation study are consistent, while there were differences in typo neuron distributions across models. In all models, ablating random neurons did not reduce accuracy on the typo dataset. In contrast, ablating typo neurons led to a drop in accuracy on both the clean and typo datasets. This indicates that typo neurons may not exclusively act on typos but could also play a crucial role in processing general grammatical or morphological features, regardless of the model.

F Consistency of Typo Neurons

In Appendix D, we showed that typo neurons activate stronger in the case of $t = 16$ than $t = 1$. We also noted that the behavior for real-world typos would fall somewhere between the results of Appendix D and §4.2. However, we have not yet

Definition	Correct Word	Generated Answer
type genus of the family Palae***x***monidae; widely distributed genus	Palaemon	Palaemon
a thin por***z***ridge (usually oatmeal or cornmeal)	gruel	porridge
a native or inhabitant of S***r***ind	Sindhi	Sringi
make f***0***lat or flatter	flatten	flatter
any plant of the genus Gazan***j***ia valued for their showy daisy flowers	Gazania	gaillardia
type ge***q***nus of the Papaveraceae; chiefly bristly hairy herbs with usually showy flowers	Papaver	poppy

Table 4: Example outputs with a typo from Gemma 2 9B. Bold italic characters mean typos

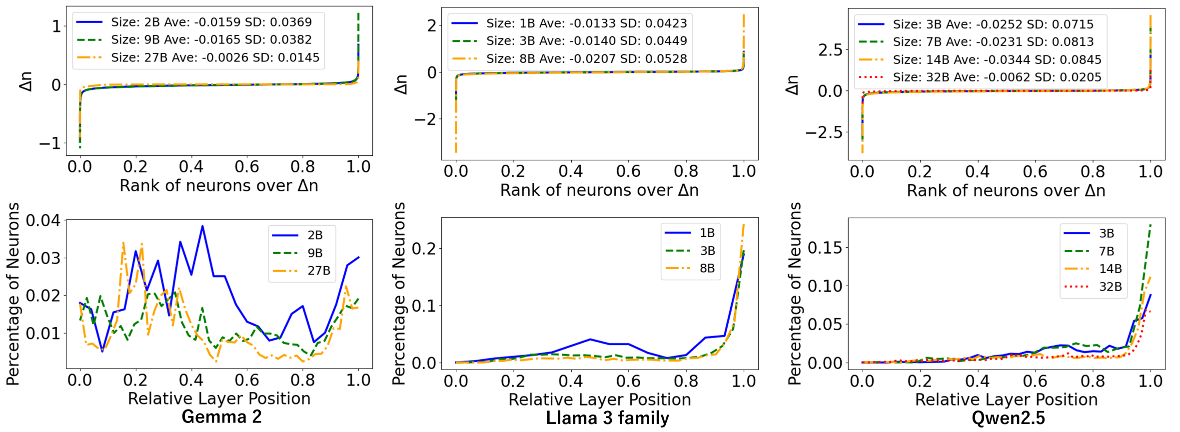

Figure 6: Distribution of Δ_n (upper) and percentage of typo neurons per layer (lower) with $t = 16$. The left figures are for Gemma 2, the center figures are for Llama 3 family and the right figures are for Qwen 2.5.

clarified the degree of consistency in neuron behavior between the $t = 1$ and $t = 16$ cases. Therefore, we computed NDCG (Normalized Discounted Cumulative Gain) by using the ranking of δ_n from the $t = 1$ case and the δ_n scores from the $t = 16$ case to show that consistency. NDCG is calculated as follows:

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\max_{\pi}(\text{DCG@}k)}, \tag{5}$$

Here,

$$\text{DCG@}k = \sum_{\pi(i) \leq k} \frac{2^{l_i} - 1}{\log_2(\pi(i) + 1)}, \tag{6}$$

where $\pi(i)$ is the rank of i, l_i is the score of i, and k is the rank cutoff used to calculate. In this experiment, we set k to 5% of the neurons in each model, consistent with Appendix D and Section 4.2.

As shown in Table 6, all models exhibit very high NDCG scores, indicating that typo neurons remain highly consistent even when the number of typos changes. Therefore, it can be concluded that model behavior for real-world typos would fall somewhere between the results of Appendix D and Section 4.2.

G Typo Heads for Many Typos

Similar to Appendix D, while §5.2 reported for $t = 1$, here we describe the behavior of typo heads under the $t = 16$ setting.

Table 7 shows that Δ_h shifts significantly in the negative direction at $t = 16$ compared to $t = 1$. the minimum values in Figure 8 also shows this transition. Additionally, the increase in dark blue areas in Figure 8 indicates that more heads respond relatively strongly. However, the difference between $t = 1$ and $t = 16$ for typo heads is smaller than for typo neurons.

H Consistency of Typo Heads

Following Appendix F, we also evaluated the consistency of typo heads between the $t = 1$ and $t = 16$ cases by NDCG. Since our analysis of typo heads focused on heads with lower negative scores, we computed NDCG with scores multiplied by -1 and reversed rankings. Following §5.3.1, we set k to 1.5% of the total number of heads.

Table 8 indicates that consistent heads respond to typos regardless of the number of typos. The number of typos affects the intensity of the response in these heads.

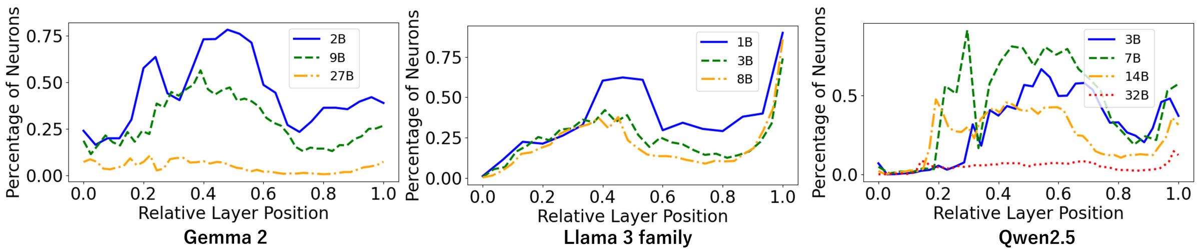

Figure 7: Percentage of typo neurons per layer with $t = 16$ when we extracted neurons that activate greater than the typo neurons at $t = 1$ as typo neurons. The left figures are for Gemma 2, the center figures are for Llama 3 family and the right figures are for Qwen 2.5.

	Clean Dataset	Typo Dataset
Llama 3.2 1B	1.00	0.69
⊖ Random Neurons	0.91	0.61
⊖ Typo Neurons	0.73	0.46
Llama 3.2 3B	1.00	0.90
⊖ Random Neurons	0.97	0.89
⊖ Typo Neurons	0.87	0.79
Llama 3.1 8B	1.00	0.94
⊖ Random Neurons	0.99	0.93
⊖ Typo Neurons	0.83	0.80
Qwen 2.5 3B	1.00	0.92
⊖ Random Neurons	0.99	0.91
⊖ Typo Neurons	0.84	0.71
Qwen 2.5 7B	1.00	0.92
⊖ Random Neurons	0.98	0.92
⊖ Typo Neurons	0.86	0.80
Qwen 2.5 14B	1.00	0.95
⊖ Random Heads	0.99	0.94
⊖ Typo Heads	0.92	0.82
Qwen 2.5 32B	1.00	0.96
⊖ Random Neurons	0.99	0.96
⊖ Typo Neurons	0.93	0.85

Table 5: Accuracy of the word identification task with neuron ablation on clean and typo datasets. "⊖ Random Neurons" and "⊖ Typo Neurons" indicate the performance by ablating random and typo neurons, respectively.

I Typo Heads for Qwen 2.5 14B

Figure 9 shows the distribution of Δ_h for Qwen 2.5 14B, which was not included in §5.2 and Appendix G due to space constraints. The results are consistent with those of other models and model sizes, as the initial layers contain fewer typo heads, and the distribution of typo heads is sparser than in smaller models.

J Head Ablation for Other Models

Similar to Appendix E, we examined the ablation study for typo heads in the Llama 3 family and Qwen 2.5.

In Table 9, both ablations significantly degraded the model's capability in the Llama 3 family, Qwen 2.5 14B and Qwen 2.5 32B, making it difficult to determine the importance of typo heads. In contrast, in Qwen 2.5 3B and Qwen 2.5 7B, the ablation of typo heads decreases accuracy more than the ablation of random heads. Compared to §5.3.1, where ablation of typo heads in the 9B model had little impact on accuracy, this suggests that typo heads remain important even in the middle model in Qwen 2.5, which has few typo neurons and typo heads in the early layers.

K Examples of Typo Head Ablation

When we ablated heads, we observed a significant drop in accuracy in the LLaMA 3 family and Qwen 2.5 14B. This task is constructed using only questions that the models originally solved correctly Therefore, such a drop suggests that the ablation may cause not only a reduced ability to handle typos but also serious damage to the overall performance of the models. To investigate this, we qualitatively examined the outputs of LLaMA 3.2 3B ablating random 1.5% of its heads (10 of the 672 heads).

In Table 10, we observed several types of broken outputs contributing to the decline in performance. These broken outputs often appeared, despite the ablated 1.5% of heads being randomly selected each time. This suggests that the degradation is unlikely due to the accidental removal of particularly important heads.

Additionally, we report several example outputs from Gemma 2 9B, for which ablation seems to work correctly, in Table 11. Unlike Table 10, we do not observe the collapsed examples. There are only errors similar to Table 4.

A possible explanation for the serious damage in Table 10 is that since the model was not trained with dropout, ablating heads during infer-

	Gemma 2			Llama 3.2		Llama 3.1	Qwen 2.5			
	2B	9B	27B	1B	3B	8B	3B	7B	14B	32B
NDCG	0.9441	0.9156	0.9054	0.9477	0.9103	0.9565	0.9386	0.9374	0.9312	0.8771

Table 6: NDCG calculated between the typo neurons in the $t = 1$ and $t = 16$ cases.

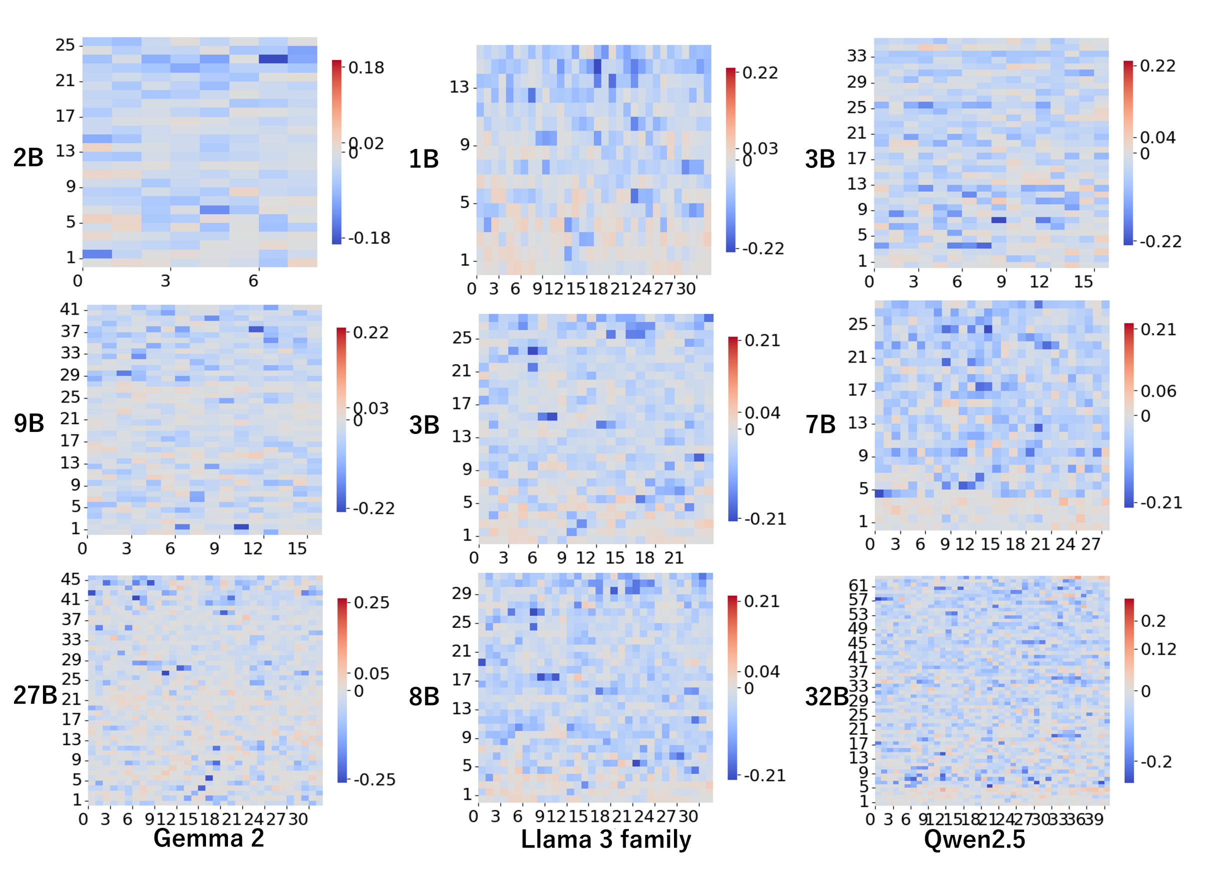

Figure 8: Distribution of Δ_h for each model with $t = 16$. The heat map colors are centered around 0, and the tick mark closest to 0 on the positive side of the heat bar represents the maximum Δ_h. The left figures are for Gemma 2, the center figures are for Llama 3 family and the right figures are for Qwen 2.5.

ence causes unstable behavior. This suggests that these models rely on the simultaneous operation of most heads, rather than assigning distinct or isolated roles to each head. However, models in which ablation studies functioned well were also trained without dropout. Therefore, the reason for this difference remains unclear. While identifying the reason is important, since it is beyond the scope of this paper, we do not investigate it more deeply here.

L Visualization of Typo Heads.

Figure 10 shows the attention maps for each input, using the top 1.5% of heads with the highest absolute value of Δ_h scores in Gemma 2 9B as typo heads.

The typo head in Layer 2 Head 11 recognizes sentence boundaries. There are two possible interpretations of this head. First, it normally detects sentence boundaries, but if there are typos, it simultaneously has the role of detecting typos. Second, it is damaged by typos. Our method has a limitation in that it cannot distinguish between heads that contribute to typo-fixing and those that are damaged by typos. The typo head in Layer 5 Head 7 responds to semantic connections and fixes typos by leveraging synonyms. This is a typical typo-fixing mechanism of early middle layers described above, which is a recognition of global contexts. The typo head in Layer 30 Head 3 fixes typos by recognizing local contexts. Additionally, most typo heads strongly attend to '<bos>'.

When attention norms are considered, such heads may assign little or no weight to the

	Gemma 2			Llama 3.2		Llama 3.1	Qwen 2.5			
	2B	9B	27B	1B	3B	8B	3B	7B	14B	32B
Average	-0.0295	-0.0276	-0.0221	-0.0330	-0.0295	-0.0368	-0.0347	-0.0401	-0.0343	-0.0369
Standard Deviation	0.0317	0.0335	0.0394	0.0442	0.0383	0.0398	0.0557	0.0434	0.0420	0.0452

Table 7: The average and standard deviation of Δ_h with $t = 16$.

	Gemma 2			Llama 3.2		Llama 3.1	Qwen 2.5			
	2B	9B	27B	1B	3B	8B	3B	7B	14B	32B
NDCG	0.9975	0.9923	0.9953	0.9905	0.9849	0.9854	0.9967	0.9932	0.9966	0.9944

Table 8: NDCG calculated between the typo heads in the $t = 1$ and $t = 16$ cases.

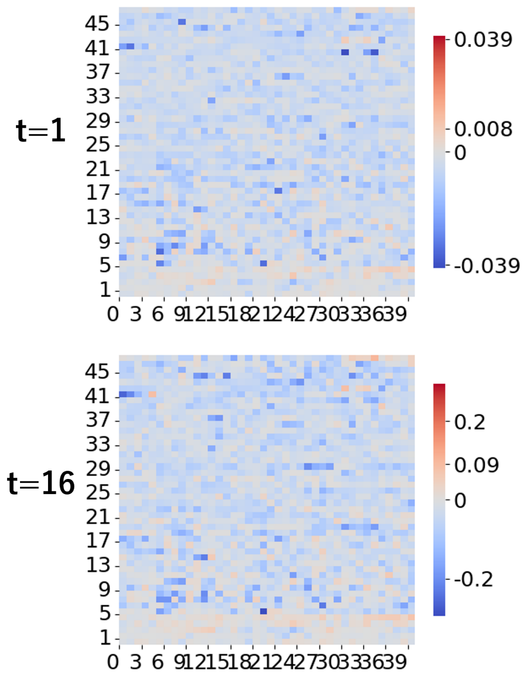

Figure 9: Distribution of Δ_h for Qwen 2.5 14B. The heat map colors are centered around 0, and the tick mark closest to 0 on the positive side of the heat bar represents the maximum Δ_h.

	Clean Dataset	Typo Dataset
Llama 3.2 1B	1.00	0.69
⊖ Random Heads	0.07	0.04
⊖ Typo Heads	0.00	0.00
Llama 3.2 3B	1.00	0.90
⊖ Random Heads	0.10	0.10
⊖ Typo Heads	0.18	0.17
Llama 3.1 8B	1.00	0.94
⊖ Random Heads	0.09	0.08
⊖ Typo Heads	0.10	0.09
Qwen 2.5 3B	1.00	0.92
⊖ Random Heads	0.97	0.88
⊖ Typo Heads	0.46	0.41
Qwen 2.5 7B	1.00	0.92
⊖ Random Heads	0.55	0.53
⊖ Typo Heads	0.39	0.37
Qwen 2.5 14B	1.00	0.95
⊖ Random Heads	0.09	0.09
⊖ Typo Heads	0.13	0.12
Qwen 2.5 32B	1.00	0.96
⊖ Random Heads	0.18	0.16
⊖ Typo Heads	0.15	0.15

Table 9: Accuracy of the word identification task with head ablation on clean and typo datasets. "⊖ Random Heads" and "⊖ Typo Heads" indicate the performance by ablating random and typo heads, respectively.

'<bos>' (Kobayashi et al., 2020). We did not incorporate it into the quantitative scoring because this method does not rescale values to the $[0, 1]$ interval. However, it can provide an instructive viewpoint for visualization. Therefore, we display the norm-corrected attention maps in Figure 11.

For Layer 2 Head 11, it becomes more clear that the head is looking at the sentence boundaries, though attention to '<bos>' remains. For Layer 5 Head 7, the focus on '<bos>' disappears, and the responses to semantic connections become more clear. For Layer 30 Head 3, this head recognizes sentence-level relationships such as attention from A to Q and strongly focuses on '<bos>', even when corrected with the norm. Additionally, the head pays attention to unusual splits in the typo and split inputs.

M Future Work

This paper focuses on the investigation of typo-related inner workings. Therefore, we do not provide any methods to improve LLM's robustness against typos. However, our findings imply how to create more robust LLMs against typos.

Our findings indicate that typo neurons in the early or late layers of Transformer-based LLMs fix typos with local contexts, while typo neurons in the middle layers fix typos with global contexts. The model's robustness against typos may enhanced by

Definition	Correct Word	Generated Answer	Type
relating to or derived from the sun or utilizing the energies of the sun	solar	s	character
a specialist in virology	virologist	v	character
a suite of rooms usually on one floor of an apartment house	apartment	apul	short
compete for something; engage in a contest; measure oneself against others	compete	comp	short
a person who operates a farm	farmer	far(11)2\1\	symbol
a note or passage that is played pizzuicato	pizzicato	ststststststststst	repetition
a structure that allows people or vehicles to cross an obstacle such as a river or canal or railway etc.	bridge	bridgeandandandandandandandand	repetition
soft silky fibers from cotton plants in their raw state	cotton	chellhellhellhellhellhellhellhell	toxic
people of Irelandq or of Irish extraction	Irish	Irhellhellhellhellhellhellhellhell	toxic

Table 10: Example outputs from LLaMA 3.2 3B ablating random 1.5% of heads. Type refers to a coarse-grained classification. Character indicates a single character, short indicates a single token, symbol indicates containing many symbols, repetition indicates repeated words, and toxic indicates harmful outputs (e.g., “hell”). Note that although an output classified as toxic may also exhibit repetition, each output is assigned only a single type for simplicity.

Definition	Correct Word	Generated Answer
install again	reinstall	install
type genus of the family Plantaginaceae; large cosmopolitan genus of mostly small herbs	Plantago	Plantagin
the spreading of a disease (especially cancer) to another part of the body	metastasis	metastasisis
type genus of the family Laminariaceae: perennial brown kelps	Laminaria	kelp
someone belonging to (or as if belonging to) the era of Edward VII	Edwardian	Tudor
type genus of the Solanaceae: nightshade; potato; eggplant; bittersweet	Solanum	Solanaceae

Table 11: Example outputs from Gemma 2 9B ablating random 1.5% of heads.

a mechanism that gives more importance to nearby tokens in the early and late layers and to distant tokens in the middle layers.

Furthermore, the results of the ablation study show that typo-fixing is related to general grammatical or morphological recognition, which suggests that methods for improving general contextual recognition could contribute to typo robustness. For example, a potential research direction could be investigating how additional training on tasks such as grammatical error correction or determining whether a given subword is part of a specific word affects robustness against typos.

Additionally, our study is an important foundation for future research on the internal mechanisms of LLMs. With sufficient computational resources and time, it would be possible to investigate when local or global typo recognition is learned, as well as how differences in training methods affect which layers are responsible for local typo recognition. Although we isolated the effects of typo-fixing by excluding subword merging, we still found that typo neurons and heads have roles in general grammatical or morphological understanding. By using other types of perturbations, such as token replacing, it may be possible to investigate a deeper understanding of these linguistic capabilities.

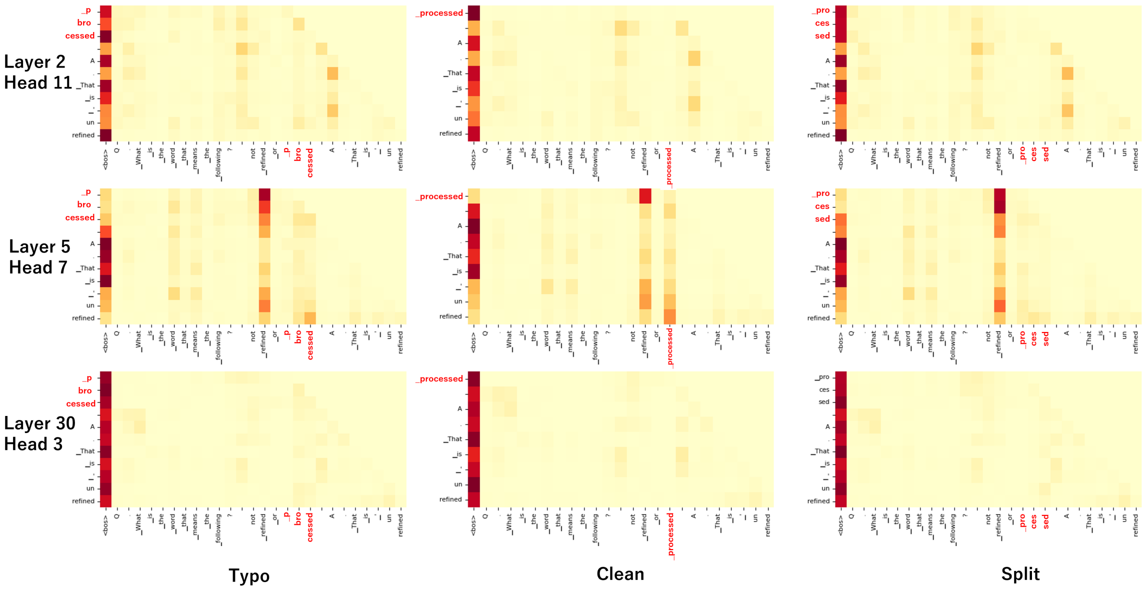

Figure 10: Visualization of typo heads in the 9B model. The word definition in the clean input is “not refined or processed,” and the correct answer is “unrefined”. The word “processed” was changed with a typo to “pbrocessed.”

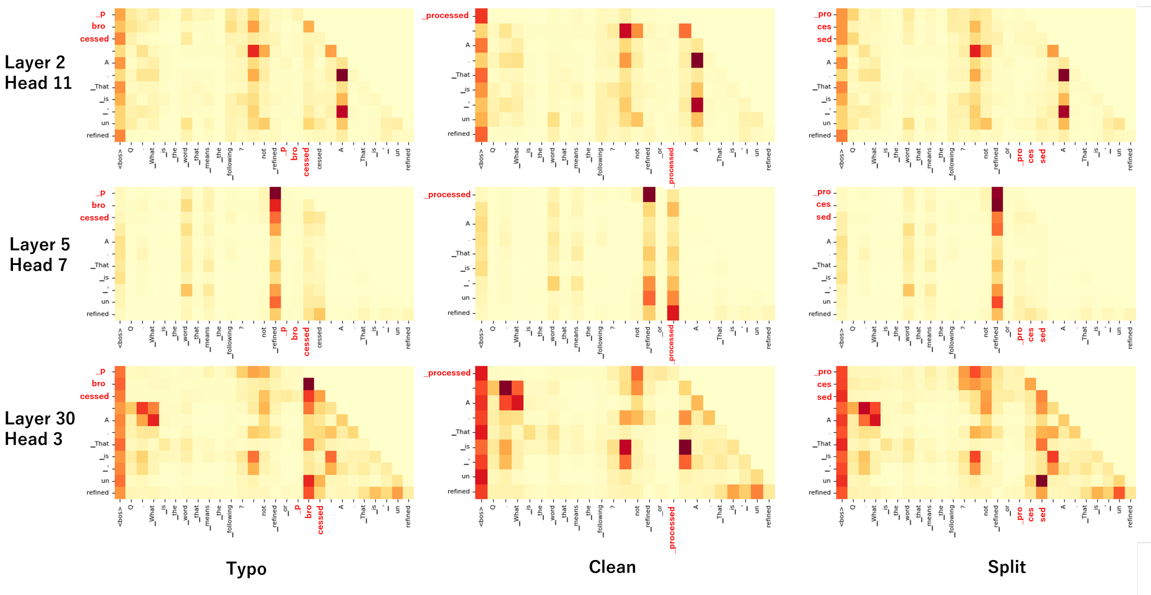

Figure 11: Visualization of typo heads in the 9B model with norm adjustment.