# User Syntax vs LLM Perfection: When do LLMs give up on fixing your spelling?

Yugank Mishra

February 2026

## 1 Abstract

Today's large language models have a robust error-correction mechanism through balancing their pre-trained priors against information provided in context. We investigate this behavior in the domain of orthography, specifically focusing on when repeated misspellings in-context overpower the model's learned correct spelling (the phenomenon of typo induction). Using GPT-2 and Pythia models, we identify a phase transition where context dominates priors. We then localize this behavior to a competition between early-layer MLP frequency biases (favoring statistically likely spelling) and late-layer Attention Heads (favoring in-context induction). We find that, critically, typo induction is distinct from standard factual recall, relying instead on induction circuits that disregard token probability in favor of recent repetition. Ablation studies confirm certain influential heads are causally responsible for typo induction, though their strength is strongly influenced by the tokenizations of the target word and its typo.

## 2 Introduction

Large Language Models have demonstrated robust error-correction abilities orthographically. A rare mispelling or two will not hinder an LLM's understanding of the prompt and the model can, internally, correct the mistake early and generate sound output. From "Tracing Typos in LLMs", it's known that LLMs fix typos internally in their very early layers. Under a mechanistic interpretability lens, the correction mechanism is the selection of the statistically correct spelling coming from the LLM's earlier layers over the incorrect spelling provided from in-context information (fetched through induction heads) in attention (comparison) layers.

However, this correction mechanism can be stressed with *frequent*, *consistent*, typos (ie: 'agetnic' vs 'agentic' as shown in Figure 1). and the model's output will contain the misspelled versions of the word, entering the model's output through the model's induction heads. Thus, we define Typo Induction as the phenomenon when an LLM adopts a misspelled token or sequence of tokens from its context, overriding the learned correct spelling.

But *why* does this occur*?* It's similar to "Competitions of Mechanisms: Tracing how Language Models Handle Facts and Counterfactuals" (Ortu et al), which looks at how LLMs reconcile false factual statements being given to them in their context and quizzed on it, except the false statement in this case is an incorrect spelling.

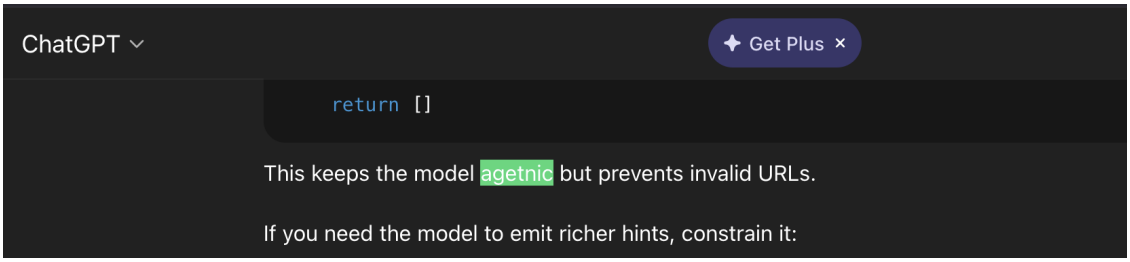



Figure 1: Example of GPT-5 undergoing typo induction where context overrides internal priors.

However, knowing the conditions of when typo induction occurs is currently unknown. In addition, we do not know where in the model's layers does the induction overpower the correct spelling. This paper attempts to look at and answer these questions as well as analyze if modifications can be made to the layers to change model behavior (in this case to either have correct spelling or promote incorrect spelling).

## 3 Methods

After observing the typo induction phenomenon on much larger models like GPT 5, the goal was to replicate typo induction on smaller, more compute friendly models (GPT2-Small {124M params}, GPT2-Large {774M params}, and Pythia 2.8B params), detailed in the next experiment.

In order to test a model's typo induction capability, the setup was simple. From a word-typo-example dataset, a prompt with instances of a *target* word in them would have a few instances of the *target* word in the prompt replaced with **a** singular typo version of that word. This corrupted prompt (minus the last instance of the target word) would then be fed into the model, and the probability of generating the typo version of the word was measured via logit difference (the log probability of generating the typo version subtracted by the log probability of generating the correct version).*

This key metric, logit difference, is going to be used numerous times throughout the work to measure the likelihood of a model generating the typo version as opposed to the correct version. One **incredibly** important thing to note is that when this metric is above 0, the LLM is more likely to generate the typo version as opposed to the correct version. As such, all graphs with this metric on the y-axis have an indicator line at 0 to represent when the typo wins out.

To accurately capture model behavior across a wide range of words, typos, and prompt lengths, a database of typo susceptible words and typo insusceptible words was generated using Gemini, due to time constraints. The prompt used was:
"What are common misspelled words and their misspellings? Include sentences that use these words frequently for each example."

The dataset was then organized via objects, where each object represents a word, its typo, and sentences using the word. The dataset is listed Table 1 (example sentences omitted for brevity).

Table 1: Selection of target words and their corresponding typos.

| **Misspelling Type** | **Word** | **Typo** |
|---|---|---|
| Frequent | accommodate | acommodate |
| Frequent | environment | enviornment |
| Infrequent | February | Febuary |
| Infrequent | calendar | calender |
| Frequent | government | goverment |
| Frequent | occurrence | occurence |
| Infrequent | definitely | definately |
| Frequent | because | becuase |
| Frequent | receive | recieve |
| Frequent | maintenance | maintanance |

# 4 Experiments & Results

## 4.1 Experiment 1: Typo Tipping

Points

With the dataset now defined, the tipping point graphs (graphs that show the percentage where a model *tips* over and begins outputted typos instead) were generated based on this dataset. In order to get a wide range of typo percentages (as well as inference samples), every possible subset of sentences containing at least 2 instances of the target word was put through the process described above to calculate the logit difference. The process was repeated for 10 trials to average out variance from random positioning of typos. The Tipping Point graphs for words are illustrated in Figure 2 and 3 (Pythia 2.8B):

Each datapoint is also colored by the last position of the typo in the input prompt with yellower representing more recent typos and being more likely to generate typos as a trend (across all model types and words), though the relationship was discovered to be only *very* weakly positive.

Another important point is that the tipping point for the same words can vary across model architectures. We hypothesize this to occur due to differences in tokenization, slight differences in training data, and stronger induction mechanisms in larger scale models.

In Figure 4, the comparisons across models is made. Of particular interest is how the majority of points are below the baseline of 0 for GPT2-Large, but above it for Pythia 2.8B. Thus, model size can have a differing effect on typo tipping (possibly making the induction mechanisms more strong, though this could also be due to differences in training / architecture).

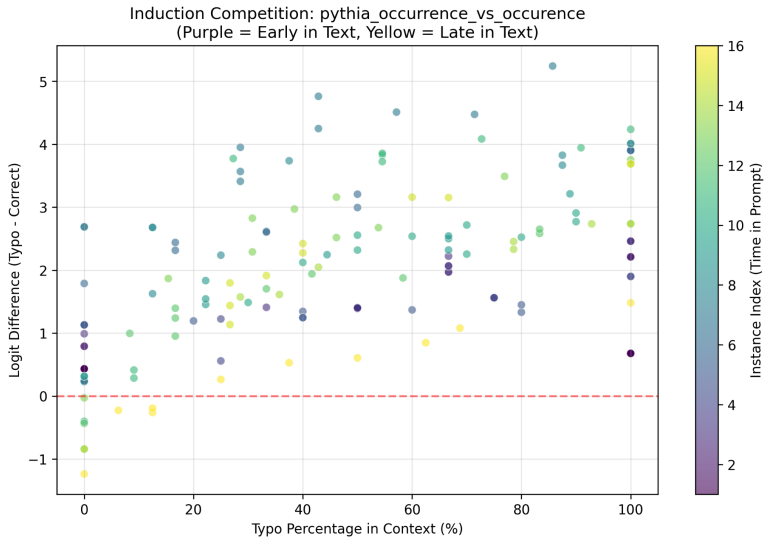



Figure 2: Tipping point graphs for a frequently misspelled word 'occurrence' for Pythia 2.8B.

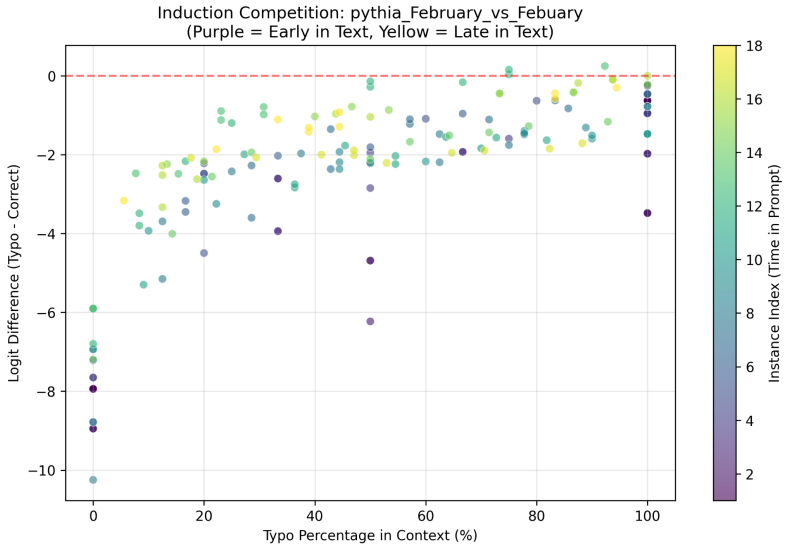



Figure 3: Tipping point graphs for a infrequent mispelled word 'February' for Pythia 2.8B.

## 4.2 Experiment 2: Layerwise and Headwise Analysis

Once we had replicated the phenomenon across various model types and understood the factors that went into inducing typos in LLM outputs, we wanted to identify where specifically in models this occurs. To do this, we used 2 main methods: logit lensing and activation patching. We did this on both Pythia 2.8B

In the first half of this experiment, we wanted to identify which layers were making the typo vs correct spelling decision, so we did both analysis methods (logit lensing and activation patching via a hook function) on all layers. We experimented with various typo percentages for input prompts, before landing on graphing a variety of them at once to see behavior across various percentages.

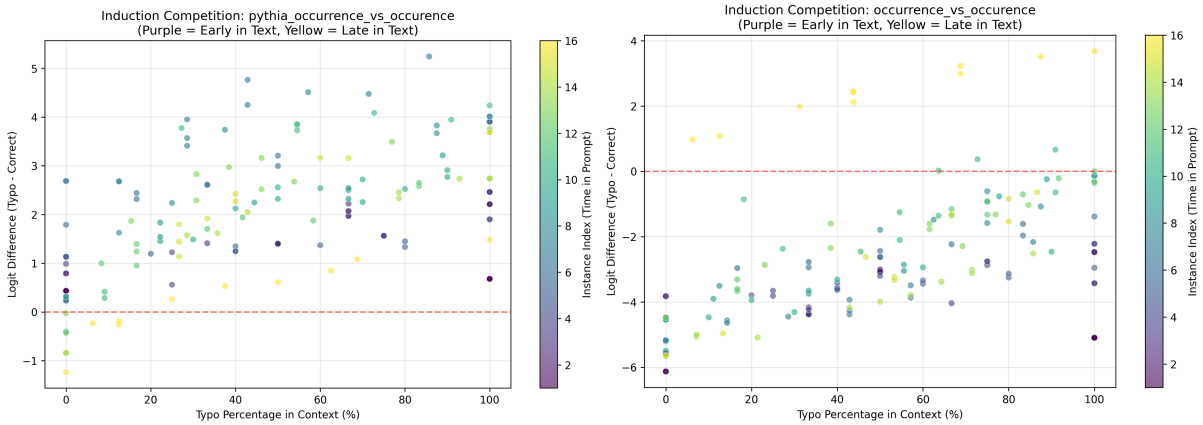



Figure 4: Tipping point comparisons for occurrence between Pythia 2.8B (left) and GPT2-Large (right), approximately 0.8B

In Figure 5, it can be seen that higher typo percentages shift the curve over the 0 baseline and into typo generation (furthering the trend from Experiment 1).

These curves, while illustrative, however, were inconsistent in shape across both words and model architectures.

There were some loose common elements for the logit lensing curves, like a double descent pattern with logit difference dropping twice (once in the early stages, another in the middle) and a final ascent in the late stage. However, this pattern was wildly variable in magnitude and no common timings beyond early-middle-late.

Therefore, because no quantitative analysis was possible, we pivoted to looking into activation patching across every head.

For headwise analysis, we continued with activation patching instead of logit lensing because the

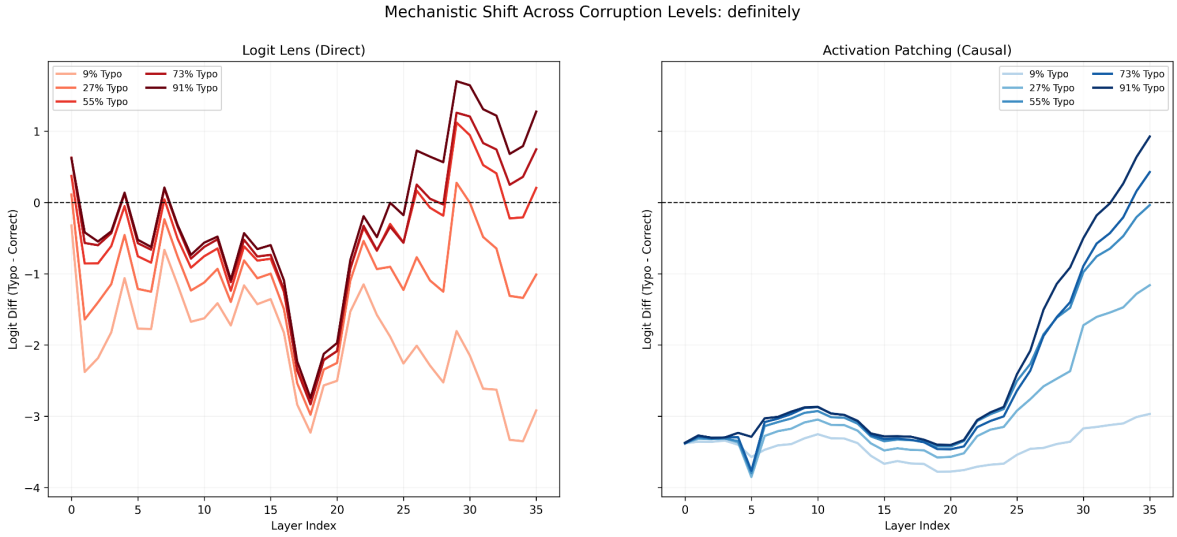
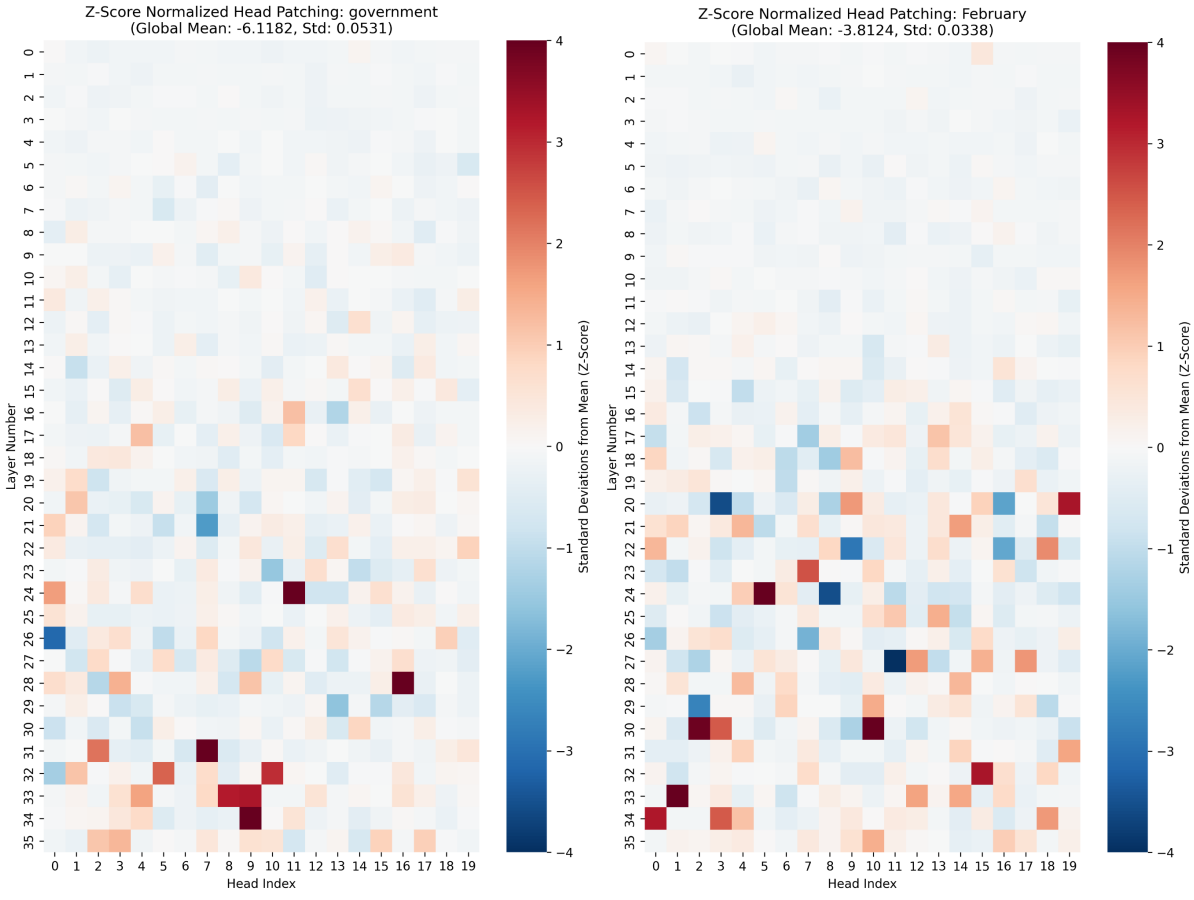



Figure 5: Results for both layerwise analysis methods on 'definitely' for GPT2-Large.

shape of the activation patching curves was more stable/common across words. We patched the activations of a prompt with 100% typos into a clean run's activations and measured the logit difference between the two to identify the most influential heads as a function of both heads and layers.

This can be seen in Figure 6, showing later layer heads have much more influence, though structure is loose.




Activation

Figure 6: Patching heatmaps for a frequently misspelled word 'government' compared to 'February' for GPT2-Large.

In Figure 7, which depicts the same heatmaps for Pythia 2.8B instead, the difference in pattern is much clearer for frequently misspelled words and rarer ones due to the larger scale of the model (and as such, activation patching being more illuminative).

To identify common influential heads (since a lot of heads vary across words in the heatmaps) we used a matchingness score calculation (which weighted consistency in direction and commonality over raw magnitude) and ranked all heads in both direction (positive / negative for increasing / decreasing probability of typo induction), magnitude (how influential it is towards the probability in its respective direction), and matchingness.

The results are shown in Table 2.

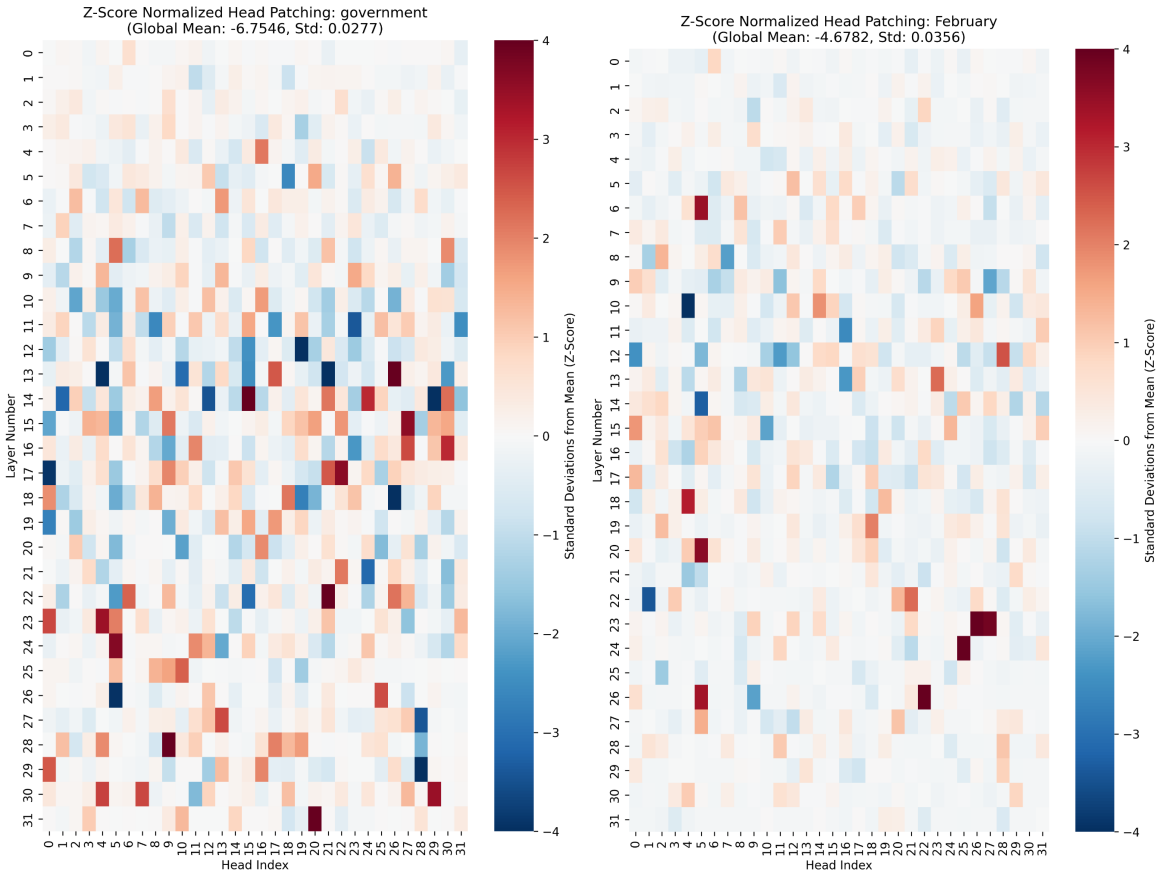



Figure 7: Heatmaps of the same words ('government', 'february') for Pythia 2.8B.

Table 2: Influential Typo Heads for GPT2-Large

| **Layer** | **Head** | **Mean of Standardized Logit Difference** | **Matchingness** |
|---|---|---|---|
| 31 | 3 | -1.79 | 0.48 |
| 27 | 17 | 3.11 | 0.48 |
| 23 | 10 | -1.61 | 0.4 |
| 27 | 8 | -1.49 | 0.4 |
| ... | ... | ... | ... |

These table results also match up with visual analysis of the heatmaps (as Layer 31 Head 3 is an influential head that we also identified with just the heatmaps; though this is a more quantitative measure). This analysis (headwise and influential heads) was also performed on Pythia 2.8B.

### 4.3 Experiment 3: Ablation Study

Once we had identified influential heads in both Pythia 2.8B and GPT2-Large, we tried controlling both model's typo induction behavior by ablating the 2 most significant heads in either direction

for both models and measuring (through logit difference) the effect ablation had.

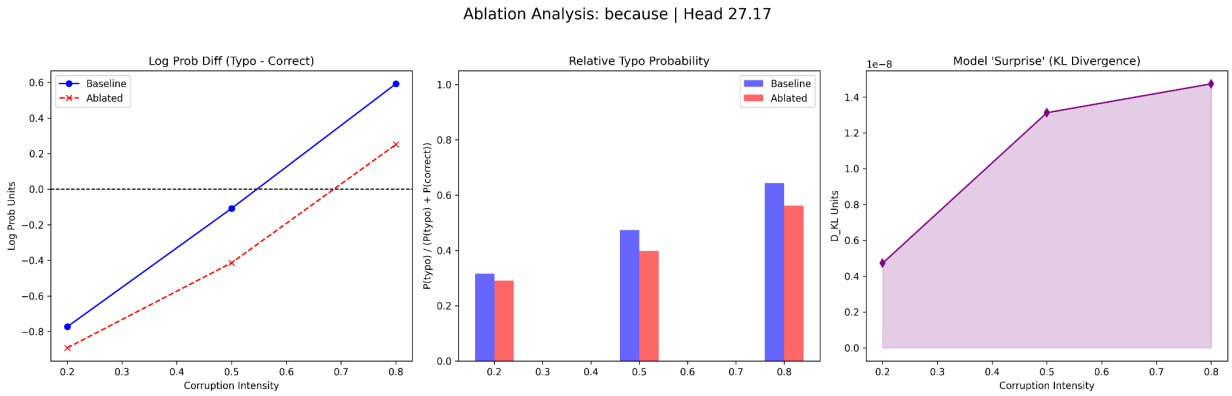



Figure 8: The effect of ablation on GPT2-Large for the word ‘because’.

As you can see in Figure 8, the ablation had a clear effect. Here, Layer 27 Head 17 was ablated (if you refer to the table this is the most influential head *promoting* typo induction), and subsequently, the model's likelihood of outputting the typo dropped markedly. The base probabilities are also shown (not logged) and they also drop. This also occurs in the other direction (ablating a head that *promoted* correct spelling), as seen in Figure 9

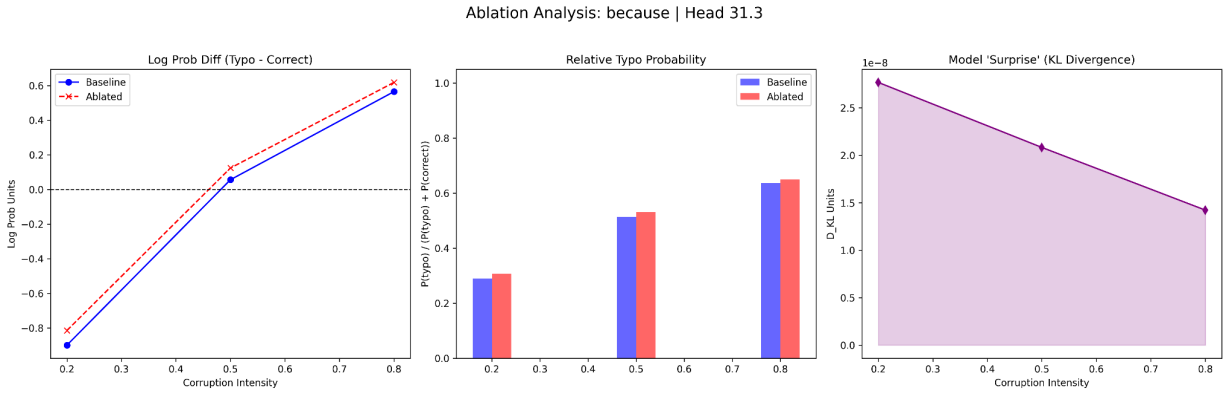



Figure 9: The effect of ablating a head that *discouraged* typo induction on GPT2-Large for the word 'because'.

Though, a weaker effect is measured to this specific head's strength not being as strong an influence as the other one.

One thing to note is that these patterns don't necessarily hold as well across all words (as some words had a slightly lower/higher probability despite ablation) but these were relatively close to the original probability so it's closer to a 0 effect than a negative effect. Furthermore, the strength of ablating individual heads is not too powerful on the full probability. From the heatmaps that the influence is spread out across multiple heads.

While ablating the most powerful one will have a noticeable effect, the decision is still distributed so full control over the model requires ablation of more heads or a better control scheme than just

ablation.

One interesting result was that infrequent words often behaved oppositely from these ablations, though only slightly, as shown in Figure 10.

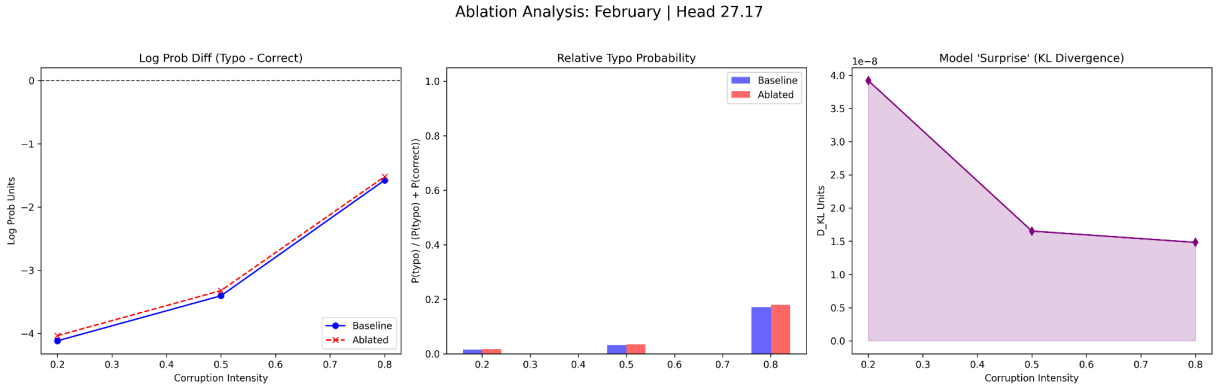



Figure 10: The effect of ablating a typo induction head on an infrequent word, February, for GPT2-Large.

# 5 Discussion:

From the experiments, it's clear to see that typo induction is controlled by 2 battling mechanisms within an LLM, its induction heads that want to copy what they just saw and its internal World Model which knows the correct spelling. The findings can be summarized as so.

## 5.1 Key Findings:

### Typo Induction is influenced by quantity of typos, location, as well as typo type

The single greatest factor relevant to inducing typos in LLM output is the quantity of typos in the input prompt. The more you include typos, the more likely the model is to output one. Interestingly, this relationship is roughly logarithmic w.r.t. the percentage of typos in the prompt (which is how typo quantity is standardized across length). Another factor is the location of the typo. Typos closer to the end of the prompt can more often induce typos in LLM output. The final factor is word type. Some words are not prone to misspellings often (ex: 'Feb***r***uary' vs 'Febuary') and as such leads to different processing by models (as training data does not include a lot of these misspellings). This is difficult to quantify but has an effect visually in graphs and effects downstream experiments.

**On all tested models (GPT2-Large, Pythia 2.8B), the decision for typo vs correct spelling occurs much later than expected**. Layerwise, the decision occurs in the latter half of the mid section and former half of the final layers. The decision occurs, surprisingly, in later layers, compared to "Tracing Typos in LLMs". Furthermore, the decision is influenced by multiple, but concentrated, heads. The heads control both directions (inducting heads that induce typos and statistically likely spelling heads that correct it).

**Ablating the most influential heads can *control model behavior***, increasing typo probability when ablating the "correct spelling heads" and decreasing when ablating the "typo spelling heads", markedly.

However, the investigation had some limitations.

### 5.2 Limitations

The relationships established are **not universal**. The trend shows some relationship but data in all 3 experiments was very noisy and varied wildly across words (even of the same misspelling frequency in both shape and typo probability).

**Compute Restrictions**: All computations involving Pythia 2.8B were done using Google's Colab service. We originally planned to include Mistral's 7B v0.1 model as a bigger scale model to use for data but was limited by compute (even with Google's Colab service with a T4 GPU could not feasibly run the analysis scripts).

**Dataset**: Considering the fact that the prompt dataset is a) *small* (to save inference time and compute) and b) was *generated* by an LLM itself (again a time saving measure), the potential for eccentric results due to a feedback loop and non strange output is possible. In the future, using a greater dataset with more variety is the direction we'd like to go in to certify these results.

## 6 Additional Notes:

The Logit Difference metric, due to words often being split across multiple tokens, actually compares the joint probability of writing the typo'd version of a word to the joint probability of writing the correct version.

The patching heatmaps were normalized in order to standardize comparison, since differing words often had very different impacts on raw logit difference (which couldn't be directly analyzed for head influence).

## 7 Sources Cited:

[1] *"Competition of Mechanisms: Tracing How Language Models Handle Facts and Counterfactuals"*, Ortu et al, 6 June 2024.

[2] *"Tracing Typos in LLMs: My Attempt at Understanding How Models Correct Misspellings"*, Ivan Dostal, 2 February 2025