Mitigating the Language Mismatch and Repetition Issues in LLM-based Machine Translation via Model Editing

Weichuan Wang[♡,♣,*], **Zhaoyi Li**[♡,♠,*], **Defu Lian**[♠], **Chen Ma**[♡], **Linqi Song**[♡,♣,†], **Ying Wei**[◇,†]
[♡]City University of Hong Kong, [♠]University of Science and Technology of China
[♣]City University of Hong Kong Shenzhen Research Institute, [◇]Zhejiang University
weicwang2-c@my.cityu.edu.hk, lizhaoyi777@mail.ustc.edu.cn
liandefu@ustc.edu.cn, {chenma,linqi.song}@cityu.edu.hk, ying.wei@zju.edu.cn

Abstract

Large Language Models (LLMs) have recently revolutionized the NLP field, while they still fall short in some specific down-stream tasks. In the work, we focus on utilizing LLMs to perform machine translation, where we observe that two patterns of errors frequently occur and drastically affect the translation quality: language mismatch and repetition. The work sets out to explore the potential for mitigating these two issues by leveraging model editing methods, e.g., by locating Feed-Forward Network (FFN) neurons or something that are responsible for the errors and deactivating them in the inference time. We find that directly applying such methods either limited effect on the targeted errors or has significant negative side-effect on the general translation quality, indicating that the located components may also be crucial for ensuring machine translation with LLMs on the rails. To this end, we propose to refine the located components by fetching the intersection of the locating results under different language settings, filtering out the aforementioned information that is irrelevant to targeted errors. The experiment results empirically demonstrate that our methods can effectively reduce the language mismatch and repetition ratios and meanwhile enhance or keep the general translation quality in most cases.

1 Introduction

Pre-trained Large Language Models (LLMs) are natural machine translators with in-context learning (Brown et al., 2020; Touvron et al., 2023; Vilar et al., 2023; Bawden and Yvon, 2023; Zhang et al., 2023a), while they still fall behind specialized Machine Translation (MT) systems like NLLB (Koishekenov et al., 2023). Previous studies utilize In-Context Learning (Agrawal et al., 2023) (ICL), instruction tuning (Xu et al., 2023; Alves et al., 2023) and post-editing methods (Jiao et al., 2023; Ki and Carpuat, 2024; Raunak et al., 2023) to improve the translation quality. One further question is: *Are there any specific issues that were ignored in previous studies hindering the LLM-based machine translation from further development?* In this work, we identify two issues in the LLM-based machine translation: *Language Mismatch* and *Repetition* (as shown in Figure 1). We check the occurrence of these errors and find that: (1) they are common errors in the whole translation set (e.g., in the en→de setting, language mismatch occurs in over 40% cases with Zero-Shot prompting); (2) they are severe errors for machine translation systems (e.g., repetition errors usually lead to an over 50% BLEU decrease compared with a standard generation).

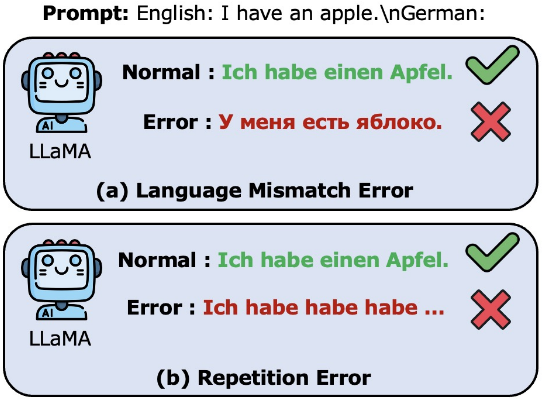

Prompt: English: I have an apple.\nGerman:
LLaMA
Normal : Ich habe einen Apfel. ✓
Error : У меня есть яблоко. ✗
(a) Language Mismatch Error
LLaMA
Normal : Ich habe einen Apfel. ✓
Error : Ich habe habe habe ... ✗
(b) Repetition Error

Figure 1: The illustration of the language mismatch error (a) and the repetition error (b).

Nonetheless, the inherent reason for these errors still remains unclear, let alone patching them. In recent research works on model editing (Dai et al., 2022; Meng et al., 2022; Todd et al., 2023), they typically leverage analyzing tools like causal mediation analysis (Pearl, 2014; Vig et al., 2020), integrated gradient attribution (Sundararajan et al., 2017) to locate important component units (e.g., Feed-Forward Network (FFN) neurons, attention

[*] The first two authors contributed equally to this work. They share the first authorship.
[†] Corresponding authors.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 15681–15700
November 12-16, 2024 ©2024 Association for Computational Linguistics

heads and stuff) that are highly responsible for specific behavior patterns of LLMs, and then precisely control these behaviors by manipulating the located components (e.g., amplifying or suppressing the activation values of neurons). Inspired by these works, we ask a research question: *Can we leverage model editing methods to mitigate aforementioned language mismatch and repetition issues?*

To explore the potential of model editing on mitigating these errors, we set out to adapt two widely-used model editing techniques, Function Vectors (Todd et al., 2023) (FV) and Knowledge Neurons (Dai et al., 2022) (KN), to MT scenarios in an aim to locate error-relevant component units inside LLMs. However, our empirical results show that directly adapting FVs and KNs either has limited effect on the targeted errors or has significant side-effect on the general translation quality, which indicates that the located component units may be not only responsible for targeted error patterns but also crucial for ensuring machine translation with LLMs on the rails and hence directly manipulate them could result in affecting the general translation behavior.

We then aim to filter out the error-irrelevant components from the located results. A possible hypothesis is that *the location for the important error-relevant modules is supposed to be independent of translation language settings.* After comparing the locating results under the different translation language settings (de→en, en→de, zh→en and en→zh), we do observe that a proportion of located component units are shared across different language settings, which valid that the error-related components are highly corresponding to the MT rather than individual languages. Grounded on this observation, we propose to refine the located components by fetching the intersection of the locating results under different language settings. The empirical results across different language settings demonstrate that the modified methods can effectively reduce the language mismatch and repetition ratios and meanwhile keep or enhance the general translation quality in most cases.

Our main contributions are three-fold:

- We identify two patterns of errors in LLM-based MT that frequently occur and badly affect the translation quality: language mismatch and repetition.
- We investigate the potential for leveraging model editing methods (FV and KN) to reduce these errors. We find that directly adapting the editing methods either has limited effect on the targeted errors or has significant side-effect on the general translation quality.
- We propose to refine the located modules by fetching the intersection of the locating results under different language settings. We show that with the refined locating results we could arouse the potential for editing methods to handle the language mismatch and repetition errors and meanwhile enhance or keep the general translation quality in most cases.

Additionally, The performance of our methods could sometimes be comparable with traditional methods that adapt LLMs to MT tasks (e.g., 5-Shot ICL (Agrawal et al., 2023), LoRA (Hu et al., 2022) and Full-FineTuning (Zhang et al., 2023c)) without additional requirements like long-context prompting or fine-tuning. Besides, the proposed methods are compatible with the above techniques for further improvements.

2 Related Work

Large Language Models for Machine Translation One surprising ability of LLMs is that they are natural machine translators with Zero-Shot or One-Shot prompt (Brown et al., 2020; Touvron et al., 2023; Vilar et al., 2023; Bawden and Yvon, 2023; Robinson et al., 2023; Zhang et al., 2023a). However, there is still a gap (Xu et al., 2023) between pre-trained LLM and large-scale NMT systems like NLLB (Koishekenov et al., 2023) on the machine translation task. To bridge this gap, previous studies utilize in-context learning (Moslem et al., 2023; Agrawal et al., 2023; Bawden and Yvon, 2023; Vilar et al., 2023), model tuning (Xu et al., 2023; Alves et al., 2023; Zhang et al., 2023b), and interaction with annotation methods (Jiao et al., 2023; Ki and Carpuat, 2024) to improve the translation quality. Even though LLM has achieved massive success in machine translation (Kocmi et al., 2023a), some of the issues from LLM itself may challenge machine translation, such as Hallucination (Bang et al., 2023). Meanwhile, these problems from LLM are challenging to detect only with MT metrics. Alves et al. (2023) find few-shot tuning can improve the translation quality based on MT metrics (Papineni et al., 2002; Rei et al., 2022a) but detect the machine translation hallucination with a case-based hallucination design. In this work, we detect language mismatch and repetition issues in current LLM-based MT works, which are also found and regarded as errors or hallucinations by some of previous works (Bawden and Yvon, 2023; Alves et al., 2023) but on a case study view.

Locating Based Model Editing Precisely locating a small set of important modules (e.g., neurons (Dai et al., 2022), hidden states (Todd et al., 2023), Multi-Head Self-Attention (MHSA) (Li et al., 2024b) and MLP (Meng et al., 2022) outputs) and editing their values to steer large-scale models toward assumed behaviours (e.g., updating factual associations (Meng et al., 2022; Hase et al., 2023), detoxifying (Wang et al., 2024a), decreasing hallucination (Li et al., 2023), switching languages (Tang et al., 2024) and patching reasoning errors (Li et al., 2024b)) is a recently emerging paradigm. Nonetheless, such techniques are still largely under-explored in the context of MT. In this work, we investigate the potential for adapting two representative locating-based editing approaches (specifically, Function Vectors (Todd et al., 2023) and Knowledge Neurons (Dai et al., 2022)) to the MT scenario to mitigate its two fundamental but crucial issues: language mismatch and repetition (Zhang et al., 2021a).

3 Preliminary

In this section, we detail the data preparation process, including the data source, prompt template, and dataset construction. Additionally, we provide information about the model, the evaluation metrics used to support the ensuing experiments and the model editing methods used in this work.

Data Source We choose three high-resource languages: *English, Chinese, German* which show good performance on MT tasks (Robinson et al., 2023). For the detailed language setting, we include two language pairs: English-Chinese and English-German, and four translation directions: en→de, de→en, en→zh and zh→en (where en, de, zh represent English, German and Chinese, respectively). In the data choice, we use the human-made dataset from general MT tasks of WMT21, WMT22 and WMT23 [1] to ensure both high data quality and flexible data domain. These data make the machine translation approach a real-life usage to help us understand the current state of machine translation using LLMs.

Prompt Template For machine translation tasks, a widely-adopted (Zhang et al., 2023a; Bawden and Yvon, 2023; Vilar et al., 2023) K-Shot In-Context Learning (ICL) prompt template (taking the language setting of en→zh for an example) is:

$$\begin{aligned}&\text{English} : src_1\backslash n\text{Chinese} : tgt_1\backslash n\\&\ldots\\&\text{English} : src_K\backslash n\text{Chinese} : tgt_K\backslash n\\&\text{English} : src_q\backslash n\text{Chinese} :\end{aligned}$$

Where (src_i, tgt_i) refers to the i-th in-context translation exemplar (src_i refers to a sentence of source language and tgt_i refers to the corresponding sentence of target language.). src_q refers to the real sentecne of source language that needs to be translated. We call this prompt template *Lang Prompt* and regard it as the default prompt template for the follow-up experiments in this paper.

Dataset Construction In the data construction part, we construct the $\mathcal{D}_{exps}$ (data from WMT21) to provide the ICL exemplars used in the K-Shot prompt for machine translation tasks. We use the WMT22 data as the $\mathcal{D}_{train}$ to fine-tune a model or locate the crucial parts in an LLM for model editing methods. For the testing and validation, we construct the $\mathcal{D}_{test}$ (data from WMT23) for various modifications (e.g. fine-tuning (Devlin et al., 2019) or model editing methods (Todd et al., 2023; Dai et al., 2022)). (Please refer to Appendix A for detailed dataset information)

Model To support the in-depth exploration and analysis of how the two kinds of errors happen. We use LLaMA2-7B as our backbone language model to implement the machine translation task and further adaptation (Touvron et al., 2023). (We also explore the scaling experiments on LLaMA2-13B with the same data and methods, which can be seen in Appendix H)

Evaluation Metrics For the machine translation metrics, we consider the overlapping-based metrics BLEU (Papineni et al., 2002) and neural-based metrics COMET22DA (Rei et al., 2022a) to evaluate the translation quality (For a detailed toolkit and detection process, please refer to Appendix B).

Model Editing Methods For the concrete model editing methods, we choose Function Vectors (FV) (Todd et al., 2023) and Knowledge Neuron (KN) (Dai et al., 2022): *FV* argues that the key information of a task ($\mathcal{T}$) is compactly represented

[1] https://github.com/wmt-conference/wmt**X**-news-systems. **X**$\in \{21, 22, 23\}$

and transported in a small set of attention heads in LLMs. Then, they utilize the summation of these located head vectors and directly add the integrated vector to the "residual stream" (Elhage et al., 2021) of forwarding computation of Transformer-based (Vaswani et al., 2017) LLMs to help them perform ideal behaviour of task $\mathcal{T}$. *KN* further develops the idea of viewing the Feed-Forward Networks (FFNs) in the Transformer (Vaswani et al., 2017) as key-value memories Geva et al. (2021) (memories can be specific words, specific topics and factual knowledge) and locating a small set of neurons in the FFNs that highly attribute to factual knowledge to manipulate.

4 Language Mismatch and Repetition Error in LLM-MT

In our initial experiments, we observe that LLM-based machine translation struggles with the following two types of common errors. One is **Language Mismatch**, referring to the language of the translation result is not the target language. For example, In the en→zh machine translation, the target language is Chinese while the language of generated sentence is still English. Another is **Repetition**, referring to a substring is generated repeatedly until the end of the generation. To evaluate these errors, we additionally introduce two metrics: **L**anguage **M**ismatch **R**atio (LMR) (the percentage of cases occurring the language mismatch error) and **R**epetition **R**atio (RR) (the percentage of cases occurring the repetition error).

Language mismatch and repetition error are common and crucial After detecting these errors, we first try to provide a quantitive analysis by analyzing the ratio of language mismatch and repetition error in Zero-Shot and One-Shot. For detailed language settings, we consider en→de, de→en, en→zh, and zh→en. We utilize the $\mathcal{D}_{test}$ and $\mathcal{D}_{exps}$ as the test set and prompt exemplar source, respectively. We choose LMR and RR to represent the ratio of language mismatch and repetition in a setting (e.g. en→de (Zero-Shot)). For translation quality evaluation, we choose the BLEU (Papineni et al., 2002) as the metrics since these errors can easily be detected on the word level with a sharp decrease on BLEU or human check. We observe that language mismatch is frequent in Zero-Shot and seldom in One-Shot. Repetition error cases in One-Shot are without language mismatch but combined with language mismatch in Zero-Shot. Based on our observation, we do experiments and analysis in Zero-Shot for the language mismatch and in One-Shot for the repetition error.

To explore the relation between the above errors and translation quality, we split the translation results into four sets to evaluate the BLEU performance after error detection. The four sets include two error sets: *language mismatch set* and *repetition error set*, one *regular set* (where instances without both errors), and one *Origin set* that includes all cases. The results of Table 1 illustrate: (1) the gap between the regular set and the original set shows both language mismatch and repetition error hurt the translation quality; (2) Language mismatch is the main reason for the low performance in Zero-Shot; (3) Even though we observe a low repetition ratio in One-Shot, the gap between repetition set and regular set shows that repetition is a severe error in the original set; (4) The performance gap between regular and error cases indicates a direct way to improve the translation quality by eliminating these errors.

In this section, we run all experiments by using the *Lang Prompt* (Zhang et al., 2023a; Bawden and Yvon, 2023; Vilar et al., 2023) as the default prompt template. Currently, we notice that other prompt templates are used in LLM-based MT research (Bawden and Yvon, 2023; Chowdhery et al., 2023; Brown et al., 2020; Lin et al., 2022b; Wei et al., 2022b). To comprehensively explore these errors, we test other prompt templates with the same data and find these errors again. The only difference is the concrete ratio. This extension experiment further demonstrates the conclusion that *language mismatch and repetition error are common and crucial*. (Detailed experimental setting and results can be found in Appendix C)

5 Can we mitigate language mismatch and repetition via model editing?

In this section, We aim to investigate the potential for leveraging model editing methods (Dai et al., 2022; Meng et al., 2022; Todd et al., 2023) to precisely mitigate the aforementioned two severe issues in MT: language mismatch and repetition. We mainly focus on two widely-used model editing methods: **F**unction **V**ectors (**FV**) (Todd et al., 2023) and **K**nowledge **N**eurons (**KN**) (Dai et al., 2022), for both of them are representative (i.e., Causal Mediation Analysis (Meng et al., 2022; Pearl, 2014) for FV and Integrated Gradient Attri-

Setting	L(↓)	OB(↑)	LB(↑)	RB(↑)
zh→en **(Z)**	0.0486	17.13	8.77	17.60
en→zh **(Z)**	0.3269	16.34	3.13	25.29
en→de **(Z)**	0.4524	12.61	1.65	21.86
de→en **(Z)**	0.0219	35.34	23.23	35.66
Setting	**R**(↓)	**OB**(↑)	**RRB**(↑)	**RB**(↑)
zh→en **(O)**	0.0035	18.87	2.13	19.06
en→zh **(O)**	0.0146	27.78	2.08	29.47
en→de **(O)**	0.0141	24.97	12.64	25.86
de→en **(O)**	0.0018	36.54	6.10	36.71

Table 1: The correlation between error ratio and BLEU. **(Z)** represents the Zero-Shot prompting, and **(O)** represents the One-Shot prompting. **L**: language mismatch ratio; **R**: repetition ratio; **OB**: The BLEU on the original set; **LB**: The BLEU on the language mismatch set; **RRB**: The BLEU on the repetition error set; **RB**: The BLEU on the regular set.

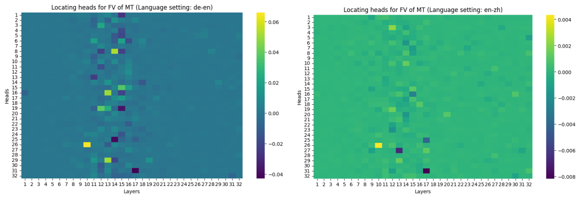
bution (Qi et al., 2019; Lundstrom et al., 2022) for KN) and influential (Bai et al., 2024; Hojel et al., 2024; Niu et al., 2024a; Chen et al., 2024). In the following paragraphs, we adapt the idea of FV (corresponding to **Machine translation vectors**) and KN (corresponding to **Machine translation neurons** and **Repetition neurons**) to MT scenarios, with an aim to both enhance the LLMs' understanding (for both language mismatch and repetition errors) and ability on MT by handling these errors.

5.1 Machine Translation Vectors

FV has demonstrated that it can uncover partial mechanisms of some simplified human-designed tasks by adding a function vector of tasks. But what about a more complex and natural NLP task like MT? To answer this difficult question, we begin with a direct and natural question: ***Can we use FV to enhance LLMs' understanding to MT and mitigate aforementioned language mismatch and repetition issues?*** We use Ten-Shot ICL prompts $\mathcal{P}$ (the template of machine translation prompts is the *Lang Prompt* (Zhang et al., 2023a) in Section 3.) to locate important attention heads, where the data are sampled from $\mathcal{D}_{train}$. For brevity, we denote the normal Ten-Shot ICL input (omitting language signs, i.e., "English", "Chinese" and "German") as: $inp = [(src_1, tgt_1), (src_2, tgt_2), ..., (src_{10}, tgt_{10}), src_q] \in \mathcal{P}$, where src and tgt refer to sentences of source and target languages respectively; index $1 \sim 10$ refers to ten ICL exemplars and q refers to "query" (the real source sentence that requires to be translated.). On its basis, we construct the *shuffled* version of the original ICL input: $\widetilde{inp} = [(src_1, \widetilde{tgt_1}), (src_2, \widetilde{tgt_2}), ..., (src_{10}, \widetilde{tgt_{10}}), src_q]$, where for each ICL exemplar $(src_k, \widetilde{tgt_k})$, $k \in [1..10]$, the target sentence $\widetilde{tgt_k} \neq tgt_k$.

Extracting machine translation vectors First, we locate attention heads that are important to the MT with a *Causal Mediation* procedure: (1) Extract the average attention head output on Ten-Shot cases: $\overline{h_j^i} = \underset{inp \in \mathcal{P}}{\mathbb{E}}[h_j^i(inp)]$, where h_j^i means the i-th head of j-th layer. (2) Send both inp and $\widetilde{inp}$ to the same LLMs (denoting the model as θ), (3) Fetch probabilities of predicting the ground-truth target sentence tgt_q from models with the shaffled input: $p_\theta(tgt_q|\widetilde{inp})$, (4) Adopt intervention: replacing *a single attention head output* in the shuffled run with $\widetilde{inp}$ with the averaged attention head output extracted in step (1) at the same place (h_j^i), (5) Calculate the **C**ausal **I**ndirect **E**ffect (CIE($\overline{h_j^i} \rightarrow h_j^i|inp$)) of the intervention on each Ten-Shot case: $p_\theta(tgt_q|\widetilde{inp}, \overline{h_j^i} \rightarrow h_j^i) - p_\theta(tgt_q|\widetilde{inp})$ and (6) Calculate the **A**verage **I**ndirect **E**ffect for head h_j^i: AIE$(h_j^i) = \underset{inp \in \mathcal{P}}{\mathbb{E}}$[CIE$(\overline{h_j^i} \rightarrow h_j^i|inp)$].

The AIE values for all heads in LLaMA2-7B under the language settings[2] of "de→en" and "en→zh" are depicted in Figure2. We observe that for machine translation there are sparsely a few heads of which the corresponding AIE values strikingly stand out among 1024 heads. We select top-32 heads (the number of heads in a layer and according to their AIE values, denoted as $\mathcal{H}$) to extract FV in the follow-up experiments.



(a) de→en (b) en→zh

Figure 2: Heatmaps of AIE values for attention heads in LLaMA2-7B for de→en setting (a) and en→zh setting (b). x-axis and y-axis refer to the layer and head. Brighter color refers to the head with larger AIE value.

Let $h_j^i(inp)$ denote the output of attention head h_j^i given the input prompt inp. Following Todd et al. (2023), we extract the machine translation

[2] Due to the page limit, We post experiment results only under part of the language settings results in the main text. For the rest language settings, we post them in Appendix D, Similarly hereinafter.

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0486	17.1288	0.722
+*MT vectors*	-72.84%	-37.35%	-1.84%
+*MT neurons*	-18.72%	-4.28%	-0.15%
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0035	18.8714	0.7376
+*MT vectors*	482.86%	-23.07%	-1.68%
+*MT neurons*	0.0%	-0.35%	-0.03%
+*RP neurons*	-8.57%	0.07%	0.0%

Table 2: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language setting of zh→en). ***Zero-Shot*** and ***One-Shot*** refer to that using zero-shot prompt (for language mismatch errors) and one-shot prompt (for repetition errors) for MT. For evaluation metrics, **L**: Language mismatch ratio; **R**: Repetition ratio; **B**: BLEU and **C**: COMET22DA, where **B** and **C** mainly evaluate the general translation quality. For plain LLaMA2-7B, the results are absolute values; for LLaMA2-7B with editing methods, the results are relative **improvement percentages**.

vector with a specific language setting $\mathcal{V}_{\text{X}\to\text{Y}}$ (e.g., $\mathcal{V}_{\text{zh}\to\text{en}}$ means the language setting of zh→en) with the following formula:

$$\mathcal{V}_{\text{X}\to\text{Y}} = \mathop{\mathbb{E}}_{inp\in\mathcal{P}_{\text{X}\to\text{Y}}}[\sum_{h_j^i\in\mathcal{H}} h_j^i(inp)] \qquad (1)$$

Editing LLMs via machine translation vectors We directly add the extracted machine translation vector to the "residual stream" (being aligned with the original FV paper, at 11-th layer for LLaMA2-7B) in the forwarding process. The performance of LLaMA2-7B (e.g., under the language setting of zh→en.) after adopting machine translation vectors are posted in Table 2.

We observe that leveraging machine translation vectors (+*MT vectors*) can (1) reduce the language mismatch errors to a large extent (-72.84%) while simultaneously (2) introduce more repetition errors ($+482.86\%$) and (3) do harm to the general translation quality: -37.35% (Zero-Shot) and -23.07% (One-Shot) for BLEU. (For the results of other language settings, we include them in Appendix E.)

5.2 Machine Translation Neurons and Repetition Neurons

Beyond the original exploration of KN on factual knowledge, we also want to know the potential of KN on MT: ***Can we use KN to locate and manipulate skilled neurons responsible for MT or the repetition error pattern?*** In the MT scenarios, We denote the input prompt inp (also omitting language sign) as $[src_q]$ (Zero-Shot) or $[(src_0, tgt_0), src_q]$ (One-Shot) and the corresponding output as tgt_q, where the (src_0, tgt_0) is the ICL exemplar (sampled from $\mathcal{D}_{exps}$) and (src_q, tgt_q) is the "query", the real case used for locating neurons (sampled from $\mathcal{D}_{train}$) or testing edited models (sampled from $\mathcal{D}_{test}$).

Locating Important Neurons for MT We randomly sample a token t in each tgt_q (without errors) and use t to split tgt_q into two parts: $tgt_q = (\overleftarrow{tgt_q}, \overrightarrow{tgt_q})$ $(t \in \overrightarrow{tgt_q})$. To fully model the MT and meanwhile restrict the computation, we focus on the probability of $p(t|inp^+)$, where t refers to the first token of $\overrightarrow{tgt_q}$ and inp^+ refers to the concatenation of inp and $\overleftarrow{tgt_q}$. Focusing on a single neuron $w_i^{(l)}$ (i-th intermediate neuron in the l-th FFN), we denote its activation value as $\overline{w_i}^{(l)}$. Then we can introduce this variable into $p(t|inp^+)$ as $p(t|inp^+, w_i^{(l)} = \overline{w_i}^{(l)}) \triangleq f(\overline{w_i}^{(l)})$ (fixing t and inp^+, the probability can be viewed as an objective function whose only variable is the value of neuron $w_i^{(l)}$). We calculate the attribution score of neuron $w_i^{(l)}$ by Integrated Gradient (Sundararajan et al., 2017):

$$\text{Attr}(w_i^{(l)}|f) = \overline{w_i}^{(l)} \int_{\alpha=0}^{1} \frac{\partial f(\alpha\overline{w_i}^{(l)})}{\partial w_i^{(l)}} d\alpha. \qquad (2)$$

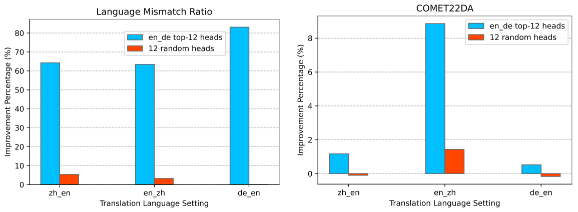
We calculate the mean value of the attribution scores for each neuron with 2,000 examples through Riemann approximation with 20 steps. We select top-5 neurons as Machine Translation neurons (*MT neurons*).

Locating Important Neurons for Repetition We first collect all of examples that occur the repetition error. For a specific input prompt inp, the completed generation y of a LLM can be divided into the following several parts: $y = [y_{norm}, y_{repe}, y_{repe}, y_{rest}]$, where y_{norm} refers to the normal generation part (except for the first-time generation of y_{repe}), y_{repe} refers to the minimal repetition unit (the first y_{repe} here is supposed to be treated as normal generation) and y_{rest} (the follow-up generation after the second-time generation of y_{repe}). To concentrate on the repetition error, we construct a new input prompt $inp_{repe} = [inp, y_{norm}, y_{repe}]$ and focus on the probability of $p(y_{repe}|inp_{repe})$. Similar to the *MT neurons* part, we define neuron $w_i^{(l)}$, its value $\overline{w_i}^{(l)}$, its objective function $p(y_{repe}|inp_{repe}, w_i^{(l)} =$

$\overline{w_i}^{(l)}) \triangleq f_{repe}(\overline{w_i}^{(l)})$ and its attribution score $\text{Attr}(w_i^{(l)}|f_{repe})$ (repetition attribution score). A natural concern here is that ***the objective function*** $f_{repe}(\overline{w_i}^{(l)})$ ***might model the pattern of generating*** y_{repe} ***rather than the repetition error pattern***. To exclude this concern, we additionally set a comparison objective function $f_{compare} = p(y_{repe}|[inp, y_{norm}], w_i^{(l)} = \overline{w_i}^{(l)})$ to model the first-time generation (normal generation) of y_{repe}. With $f_{compare}$, we can also get the attribution score $\text{Attr}(w_i^{(l)}|f_{compare})$ (comparison attribution score) of neuron $w_i^{(l)}$. We calculate the mean values of repetition and comparison attribution scores separately for each neuron $w_i^{(l)}$ with all of the cases in $\mathcal{D}_{train}$ that occur the repetition error. We separately select top-300 neurons according to mean repetition and comparison attribution score, denoting the fetched sets as $\mathcal{N}_{repe}$ and $\mathcal{N}_{compare}$. We select 5 neurons with the largest repetition attribution scores from $\mathcal{N}_{repe} \backslash \mathcal{N}_{compare}$ as the Repetition Neurons (*RP neurons*).

Editing LLMs via *MT* neurons and *RP* neurons For *MT* neurons, we edit LLMs by *amplifying* the activation values of these neurons (set the new values to be twice the original ones). For *RP* neurons, we edit LLMs by *erasing* the activation values of these neurons (set the new values to be zero). The performance of LLaMA2-7B (e.g., under the language setting of zh→en.) after adopting *MT* neurons and *RP* neurons are posted in Table 2. We observe that (1) adopting *MT* neurons can indeed help reduce language mismatch ratio to some extent(-18.72%) while also bring small negative side-effect to the translation quality (-4.28% for the BLEU score), (2) adopting *MT* neurons nearly have no effect on the repetition ratio and (3) adopting *RP* neurons can reduce the repetition ratio slightly (-8.57%) without affecting the metrics (BLEU and COMET22DA) of evaluating general translation quality.

Hence a short response to the question of this section is that ***Directly leveraging model editing methods either has limited effect on errors (MT neurons and RP neurons) or significant negative side-effect on general translation quality (MT vectors)***. Nonetheless, we do observe the potential for mitigating the aforementioned errors with editing methods.



(a) Language Mismatch Ratio (b) COMET22DA

Figure 3: Performance ((a) for the decrease percentage of LMR; (b) for the improvement percentage of COMET22DA) of intervention (blue bars) with language settings of **zh→en**, **en→zh** and **de→en** on the heads located with the language setting of **en→de**. The red bars (comparison group) refer to the results for intervention on random heads of the same number.

6 Modifications to FV and KN in MT scenarios

In section, we mainly discuss our modifications (Section 6.1) to FV and KN methods (Section 5) to release their potential for better mitigating the language mismatch errors, repetition errors and hopefully improving the general translation quality. Besides, we present systematical evaluation results for the modified editing methods and baselines in Section 6.2 to facilitate a deeper understanding of LLM-based MT.

6.1 Modifications

Previous empirical results (Section 5) show that *MT vectors* are more effective to reduce language mismatch errors in comparison with *MT neurons* while the *RP neurons* are more promising for handling repetition errors, suggesting that the inherent mechanisms for the recognition of target language and generating strings repeatedly locate in heads and FFN neurons of LLMs, respectively. To this end, in the follow-up experiments, we concentrate on modifying *MT vectors* to handle language mismatch errors and *RP neurons* to handle repetition errors. Our first modification is based on a natural hypothesis: ***The location for the important modules inside LLMs that are responsible for target language recognition and repetition errors is supposed to be independent to language settings.*** The hypothesis can also be verified to some extent by the important head locating experiments depicted in Figure 2, where results for different language settings (**de→en** and **zh→en**) share a large proportion of top heads. Moreover, we locate top-12 important attention heads in LLaMA2-7B under the language setting of en→de and apply *MT*

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0486	17.1288	0.722
+*MTV*	$\mathbf{-92.46\%}$	-0.81%	2.65%
+*MTV-I*	-80.15%	53.5%	15.51%
+*MTV-I-D*	-86.12%	$\mathbf{76.82\%}$	$\mathbf{16.02\%}$
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0035	18.8714	0.7376
+*RPN*	-8.57%	0.07%	$\mathbf{0.0\%}$
+*RPN-I*	$\mathbf{-25.71\%}$	$\mathbf{0.51\%}$	-0.04%

Table 3: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language settings of zh→en for ***Zero-Shot*** and zh→en for ***One-Shot***). Other notations and abbreviations are following Table 2.

vectors to LLaMA2-7B with these located heads under the language settings of **zh→en**, **en→zh** and **de→en**. The results of Zero-Shot translation are depicted in Figure 3 (experimental group, blue bars). We additionally randomly select 12 heads to apply *MT vectors* and the results (comparison group) are shown with red bars. We observe that for both the language mismatch ratio and COMET22DA[3], the performance of experimental group largely exceeds the performance of comparison group under all three other language settings, indicating that the attention heads located under a single language setting can transfer to other language settings. Given these evidences, we propose our first modification to both *MT vectors* and *RP neurons*: ***We firstly locate attention heads or FFN neurons separately for each language setting and then get the final located results by intersecting the located results for all of language settings.*** We denote the MT vectors fetched by intersected attention heads as ***MT V****ectors*-***I****ntersection* (*MTV-I*) and intersected RP neurons as ***R****e****P****etition* ***N****eurons*-***I****ntersection* (*RPN-I*). We post the results for leveraging *MTV-I* and *RPN-I* under the language settings of en→de and zh→en in Table 3. We observe that: (1) for *MTV-I*, the decrease percentage of language mismatch error ratio (-80.15%) is slightly lower than *MTV* (-92.46%) while improvement percentage of the BLEU score (53.5%) and COMET22DA score (15.51%) exceed *MTV* (-0.81% and 2.65%) by a large margin and (2) for *RPN-I*, the decrease percentage of repetition error ratio (-25.71%) is much higher than RPN (-8.57%), suggesting that intersection of different language settings can filter attention heads and FFN neurons that are irrelevant to language mismatch errors and repetition errors out. On the basis of *MTV-I*, we propose another slight modification: ***Firstly calculate the MTV-I, then divide it evenly according to the number of the intersected attention heads and add them to those heads***. We denote this manner of leveraging *MTV-I* as *MTV-I-Distributional* (*MTV-I-D*). We also post the results of leveraging *MTV-I-D* in Table 3, where the results demonstrate that *MTV-I-D* can further achiever better performance than *MTV-I* in terms of language mismatch ratio, BLEU and COMET22DA.

6.2 Overall Results

To make readers get a better sense of the LLMs edited with our methods (*MTV-I-D* and *RPN-I*), we show the overall evaluation results for both our methods and traditional adaptation methods, including 5-Shot In-Context Learning (Brown et al., 2020) (5-Shot ICL), Low Rank Adaptation Tuning (Hu et al., 2022) (LoRA) and Full parameter Supervised Fine-Tuning (Alves et al., 2023) (Full-FT) for LLM-based MT in Table 4.[4] We post the performance on the metrics of language mismatch error ratio, repetition error ratio and BLEU score (We find that performance on COMET score is highly aligned with BLEU score) and observe that: (1) Applying the modified editing methods, *MTV-I-D* and *RPN-I* can generally reduce the error ratios for both language mismatch (**L**) and repetition (**R**) to a large degree, (2) The negative side-effect on the general translation quality (BLEU score, **B**) is minor (except when applying *MTV-I-D* under the setting of **zh→en**, with a -14.08% decrease percentage on BLEU score). It is noteworthy that applying *MTV-I-D* can even improve the general translation quality to a large extent on the settings of **en→de** (76.82%) and **en→zh** (24.64%) and (3) The performance of *MTV-I-D* and *RPN-I* can sometimes be comparable with (and even surpass) the traditional methods that adapt LLMs to the MT without additional requirements like long-context prompting and fine-tuning.

Besides, we also investigate *whether applying our editing methods to correct the language mismatch and repetition errors will bring negative effects on the general abilities of LLMs?* and *the additional computation cost brought by applying these editing methods*. The detailed empirical results and analysis are presented in Appendix J.

[3] https://huggingface.co/Unbabel/wmt22-comet-da

[4] We train the model under Zero-Shot and One-shot respectively except for Five-Shot ICL, other details can be seen on Appendix G.

	de→en		**en→de**		**zh→en**		**en→zh**	
Zero-Shot	**L(↓)**	**B(↑)**	**L(↓)**	**B(↑)**	**L(↓)**	**B(↑)**	**L(↓)**	**B(↑)**
LLaMA2-7B	0.0219	35.3448	0.4524	12.6084	0.0486	17.1288	0.3269	16.3441
+5-Shot ICL	−74.89%	**4.93%**	−92.06%	101.27%	−50.0%	**12.46%**	**-82.59%**	76.9%
+LoRA	**-83.56%**	0.68%	**-95.25%**	**115.24%**	**-79.22%**	6.62%	−77.58%	**82.62%**
+Full-FT	−8.68%	2.25%	−62.69%	55.41%	−33.33%	3.15%	−66.23%	62.64%
+*MTV-I-D*	−33.33%	−0.53%	−86.12%	76.82%	−54.12%	−14.08%	−69.9%	24.64%
One-Shot	**R(↓)**	**B(↑)**	**R(↓)**	**B(↑)**	**R(↓)**	**B(↑)**	**R(↓)**	**B(↑)**
LLaMA2-7B	0.0018	36.5445	0.0141	24.9685	0.0035	18.8714	0.0146	27.7798
+5-Shot ICL	0.0%	**1.49%**	14.89%	1.63%	−14.29%	2.07%	−17.12%	4.08%
+LoRA	**-77.78%**	−9.47%	**-74.47%**	−2.39%	5.71%	0.07%	−10.27%	0.37%
+Full-FT	22.22%	1.26%	−25.53%	**4.9%**	−22.86%	**2.5%**	22.6%	**4.47%**
+*RPN-I*	−38.89%	0.74%	−27.66%	0.35%	**-25.71%**	0.51%	**-19.18%**	−0.23%

Table 4: Overall Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ under all language settings. Other notations and abbreviations are following Table 2. The **bold** value means the best performance while the underline value represents the worst performance.

For the first point, we demonstrate that our editing methods would hardly hurt (sometimes even boost) the general performance of LLMs on five popular benchmarks (MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2022a), MMLU-Pro (Wang et al., 2024b), CMMLU (Li et al., 2024a)and CommonQA (Talmor et al., 2019)). For the second point, through both running time complexity analysis and empirical statistics, we show that the additional computation overhead brought by our editing methods is marginal. We release our code and data on GitHub[5] for the reproduction and exploration of others.

7 Conclusion

In the work, we find that two types of errors, language mismatch and repetition, occur frequently when performing machine translation with LLMs, bringing severe negative effects on the translation quality. We investigate the potentials of leveraging model editing methods to mitigate these issues and find that directly adopting function vectors and knowledge neurons may either have limited improvement on the identified errors or bring noteworthy negative effect on the general machine translation quality (e.g., BLEU score), which indicates that the located attention heads and FFN neurons might be too coarse to only affect the error ratios without hurting the translation quality. To this end, we propose to refine the located attention heads and neurons by fetching the intersection of the locating results under different language settings. Our empirical results suggest that the modified function vectors and knowledge neurons methods (*MTV-I-D* and *RPN-I*) can effectively reduce the aforementioned two types of errors and generally bring a positive influence evaluated with the translation quality metrics in most settings, indicating that there indeed exist a small set of modules that are highly responsible for the language mismatch and the repetition errors.

Limitations

Our work is based on open-source LLaMA series models LLaMA2[6]. However, the effectiveness of these findings on other models, such as other state-of-the-art open-sourced LLMs (e.g., Mistral-7B[7], OLMO[8] and so on) or the close-sourced LLMs (e.g., GPT-4[9] and Claude-3.5-Sonnet[10]), remains under explored. We leave it for the future work.

The model editing methods used in this paper require computational resources proportional to the size of the LLM. When applying our methods to a larger model, more computational resources will be necessary to achieve improved results. Our focus is on high-resource language settings for machine translation. However, the observations and conclusions may differ when applied to low-resource or non-English language pair settings (e.g., zh→de machine translation)

[5] https://github.com/weichuanW/llm-based-mt-via-model-editing

[6] https://llama.meta.com/llama2/

[7] https://mistral.ai/news/announcing-mistral-7b/

[8] https://allenai.org/olmo

[9] https://openai.com/index/gpt-4/

[10] https://www.anthropic.com/news/claude-3-5-sonnet

We utilise automatic metrics for error and machine translation evaluation in our measurements. However, employing human-involved evaluations (Kocmi et al., 2023b) can offer a more profound understanding of machine translation with LLMs.

Ethics Statement

This paper utilizes a pre-trained LLM, with its training data sourced from web corpora that have not undergone ethical filtering. Consequently, it is capable of generating toxic content in the machine translation (Wen et al., 2023). Moreover, we do not filter the source data or translation output in our work. Future research may build on our results to enhance the model, and we advocate for incorporating content supervision to prevent the dissemination of toxic content.

Acknowledgements

We sincerely thank the anonymous reviewers for their insightful suggestions to this work. The work was supported by grants from the National Key R&D Program of China (No.2021ZD0111801), the Start-up Grant (No. 9610564), the Donations for Research Projects (No. 9229129) of the City University of Hong Kong, and the Early Career Scheme (No. CityU 21219323) of the University Grants Committee (UGC), the Research Grants Council of the Hong Kong SAR under Grant GRF 11217823 and Collaborative Research Fund C1042-23GF, the National Natural Science Foundation of China under Grant 62371411, InnoHK initiative, The Government of the HKSAR, and Laboratory for AI-Powered Financial Technologies.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8857–8873. Association for Computational Linguistics.

Duarte M. Alves, Nuno Miguel Guerreiro, João Alves, José Pombal, Ricardo Rei, José Guilherme Camargo de Souza, Pierre Colombo, and André F. T. Martins. 2023. Steering large language models for machine translation with finetuning and in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 11127–11148. Association for Computational Linguistics.

Yu Bai, Heyan Huang, Cesare Spinoso-Di Piano, Marc-Antoine Rondeau, Sanxing Chen, Yang Gao, and Jackie Chi Kit Cheung. 2024. Identifying and analyzing task-encoding tokens in large language models. *Preprint*, arXiv:2401.11323.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 675–718. Association for Computational Linguistics.

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*, pages 157–170. European Association for Machine Translation.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17817–17825.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

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.

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: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1:1.

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.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Alberto Hojel, Yutong Bai, Trevor Darrell, Amir Globerson, and Amir Bar. 2024. Finding visual task vectors. *Preprint*, arXiv:2404.05729.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Maor Ivgi, Ori Yoran, Jonathan Berant, and Mor Geva. 2024. From loops to oops: Fallback behaviors of language models under uncertainty. *arXiv preprint arXiv:2407.06071*.

Wenxiang Jiao, Jen-tse Huang, Wenxuan Wang, Zhiwei He, Tian Liang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Parrot: Translating during chat using large language models tuned with human translation and feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15009–15020. Association for Computational Linguistics.

Dayeon Ki and Marine Carpuat. 2024. Guiding large language models to post-edit machine translation with error annotations. *CoRR*, abs/2404.07851.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popovic, and Mariya Shmatova. 2023a. Findings of the 2023 conference on machine translation (WMT23): llms are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation, WMT 2023, Singapore, December 6-7, 2023*, pages 1–42. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023b. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Yeskendir Koishekenov, Alexandre Berard, and Vassilina Nikoulina. 2023. Memory-efficient NLLB-200: language-specific expert pruning of a massively multilingual machine translation model. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3567–3585. Association for Computational Linguistics.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024a. CMMLU: measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics, ACL*

2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024, pages 11260–11285. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024b. Understanding and patching compositional reasoning in llms. *Preprint*, arXiv:2402.14328.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022a. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2022b. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 9019–9052. Association for Computational Linguistics.

Daniel D Lundstrom, Tianjian Huang, and Meisam Razaviyayn. 2022. A rigorous study of integrated gradients method and extensions to internal neuron attributions. In *International Conference on Machine Learning*, pages 14485–14508. PMLR.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive machine translation with large language models. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*, pages 227–237. European Association for Machine Translation.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. 2024a. What does the knowledge neuron thesis have to do with knowledge? In *The Twelfth International Conference on Learning Representations*.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. 2024b. What does the knowledge neuron thesis have to do with knowledge? *CoRR*, abs/2405.02421.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Judea Pearl. 2014. Interpretation and identification of causal mediation. *Psychological methods*, 19(4):459.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Zhongang Qi, Saeed Khorram, and Fuxin Li. 2019. Visualizing deep networks by optimizing with integrated gradients. In *CVPR workshops*, volume 2, pages 1–4.

Vikas Raunak, Amr Sharaf, Yiren Wang, Hany Hassan Awadalla, and Arul Menezes. 2023. Leveraging GPT-4 for automatic translation post-editing. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 12009–12024. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022a. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022b. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nathaniel R. Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. Chatgpt MT: competitive for high- (but not low-) resource languages. In *Proceedings of the Eighth Conference on Machine Translation, WMT 2023, Singapore, December 6-7, 2023*, pages 392–418. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. *Preprint*, arXiv:2402.16438.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2023. Function vectors in large language models. *CoRR*, abs/2310.15213.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George F. Foster. 2023. Prompting palm for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15406–15427. Association for Computational Linguistics.

Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024a. Detoxifying large language models via knowledge editing. *Preprint*, arXiv:2403.14472.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *CoRR*, abs/2406.01574.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. 2023. Unveiling the implicit toxicity in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1322–1338, Singapore. Association for Computational Linguistics.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024a. Do llamas work in english? on the latent language of multilingual transformers. *Preprint*, arXiv:2402.10588.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024b. Do llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, Bangkok, Thailand. Association for Computational Linguistics.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *CoRR*, abs/2309.11674.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023b. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *CoRR*, abs/2306.10968.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023c. Instruction tuning for large language models: A survey. *CoRR*, abs/2308.10792.

Ying Zhang, Hidetaka Kamigaito, Tatsuya Aoki, Hiroya Takamura, and Manabu Okumura. 2021a. Generic mechanism for reducing repetitions in encoder-decoder models. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1606–1615, Held Online. INCOMA Ltd.

Ying Zhang, Hidetaka Kamigaito, Tatsuya Aoki, Hiroya Takamura, and Manabu Okumura. 2021b. Generic mechanism for reducing repetitions in encoder-decoder models. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Held Online, 1-3 September, 2021*, pages 1606–1615. INCOMA Ltd.

A Dataset Information

All data used in this work are human-checked from the WMT conference to ensure the data quality. Table 5 shows the detailed data size for $\mathcal{D}_{exps}$, $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$. We use the WMT21 test set[11] as the $\mathcal{D}_{exps}$, WMT22 test set[12] as $\mathcal{D}_{train}$ and WMT23 test set[13] as$\mathcal{D}_{test}$.

Setting	$\mathcal{D}_{exps}$ Size	$\mathcal{D}_{train}$ Size	$\mathcal{D}_{test}$ size
en→de	1002	2037	557
de→en	1000	1984	549
en→zh	1002	2037	2074
zh→en	1948	1875	1976

Table 5: Data size of $\mathcal{D}_{exps}$, $\mathcal{D}_{train}$, $\mathcal{D}_{test}$ on four language settings.

The detailed data size for the K-shot ($K = 0, 1, 5$) setting is shown in Table 6. For all settings, we use the *lang prompt* as the prompt template (as shown in Section 3). For the Zero-Shot setting, we directly combine the source data with the *lang prompt*. For the One-Shot setting, we uniformly sample the data from $\mathcal{D}_{exps}$ based on the length of the example source to alleviate the potential length bias from prompt example (Zhang et al., 2023a). We use the most natural selection method for the Five-Shot setting by randomly selecting five examples from $\mathcal{D}_{exps}$.

Setting	$\mathcal{D}_0$ Size	$\mathcal{D}_1$ Size	$\mathcal{D}_5$ size
en→de ($\mathcal{D}_{train}$)	2037	12222	2037
de→en ($\mathcal{D}_{train}$)	1984	9920	1984
en→zh ($\mathcal{D}_{train}$)	2037	12222	2037
zh→en ($\mathcal{D}_{train}$)	1875	11250	1875
en→de ($\mathcal{D}_{test}$)	557	3342	557
de→en ($\mathcal{D}_{test}$)	549	2745	549
en→zh ($\mathcal{D}_{test}$)	2074	12444	2074
zh→en ($\mathcal{D}_{test}$)	1976	11856	1976

Table 6: Data size of Zero-Shot ($\mathcal{D}_0$), One-Shot($\mathcal{D}_1$) and Five-Shot($\mathcal{D}_5$) on four language settings. $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ represent the source data in the prompt.

B Toolkits for evaluation

We use spBLEU (Post, 2018) and COMET22DA (Rei et al., 2022a) from huggingface API[14] to evaluate MT quality. For the language mismatch detection, we use the Polyglot toolkit[15] to detect the language error. For repetition error, based on the definition of repetition error, we follow two rules to judge whether a translation result is repeated: (1) the generation length reaches the max_new_tokens setting[16] (We use 400 in our work); (2) there exists a substring happening until the end of the generation. For the machine translation metrics, we use SacreBLEU (Post, 2018), Unbabel/wmt22-comet-da[17] and Unbabel/wmt22-cometkiwi-da[18] to do evaluation.

C Comprehensive exploring of language mismatch and repetition errors

When leveraging LLM for translation, prompts matter a lot. To comprehensively understand the effect of prompt templates, we include multiple templates and with the following formats:

In Table 7, we include six templates, including the template used in this paper (Ours (Vi-

[11] https://github.com/wmt-conference/wmt21-news-systems

[12] https://github.com/wmt-conference/wmt22-news-systems

[13] https://github.com/wmt-conference/wmt23-news-systems

[14] https://huggingface.co/docs/evaluate/index

[15] https://github.com/aboSamoor/polyglot

[16] https://github.com/huggingface/tokenizers

[17] https://huggingface.co/Unbabel/wmt22-comet-da

[18] https://huggingface.co/Unbabel/wmt22-cometkiwi-da

Template	Prompt
Ours	L1: src_q \n L2:
Temp1	Given the following source text: src_q, a good L2 translation is:
Temp2	If the original version says src_q then the L2 version should say:
Temp3	What is the L2 translation of the sentence: src_q?
Temp4	L1: src_q = L2:
Temp5	src_q translates into L2 as:

Table 7: Different template formats for Zero-Shot Settings. *L1* and *L2* represent for the source language and target language respectively. src_q means the source sentence to be translated.

lar et al., 2023)). And other five templates from PaLM (Chowdhery et al., 2023), PaLM (Chowdhery et al., 2023), GPT-3 (Brown et al., 2020), XGLM (Lin et al., 2022b) and CoT prompting (Wei et al., 2022b) respectively. For the One-Shot settings, we use a similar way of *Lang Prompt* in Section 3 by adding exemplars to the same template as the One-Shot prompt. We further evaluate the Language Mismatch Ratio (Zero-Shot) and Repetition Ratio (One-Shot) of LLaMA2-7B (Touvron et al., 2023) under the en→zh setting. Table 8 demonstrates that the identified errors are general even if we change the prompt format.

Template	L↓	R↓
Ours	0.3269	0.0146
Temp1	0.2864	0.0133
Temp2	0.5598	0.0134
Temp3	0.9904	0.0161
Temp4	0.3896	0.0138
Temp5	0.6962	0.0127

Table 8: The language mismatch rato and repetition ratio on six different templates of en→zh under Zero-Shot Settings and One-shot Settings, respectively. **L**: language mismatch ratio; **R**: repetition ratio.

In this paper. We demonstrate that our detected MT vector and MT neurons are effective for our default setting across different language settings. Another interesting question is: ***Are our detected MT vector, MT neurons and our proposed methods general if we change the prompts?***

Table 9 can answer this question to some extent. The first surprising thing is that they are generally effective despite prompt shifts. For example, **Temp3** has a nearly total failure in language mismatch (almost 100% mismatch), while our methods can still improve this error with 8.56%. Besides, our detected repetition neurons are pretty helpful for the repetition error across different prompt templates. Another surprising thing is that although we use our prompt setting directly to extract the MT vector or repetition neurons, it is effective on all prompts. This indicates that these detected key components are highly related to MT rather than prompts. A further study on the intersection of MT vectors and repetition neurons among different template prompts will be one of our next steps.

Template	L↓	MTV-I-D	R↓	RPN-I
Temp1	0.2864	-79.80%	0.0133	-49.40%
Temp2	0.5598	-72.18%	0.0134	-56.29%
Temp3	0.9904	-8.56%	0.0161	-47.00%
Temp4	0.3896	-53.47%	0.0138	-55.81%
Temp5	0.6962	-60.66%	0.0127	-60.75%

Table 9: The effect of our proposed methods on other prompts on en→zh setting. Other notations and abbreviations are following Table 2.

D The AIE values for all heads

For a comprehensive observation and validation of our independent assumption for machine translation heads. We show all language setting results in Figure 4. The AIE values of all heads of LLaMA2-7B include en→de, de→en, en→zh and zh→en settings. We also include LLaMA2-13B, a larger LLM than LLaMA2-7B in the same LLM family. Figure 5 shows the AIE values of all heads of LLaMA2-13B.

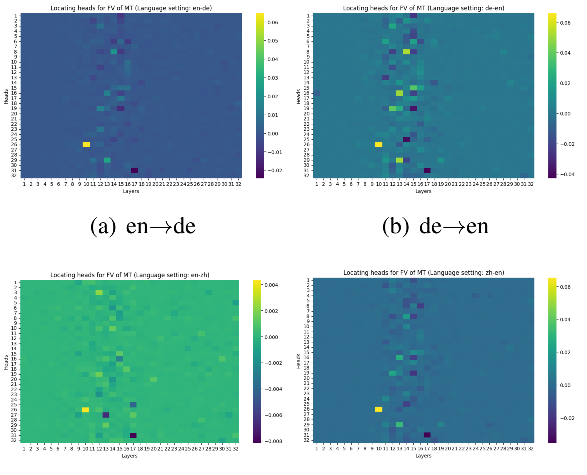
We can observe sets of overlapped heads on both LLaMA2-7B and LLaMA2-13B. This indicates that our detected MT heads may be a crucial mechanism for LLMs when processing MT. Another valuable information from these figures is the locations of these heads are on the middle layers, which can also be found and explored by other current multilingual research (Wendler et al., 2024a,b). Further analysis and concrete results are shown in the Appendix H.

Figure 5 shows the AIE values of all heads of LLaMA2-13B on en→de, de→en, en→zh and zh→en settings.

E Results for direct adaptation

The complete results of direct adaptation on four language settings are shown in Table 10 (en→de), 11 (de→en), 12 (en→zh) and 13 (zh→de).

These tables show that the MT vectors can decrease the language mismatch ratio while the RP neurons help decrease repetition errors in all language settings. One notable case is on Table 11, where we do not extract the *RP neurons* since there



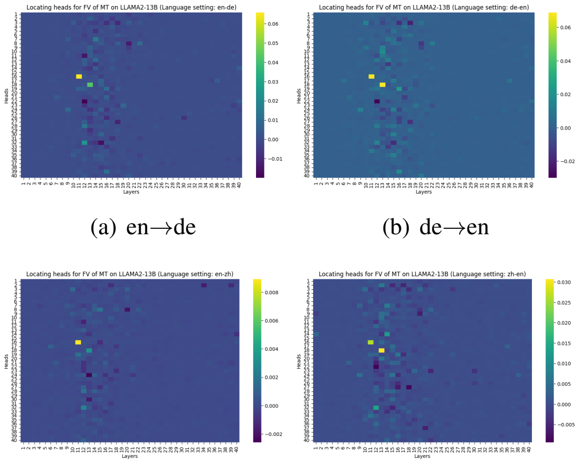
(a) en→de

(b) de→en

(c) en→zh

(d) zh→en

Figure 4: Heatmaps of AIE values for attention heads in LLaMA2-7B for en→de setting (a), de→en setting (b), en→zh setting (c) and zh→en setting (d). The x-axis and y-axis refer to the layer and head, respectively. Brighter color refers to the head with larger AIE value.



(a) en→de

(b) de→en

(c) en→zh

(d) zh→en

Figure 5: Heatmaps of AIE values for attention heads in LLaMA2-13B for en→de setting (a), de→en setting (b), en→zh setting (c) and zh→en setting (d). The x-axis and y-axis refer to the layer and head, respectively. Brighter color refers to the head with larger AIE value.

is no repetition error on the $\mathcal{D}_{train}$. However, we do observe repetition errors when applying the data sourced from $\mathcal{D}_{test}$. This phenomenon indicates that sometimes, the error is easily ignored when we use out-of-date data or a small set of data to check. Our next step is to improve the robustness of our proposed method by exploring the potential properties of these error prompts on a data view.

F Results for improved adaptation

Table 15, 14, 17 and 16 show the results for improved adaptation on en→de, de→en, en→zh and zh→en respectively. Our proposed *RPN-I* generally show stable improvements in all language settings. For Table 14, we skip the **RPN** and **RPN-I** since we do not detect repetition errors with $\mathcal{D}_{train}$. Even though we observe stable improvements across all language settings on the language mismatch when applying our proposed *MTV-I-D*, an inevitable decrease happens on X→en (X refers to de or zh in our work) on any machine translation heads application. Considering LLaMA2 is an English-centric large language model, we think English may not only be used as language recognition but also for other potential mechanisms like general concepts (Wendler et al., 2024a,b). This interesting phenomenon can be connected to multilingual

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.4524	12.6084	0.6113
+MT vectors	−92.46%	−0.81%	2.65%
+MT neurons	−11.1%	1.78%	0.15%
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0141	24.9685	0.7279
+MT vectors	487.94%	−39.11%	−10.87%
+MT neurons	4.26%	−1.05%	−1.06%
+RP neurons	−27.66%	0.77%	−0.3%

Table 10: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language setting of **en→de**). ***Zero-Shot*** and ***One-Shot*** refer to using a Zero-Shot prompt (for language mismatch errors) and one-shot prompt (for repetition errors) for MT tasks. For evaluation metrics, **L**: Language mismatch ratio; **R**: Repetition ratio; **B**: BLEU and **C**: COMET22DA, where **B** and **C** mainly evaluate the general translation quality. For plain LLaMA2-7B, the results are absolute values; for LLaMA2-7B with editing methods, the results are relative **improvement percentages**.

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0219	35.3448	0.7836
+MT vectors	−74.89%	−33.85%	−5.53%
+MT neurons	8.22%	0.03%	0.23%
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0018	36.5445	0.7893
+MT vectors	727.78%	−33.62%	−4.38%
+MT neurons	22.22%	−0.35%	−0.11%
+RP neurons	$--\%$	$--\%$	$--\%$

Table 11: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language setting of **de→en**). The $--$ means the same result as the LLaMA2-7B since we do not detect any repetition on the training set under the same language setting. Notation and corresponding explanations can refer to Table 10.

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.3269	16.3441	0.6567
+MT vectors	−70.05%	18.2%	5.07%
+MT neurons	−5.32%	3.16%	0.35%
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0146	27.7798	0.7444
+MT vectors	162.33%	−15.29%	−4.0%
+MT neurons	5.48%	−4.28%	−0.28%
+RP neurons	−4.11%	0.55%	0.05%

Table 12: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language setting of **en**→**zh**). ***Zero-Shot*** and ***One-Shot*** refer to using a Zero-Shot prompt (for language mismatch errors) and one-shot prompt (for repetition errors) for MT tasks. Notation and corresponding explanations can refer to Table 10.

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0486	17.1288	0.722
+MT vectors	−72.84%	−37.35%	−1.84%
+MT neurons	−18.72%	4.28%	−0.15%
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0035	18.8714	0.7376
+MT vectors	482.86%	−23.07%	−1.68%
+MT neurons	0.0%	−0.35%	−0.03%
+RP neurons	−8.57%	0.07%	0.0%

Table 13: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language setting of **zh**→**en**). ***Zero-Shot*** and ***One-Shot*** refer to using a Zero-Shot prompt (for language mismatch errors) and one-shot prompt (for repetition errors) for MT tasks. Notation and corresponding explanations can refer to Table 10.

research for further exploration.

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0219	35.3448	0.7836
+MTV	**−74.89%**	−33.85%	**0.0036%**
+MTV-I	−58.45%	−4.84%	−5.53%
+MTV-I-D	−33.33%	**−0.53%**	−0.22%
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0018	36.5445	0.7893
+RPN	− − %	− − %	− − %
+RPN-I	−−%	−−%	− − %

Table 14: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language settings of de→en for ***Zero-Shot*** and de→en for ***One-Shot***). The **–** means the results is the same as the LLaMA2-7B since there is no repetition cases in the $\mathcal{D}_{train}$. Other notations and abbreviations following Table 10.

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.4524	12.6084	0.6113
+MTV	**−92.46%**	−0.81%	2.65%
+MTV-I	−80.15%	53.5%	15.51%
+MTV-I-D	−86.12%	**76.82%**	**16.02%**
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0141	24.9685	0.7279
+RPN	−27.66%	**0.77%**	−0.3%
+RPN-I	**−27.66%**	0.35%	**−0.03%**

Table 15: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language settings of en→de for ***Zero-Shot*** and en→de for ***One-Shot***). Other notations and abbreviations following Table 10.

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0486	17.1288	0.722
+MTV	**−72.84%**	−37.35%	−1.84%
+MTV-I	−54.12%	−20.75%	0.0%
+MTV-I-D	−54.12%	**−14.08%**	**0.36%**
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0035	18.8714	0.7376
+RPN	−8.57%	0.07%	**0.0%**
+RPN-I	**−25.71%**	**0.51%**	−0.04%

Table 16: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language settings of zh→en for ***Zero-Shot*** and zh→en for ***One-Shot***). Other notations and abbreviations are following Table 13.

Zero-Shot	**L**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.3269	16.3441	0.6567
+MTV	**−70.05%**	18.2%	5.07%
+MTV-I	−67.27%	19.08%	7.54%
+MTV-I-D	−69.9%	**24.64%**	**8.82%**
One-Shot	**R**(↓)	**B**(↑)	**C**(↑)
LLaMA2-7B	0.0146	27.7798	0.7444
+RPN	−4.11%	0.55%	**0.05%**
+RPN-I	**−19.18%**	0.01%	−0.23%

Table 17: Performance of LLaMA2-7B (and after applying model editing methods) on $\mathcal{D}_{test}$ (under the language settings of en→zh for ***Zero-Shot*** and en→zh for ***One-Shot***). Other notations and abbreviations are following Table 10.

G Implementation Details

For all machine translation results on LLMs, we recognise the end of the generation through the *line break* based on the format design of *lang prompt*. In the real translation process, we use batch generation techniques (batch size = 4) and set the maximum generation length of tokens to 400 with the

	de→en		en→de		zh→en		en→zh	
One-Shot	**R(↓)**	**B(↑)**	**R(↓)**	**B(↑)**	**R(↓)**	**B(↑)**	**R(↓)**	**B(↑)**
LLaMA2-7B	1.0	6.1	1.0	12.64	1.0	2.13	1.0	2.08
+case-RM	0.0%	−46.51%	−35.29%	−18.25%	−28.57%	27.09%	−31.79%	43.7%
+case-FM	−33.33%	−39.75%	−64.71%	−4.92%	−60.71%	148.54%	−64.16%	181.75%

Table 18: Case-editing on Repetition cases. The *case-RM* means we detect the RPN on the first repetition region and try to do modifications to remove the first repetition region. The *case-FM* means we detect the repetition region on the first repetition token only and do the modification.

Huggingface API[19] to do translations for all settings in this work.

Five-Shot For the Five-Shot setting, we directly use the $\mathcal{D}_5$ on LLaMA2-7B to run machine translation task without intervention.

LoRA fine-tuning LoRA (Low-Rank Adaptation) (Hu et al., 2022) is a parameter-efficient tuning technique generally used in natural language processing. In our work, we use the LoRA (Hu et al., 2022) method to align the LLaMA2-7B model to the machine translation task. For the fine-tuning data, we combine the data of all language settings from $\mathcal{D}_{train}$ into $\mathcal{D}_0$ and $\mathcal{D}_1$ for Zero-Shot setting and One-Shot setting respectively (this means we train two different models for Zero-Shot and One-Shot with the same source data from $\mathcal{D}_{train}$). Finally, we tune two LoRA models with the trl tool[20] with the self-supervised tuning method combined with *lang prompt*. We train one epoch with a rank of 64 and a learning rate of $2e^{-4}$ for both Zero-Shot and One-Shot. We use one NVIDIA A100 80GB Tensor Core GPU card for the SFT training; either the Zero-Shot or One-Shot costs less than a half day.

Full fine-tuning We use the same data and training tool in the LoRA setting for full fine-tuning. In the training process, we use the bfloat16 precious to train the model on one NVIDIA A100 80GB Tensor Core GPU card for full fine-tuning with a lower learning rate $1e^{-6}$ compared to LoRA.

We claim that there is still room for improvements in the LoRA or Full fine-tuning methods. However, a complete understanding of the mismatch and repetition error should also be evaluated on large-scale data, which is one of the following steps for our research.

The effect on inherent abilities of LLMs. We use five-shot prompts for all benchmarks except for TruthfulQA (Lin et al., 2022a) where we adopt the zero-shot prompts. All the experiments are implemented based on our codes and the open-sourced LLMs evaluation harness repository.[21]

Detected MT heads and Repetition Neurons. To facilitate the following research on LLM-based MT, we also provide the detailed MT heads and Repetition Neurons detected in this work. For LLaMA2-7B, we find the following overlapped heads after doing an intersection on top-100 AIE heads of each language setting: [[9, 25], [12, 28], [13, 7], [11, 18], [12, 15], [14, 14], [11, 2], [15, 10], [14, 5], [10, 31], [12, 20], [16, 1]], where the first coordinate represents the head index $\in [0, 31]$ and the second coordinate represents the layer $\in [0, 31]$. For the Repetition Neurons, after choosing the top 300 repetition neurons and doing an intersection operation on 2000 cases for each language setting, we get these consistently activated neurons across all the language settings: [6642, 15], [1648, 10], [4531, 8], [5077, 16] and [1392, 7], where the first coordinate represents the neuron index $\in [0, 11007]$ and the second coordinate represents the layer $\in [0, 31]$. We hope these data can accelerate the understanding and exploration of LLM-based MT.

H Scaling Experiments on LLaMA2-13B

The scaling experiments on LLaMA2-13B are shown in Table 19. The *MT-I neurons* is the intersection of the top 100 MT neurons of all language settings (similar refinement like *RPN-I*). We further include COMET22KIWI (Rei et al., 2022b), which is a reference-free evaluation method for a comprehensive evaluation of the translation quality. We can observe similar results for improving the language mismatch error compared with table 4 on LLaMA2-7B. This means our detected Machine Translation heads exist and work in LLMs, and

[19] `https://huggingface.co/`

[20] `https://github.com/huggingface/trl`

[21] https://github.com/EleutherAI/lm-evaluation-harness/tree/main

both heads and vectors matter for MT (the **MTV-I-D** achieves the best performance on improving the language mismatch error). Besides, our distributed MT head intervention method constantly improves the language mismatch issue in all language settings, which shows our finding is general enough for the MT task. Additionally, we also find that the KN method is not stable for improving the repetition error. We think there are two possible reasons. The first reason is the repetition error is much more complicated based on previous findings (Zhang et al., 2021b). The second reason is the KN theory is over-simplified to solve it (Niu et al., 2024b).

I Case-Editing on Repetition Error

We do the case-editing experiment on repetition cases only. Table 18 shows the results. Even though we hope to see some general KN set, KN effectively changes a token rather than a region of errors. In most cases, changing a repeated token can prevent the repetition error, which indicates that the repetition behavior has a connection with some token patterns. We leave this part with current research on repetition (Ivgi et al., 2024)as our future works.

J Other Discussions

The effect on inherent abilities of LLMs One consideration when applying model editing methods to LLMs is their effect on other LLM abilities (Meng et al., 2022; Wei et al., 2022a). Even though we focus on machine translation and extracting corresponding vital components, these components may also be responsible for other potential abilities of LLMs like ICL. To further explore the effect of our proposed methods on other abilities of LLMs, we evaluate the models patched (i.e., edited) by our proposed methods on five popular and representative LLM benchmarks of general abilities. The evluation Benchmarks include MMLU (Hendrycks et al., 2021) (testing the general multitask abilities of LLM), TruthfulQA (Lin et al., 2022a) (testing the truthfulness of LLM, where we use the difficult multi-true task named mc2), MMLU-Pro (Wang et al., 2024b) (a more discriminative benchmark compared to MMLU), CMMLU (Li et al., 2024a) (testing the effect of methods on Non-English languages, i.e., Chinese) and CommonsenseQA (Talmor et al., 2019) (testing the commonsense knowledge of LLM). We take the averaged accuracy measure, following the original papers. Higher values indicate better performance. Table 21 shows nearly no sharp performance drop on these evaluation benchmarks. This suggests that our editing methods do not affect other abilities of LLMs after patching MT-related components. One surprising phenomenon is that sometimes our methods even help improve the performance, as evidenced by the bold values, which could be explained by the enhanced language understanding abilities after editing (For implementation details, please see Appendix G).

Additional computation cost. One advantage of our proposed methods compared to traditional fine-tuning or LoRA is they incur minimal and acceptable computational costs. We do a concrete analysis for the time complexity: For conventional Transformers, the computational complexity for a complete forward pass is $\mathcal{O}((N^2d + Nd^2) \cdot L)$, where N, d, L refer to the input length, hidden dimension and the number of transformer layers. **For the Function Vectors (FV) based editing methods**, we finally select 12 attention heads to manipulate, resulting in an additional computation overhead of $\mathcal{O}(12 \cdot \frac{d}{H}) = \mathcal{O}(\frac{d}{H})$ (head addition takes the $\mathcal{O}(\frac{d}{H})$ complexity, where H is the total number of heads of each layer). In practice, we directly manipulate the head output (after multiplication with the attention output matrix $W_O \in \mathbb{R}^{\frac{d}{H} \times d}$), making the complexity $\mathcal{O}(\frac{d^2}{H})$. **For the Knowledge Neurons (KN) based editing methods**, we select 31 neurons to manipulate, resulting in an additional computational overhead of $\mathcal{O}(31 \cdot d') = \mathcal{O}(d')$, where d' is the up-projection hidden dimension in the middle the MLP module. For LLaMA2, $d' = \frac{8}{3}d$. In practice, we manipulate the MLP output (after multiplication with the down-projection matrix $W_{down} \in \mathbb{R}^{d' \times d}$), making the complexity $\mathcal{O}(d' \cdot d) = \mathcal{O}(\frac{8d^2}{3})$. An optimised implementation can reduce the complexity of the FV editing and the KN editing to $\mathcal{O}(\frac{d}{H})$ and $\mathcal{O}(\frac{8d}{3})$ by only manipulating a single row of the multiplied matrices.

Apart from the theoretical analysis, we run an empirical experiment in the zh→en setting, averaging over 1,000 cases. We calculate the statistical results per token and per case. Table 20 shows that the additional computational overhead brought by our editing methods is marginal.

	de→en				en→de				zh→en				en→zh			
Zero-Shot	**L**(↓)	**B**(↑)	**C-1**(↑)	**C-2**(↑)	**L**(↓)	**B**(↑)	**C-1**(↑)	**C-2**(↑)	**L**(↓)	**B**(↑)	**C-1**(↑)	**C-2**(↑)	**L**(↓)	**B**(↑)	**C-1**(↑)	**C-2**(↑)
llama2-13b	0.0073	38.9229	0.7949	0.7796	0.1472	23.5531	0.7116	0.6911	0.1514	22.3365	0.7244	0.6882	0.0304	19.5333	0.7401	0.7578
+MTV	0.0%	−1.01%	0.1%	−0.13%	7.34%	−0.02%	−0.8%	−0.97%	−8.92%	−12.86%	−1.49%	−1.95%	16.45%	0.1%	−0.05%	−0.01%
+MTV-I	0.0%	−3.96%	−0.03%	−0.53%	−78.06%	7.31%	2.26%	4.24%	−51.25%	−17.4%	−0.25%	0.93%	−35.2%	−5.29%	−0.16%	−0.28%
+MTV-I-D	−50.68%	−1.4%	0.06%	−0.15%	−81.73%	12.75%	5.27%	7.18%	−59.84%	1.48%	2.07%	3.4%	−46.71%	−9.0%	0.14%	0.42%
+MT neurons	0.0%	−0.46%	−0.14%	0.06%	−1.22%	1.96%	−0.42%	−0.39%	1.92%	−0.98%	−0.36%	−0.35%	−5.26%	0.88%	0.18%	0.25%
+MT-I neurons	0.0%	−0.01%	−0.04%	−0.06%	−2.45%	1.51%	0.04%	−0.07%	1.25%	4.95%	0.14%	−0.1%	−8.55%	2.41%	−0.03%	0.03%
One-Shot	**R**(↓)	**B**(↑)	**C-1**(↑)	**C-2**(↑)	**R**(↓)	**B**(↑)	**C-1**(↑)	**C-2**(↑)	**R**(↓)	**B**(↑)	**C-1**(↑)	**C-2**(↑)	**R**(↓)	**B**(↑)	**C-1**(↑)	**C-2**(↑)
llama2-13b	0.0007	39.9956	0.801	0.7849	0.0087	30.0092	0.761	0.7591	0.0128	30.7585	0.7638	0.74	0.0009	21.5548	0.7513	0.7709
+MT neurons	−100.0%	0.13%	−0.01%	0.06%	−17.24%	−0.16%	−0.24%	−0.05%	−12.5%	0.18%	−0.04%	−0.11%	0.0%	1.03%	0.05%	0.14%
+MT-I neurons	−42.86%	−0.01%	0.02%	−0.04%	−3.45%	0.26%	0.08%	−0.09%	−35.16%	3.23%	0.16%	0.08%	55.56%	0.13%	−0.07%	−0.1%
+RPN	−42.86%	−0.23%	−0.04%	−0.06%	−17.24%	0.32%	−0.01%	0.01%	−3.91%	−1.06%	−0.3%	−0.34%	−11.11%	0.41%	−0.03%	−0.03%
+RPN-I	−100.0%	0.25%	0.02%	0.04%	−13.79%	−0.19%	0.05%	−0.03%	0.78%	−1.7%	−0.35%	−0.43%	55.56%	−0.63%	−0.27%	−0.26%

Table 19: Scaling experiments on LLaMA2-13B on $\mathcal{D}_{test}$ under all language settings. For evaluation metrics: **L**: Language mismatch ratio; **R**: Repetition ratio; **B**: BLEU; **C-1**: COMET22DA, **C-2**: COMET22KIWI, where **B**, **C-1** and **C-2** evaluate the general translation quality. Other notations and abbreviations follow Table 2 for detailed methods, where **-I** means the intersection part based on all language settings. We underline the best performance for improving the corresponding errors.

Methods	**MTV-I-D**	**RPN-I**
Per-token (*original*)	0.037s	0.036s
Per-token (*edited*)	0.042s	0.044s
Per-case (*original*)	1.62s	1.33s
Per-case (*edited*)	2.22s	1.70s

Table 20: The computation cost of our proposed methods compared with no-editing on 1000 cases in the zh→en setting. The *original* means using the LLaMA2-7B model only without any modifications. The *edited* means using our proposed MTV-I-D or RPN-I editing. Other notations and abbreviations are following Table 2

Template	MMLU	TruthfulQA	MMLU-Pro	CMMLU	CommonQA
LLaMA2-7B	0.4593	0.3897	0.1860	0.3267	0.5659
+MTV-I-D	**0.4699**	0.3884	**0.1975**	0.3253	**0.5684**
+RPN-I	0.4613	**0.3898**	0.1862	0.3254	0.5659

Table 21: Evaluate our proposed methods on general LLM benchmarks. The results are averaged accuracy for corresponding benchmarks.