Tell Your Model Where to Attend: Post-hoc Attention Steering for LLMs

Qingru Zhang[†][*] **Chandan Singh**[◇] **Liyuan Liu**[◇] **Xiaodong Liu**[◇] **Bin Yu**[‡],
Jianfeng Gao[◇] **Tuo Zhao**[†]
[†]Georgia Institute of Technology [‡]University of California, Berkeley [◇]Microsoft Research
`{qingru.zhang,tourzhao}@gatech.edu` `binyu@berkeley.edu`
`{chansingh,lucliu,xiaodl,jfgao}@microsoft.com`

Abstract

In human-written articles, we often leverage the subtleties of text style, such as **bold** and *italics*, to guide the attention of readers. These textual emphases are vital for the readers to grasp the conveyed information. When interacting with large language models (LLMs), we have a similar need – steering the model to pay closer attention to user-specified information, e.g., an instruction. Existing methods, however, are constrained to process plain text and do not support such a mechanism. This motivates us to introduce PASTA – Post-hoc Attention STeering Approach, a method that allows LLMs to read text with user-specified emphasis marks. To this end, PASTA identifies a small subset of attention heads and applies precise attention reweighting on them, directing the model attention to user-specified parts. Like prompting, PASTA is applied at inference time and does not require changing any model parameters. Experiments demonstrate that PASTA can substantially enhance an LLM's ability to follow user instructions or integrate new knowledge from user inputs, leading to a significant performance improvement on a variety of tasks, e.g., an average accuracy improvement of 22% for LLAMA-7B. Our code is publicly available at `https://github.com/QingruZhang/PASTA`.

1 Introduction

The advent of large language models (LLMs) has marked a significant milestone in natural language processing (NLP) and artificial intelligence (AI), showcasing exceptional performance across a wide range of tasks [Vaswani et al., 2017, Brown et al., 2020a, OpenAI, 2023]. Efforts to further refine these models have been relentless, aiming to enable them to process and respond to natural and programming languages with human-like expertise [Stiennon et al., 2020, Yao et al., 2023].

Despite their remarkable achievements, LLMs often encounter challenges in understanding their contextual inputs during interactions with users [Shen et al., 2023, Lu et al., 2021]. This difficulty becomes particular evident when they are presented prompts[2] containing extensive background contexts or complex user instructions. Lengthy contexts can overwhelm LLMs, as their attention modules, learned from data, are unable to fully capture crucial details [Liu et al., 2023]. Complex instructions can further inhibit the model from focusing on the user's intentions, resulting in undesired outputs [Wei et al., 2022]. Additionally, for time-sensitive data, such as news articles, there can exist factual knowledge within contexts, which contradicts with model prior beliefs induced from outdated pre-training. As a result, a model may generate outputs conditioned on its pre-existing belief instead of attending to new facts within the contexts [Meng et al., 2022a,b, Mitchell et al., 2022, Hernandez et al., 2023]. All of these challenges contribute to LLMs struggling to comprehend user intentions.

[*] Work is done during Qingru Zhang's internship at Microsoft Research.

[2] We use *prompts* to refer to all LLM text inputs, including user instructions, and the other background information (which we refer to as *context*).

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

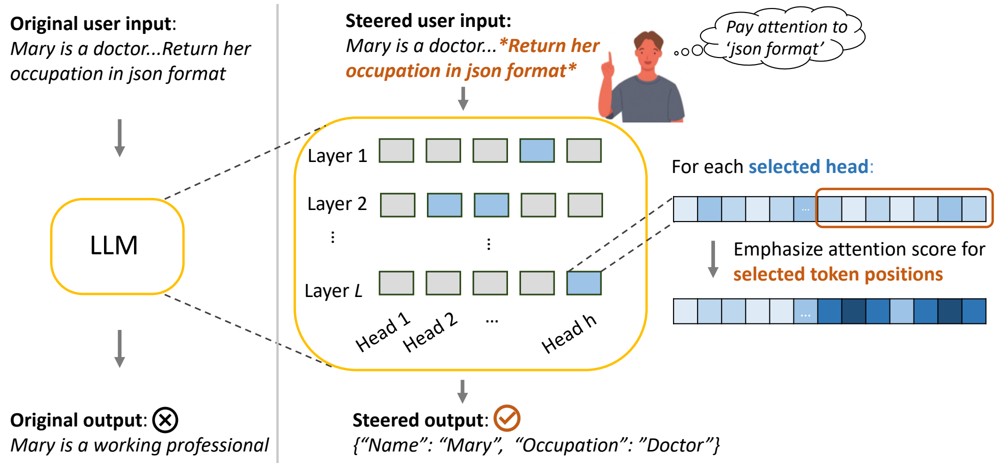

Original user input:
Mary is a doctor...Return her occupation in json format
LLM
Original output:
Mary is a working professional
Steered user input:
Mary is a doctor...*Return her occupation in json format*
Pay attention to 'json format'
Layer 1
Layer 2
Layer L
Head 1
Head 2
...
Head h
For each selected head:
Emphasize attention score for selected token positions
Steered output:
{"Name": "Mary", "Occupation": "Doctor"}

Figure 1: PASTA uses a user-specified part of the input to steer the model generation aligning with user intentions. PASTA modifies the attention scores generated during inference, by emphasizing the scores generated at token positions corresponding to the user-specified part of the context.

Compared to LLMs, human readers rarely struggle to understand the emphases of articles and intentions of writers. Writers often leverage a variety of text styles, such as **bold** and *italics*, to emphasize specific contents. This mechanism enables writers to direct and maintain the attention of human readers, ensuring that the intended information is accurately captured. In interactions between users and LLMs, it is users also need to highlight specific information for the model. Consequently, model generation can be effectively biased in accordance with user guidance, thus addressing the challenges mentioned earlier. This feature is particularly essential when designing user-AI interfaces, and can be frequently applied in extensive conversations between users and models. Existing methods, however, do not support such a mechanism. LLMs are inherently limited to processing plain texts, devoid of any stylistic cues or emphasis markers [Brown et al., 2020b, Liu et al., 2021, Wei et al., 2022]. Even when emphasis markers are added to prompts, state-of-the-art LLMs often struggle to discern weak signals from a couple of marker tokens (See evidence in Section 5.1).

Motivated by the need to convey user emphasis, we introduce *PASTA* (Post-hoc Attention STeering Approach), a post-hoc method[3] that enables users to highlight specific information, e.g., an instruction as in Figure 1, and steer models to interpret emphasized texts like human readers. Specifically, PASTA selects a small subset of attention heads and applies precise attention reweighting on them. As illustrated in Figure 1, PASTA upweights the attention scores of the user-specified tokens while downweighting the other tokens at specific attention heads. Our method is inspired by the observation that attention modules exhibit various token-attending patterns across different heads [Michel et al., 2019, Voita et al., 2019, Clark et al., 2019]. These attention patterns can be interpreted as encoding diverse semantic or syntactic information, and altering them can substantially influence model behaviors [Shi et al., 2023a, Hu et al., 2021b]. Through steering attention modules, PASTA directs the model to pay close attention to the user-specified parts and hence generate the desired output aligning with the highlighted contents. Notably, PASTA is applied after training and does not require changing any model parameters; PASTA only requires access to the attention scores of specific heads of an LLM.

Since attention heads can serve different functions [Tenney et al., 2019, Deb et al., 2023], we introduce an efficient model profiling algorithm to identify which heads are effective for steering. Specifically, we subsample small training sets from multiple tasks and evaluate the performance of attention steering for each individual head across these tasks. PASTA selects the attention heads that, when steered, generally improve the multi-task performance. We empirically observe that steering these heads not only benefits the existing tasks but also enhances the performance on unseen tasks. Notably, the model profiling is performed only once for an LLM. The selected attention heads can be regarded as a model-level profile, effective for steering the LLM on unseen tasks.

We conduct experiments on diverse tasks to demonstrate the effectiveness of PASTA. Specifically, we evaluate PASTA using GPT-J-6B [Wang and Komatsuzaki, 2021] and LLAMA-7B [Touvron et al., 2023] on tasks that span complex instructions, lengthy contexts, and knowledge conflicts within contexts. The results demonstrate that PASTA consistently provides a significant performance

[3] *Post-hoc* means that our method does not update the model weights.

improvement over baseline prompting strategies. For example, PASTA achieve an average accuracy improvement of 22% over few-shot prompting for LLAMA-7B across 4 challenging tasks.

2 Background

Problem description In standard LLM prompting, we are given a pre-trained LLM and a text prompt $\boldsymbol{x}$. In our setting, we additionally require (i) access to attention scores produced by attention modules in the LLM[4] and (ii) we are provided a user-specified subset of the prompt $\boldsymbol{x}_g \subset \boldsymbol{x}$ to be emphasized.

As in the example in Figure 1, $\boldsymbol{x}$ can be a string that ends in an instruction, such as *Mary is a doctor but used to be a nurse...Return her occupation in json format*. If a user emphasizes the instruction, $\boldsymbol{x}_g$ can simply be the final instruction *Return her occupation in json format*. In evaluation datasets, we assume that the user-specified part of each example is already provided by enclosing at its both ends in some emphasis markers, like '$*$' marker in Markdown. Generating these well-structured data often incurs little overhead. For example, in the dataset tailored for evaluting model ability to follow user instruction, we can simply mark the final instruction for every example, which are fixed and shared across examples. When it comes to user-LLM interface, users can specify $\boldsymbol{x}_g$ by enclosing it with the same emphasis markers. $\boldsymbol{x}_g$ can be specified flexibly. Namely, it need not be a continuous span, and can be used to emphasize diverse information.

Multi-Head Attention. A typical transformer model consists of L stacked layers, where each layer contains two submodules: a multi-head attention (MHA) and a fully connected feed-forward network (FFN). Given the input $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, MHA of the layer l performs the attention function in parallel H heads: $\text{MHA}^{(l)}\left(\boldsymbol{X}\right) = \text{Concat}(\boldsymbol{H}^{(l,1)}, ..., \boldsymbol{H}^{(l,H)})\boldsymbol{W}_o$ where

$$\boldsymbol{H}^{(l,h)} = \boldsymbol{A}^{(l,h)}\boldsymbol{V} = \text{Softmax}\left(\boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{d_h}\right)\boldsymbol{V} \tag{1}$$

where $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_{q_h}, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_{k_h}, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_{v_h}$ and $\boldsymbol{W}_{q_h}, \boldsymbol{W}_{k_h}, \boldsymbol{W}_{v_h} \in \mathbb{R}^{d\times d_h}$ are learnable projection matrices of head h. d_h is typically set to d/H. Specifically, denote the attention scores at the head h of the l-th layer as $\boldsymbol{A}^{(l,h)}$.

3 Method

PASTA (Algorithm 1) consists of two components: (i) post-hoc attention steering, which emphasizes the user-specified parts of the input during inference, see Section 3.1 and (ii) multi-task model profiling, which selects the effective attention heads for steering, see Section 3.2.

Algorithm 1 PASTA: Post-hoc Attention Steering Approach

```
Multi-task model profiling (Section 3.2)
Input: small training sets $\{\mathcal{D}^{(i)}\}_{i=1}^m$, the hyperparameters $\alpha$, $k$;
for $1 \le i \le m$ do
    for $1 \le l \le L, 1 \le h \le H$ do
        Evaluate the model performance on $\mathcal{D}^{(i)}$ when steering the head $(l,h)$ by (2);
        Return the evaluation result of steering $(l,h)$ on $\mathcal{D}^{(i)}$;
    end for
    Collect the steering results of all heads and return the task profiling $R^{(i)}$;
end for
Output: The attention head set $\mathcal{H} = \cap_{i=1}^m R^{(i)}_{1:k}$.

Inference-time steering (Section 3.1)
Input: text inputs $\boldsymbol{x}$, user-underlined segments $\mathcal{G}$, coefficient $\alpha$;
Output: the model generations while steering every head $(l,h)$ in $\mathcal{H}$ by (2).
```

3.1 Post-hoc Attention Steering

PASTA emphasizes the user-specified input subset by downweighting the attention scores of tokens that are not specified by the user. Specifically, given the index set of highlighted input spans as $\mathcal{G}$,

[4] We do not need access model weights nor intermediate outputs from the other modules like FFNs.

PASTA emphasizes these user-specified tokens by an attention projection $\mathcal{T}$:

$$\boldsymbol{H}^{(l,h)} = \mathcal{T}(\boldsymbol{A}^{(l,h)})\boldsymbol{V}, \text{ where } [\mathcal{T}(\boldsymbol{A})]_{ij} = \begin{cases} \alpha \boldsymbol{A}_{ij}/C_i & \text{if } j \in \mathcal{G}^- \\ \boldsymbol{A}_{ij}/C_i & \text{otherwise.} \end{cases} \tag{2}$$

where $0 \leq \alpha < 1$ is a scaling coefficient and $\mathcal{G}^- = [n] - \mathcal{G}$ is the index set of tokens that are not in $\mathcal{G}$. The term $C_i = \sum_{j \in \mathcal{G}} \boldsymbol{A}_{ij} + \sum_{j \in \mathcal{G}^-} \alpha \boldsymbol{A}_{ij}$ normalizes the scores so that they sum to one. The attention steering (2) is conducted during the inference time and does not require any training.

(2) steers the model attention by scaling down the scores of tokens that are not highlighted by the user. When the coefficient α is set very small, user-specified segments are highlighted given their increased attention scores after renormalization. Consequently, we can direct the model to concentrate more on the user-specified tokens, biasing the generation to align with the specified contents.

PASTA scales down the attention scores of non-specified tokens by α. As renormalization is followed, it is equivalent to scaling up the attention scores of user-specified tokens by $1/\alpha$. The reason of selecting (2) is that it can be more numerically stable compared to scaling up scores. Alternatively, one can also scale the attention scores by adding a positive constant to the underlined tokens $\mathcal{G}$. The reason of we select multiplication in (2) instead of addition is that it preserves the difference on attention magnitude among the highlighted tokens. As such, the steering operation only adjusts overall attention scales of two groups of tokens. In contrast, addition by a large constant to the highlighted tokens results in their attention scores almost uniformly distributed, leading to unnecessary information loss and performance degeneration.

3.2 Multi-Task Model Profiling

Empirically, we find that applying attention steering in (2) to all attention heads performs worse than applying it only to specific heads (see Section 5.3). It is important to specify the correct attention heads, given that different heads serve distinctive roles in encoding semantic/syntactic information. To this end, we propose a multi-task model profiling algorithm to identify the effective attention heads for steering. Specifically, given m tasks involving user emphases, we subsample a small training set $\mathcal{D}^{(i)}$ (e.g., $|\mathcal{D}^{(i)}| = 1000$) from each task i. Then, we evaluate the performance of steering every individual attention head (l, h) ($1 \leq l \leq L, 1 \leq h \leq H$) on each small subset $\mathcal{D}^{(i)}$ ($1 \leq i \leq m$). For every task i, we rank all of heads according to their steering performance on $\mathcal{D}^{(i)}$ and regard the ranking $R^{(i)} = [(l_1, h_1), (l_2, h_2), \ldots]$ as the profiling of task i. We then set the attention head set $\mathcal{H}$ for steering as the intersection of top-k performing heads, $\mathcal{H} = \cap_{i=1}^{m} R^{(i)}_{1:k}$ (see Section 5.3 for alternative choices). Intuitively, we expect performance to improve as the number of tasks m increases.

Like attention steering, model profiling requires only access to attention scores, in addition to its inputs and outputs (model weights and gradients are not required). Importantly, this process needs to be performed only once for a LLM, similar to finetuning. However, unlike finetuning, model steering does not modify model weights and, more importantly, generalizes to new tasks. The resulting head set $\mathcal{H}$ can be regarded as a model-level profile. Once it is determined, we can apply the attention steering on $\mathcal{H}$ to both existing tasks and unseen tasks to enhance model contextual understanding and benefit downstream performance.

4 Experimental setup

Evaluation tasks and metrics. We implement PASTA for two pre-trained models: GPT-J (6 billion parameters, [Wang and Komatsuzaki, 2021]) and LLaMA-7B (7 billion parameters, [Touvron et al., 2023]). We evaluate the effectiveness of PASTA at (i) handling complex user instructions, (ii) interpreting lengthy contexts, and (iii) resolving in-context knowledge conflicts. For (i), we introduce two new tasks: *JSON formatting* and *Pronouns changing*. For (ii) and (iii), we study *Bias in Bios* [De-Arteaga et al., 2019] and *CounterFact* [Meng et al., 2022a]. For each task, we provide a description, describing which part of the input we emphasize, and what metrics we use for evaluation (see Appendix B for full dataset details).

• *JSON Formatting* is a new task that evaluates an LLM's ability to produce outputs in a user-desired format (JSON). This is an important usecase for LLMs when their output is being used in a downstream process. This task utilizes the biographical data from BiasBios (described below) but appends a different instruction to the end of contexts: *answer the occupation of {person} and* *generate the answer as JSON format*. The instruction prompts models to generate outputs in JSON format.

☞ We emphasize the final instruction

Metrics: (a) **Format accuracy** (F. Acc.) measures the accuracy at generating valid JSON. (b) **Prediction accuracy** (P. Acc.) measures the accuracy at generating the correct target in JSON values after loading the JSON-formatted generations.

• *Pronouns changing* is a new task that evaluates an LLM's ability to follow a difficult user instruction. It again uses the biographical contexts from BiasBios but instead instructs models to: *substitute 'she' and 'he' with 'they' and generate the occupation of {person} after changing pronouns*.

☞ We emphasize the final instruction.

Metrics: (a) **Accuracy** evaluates the ratio that 'she/he' are successfully changed to 'they' in model generations. (b) **All-changed accuracy** (A. Acc.) is the ratio that models replace all corresponding pronouns, i.e., changing she/he/her/him/hers/his to they/them/their/theirs.

• *CounterFact* measures an LLM's ability to generate text consistent with a new fact. Each example consists of (*subject*, *relation*, *old target*, *new target*), e.g., (*Kevin Garnett*, *is a professional*, *basketball player*, *baseball player*). We present the model both old and new facts following the prompt: *Previously, {old fact}, but currently, {new fact}*. {*question*}. This change in facts over time often confuses LLMs, resulting in random guesses on two of them when answering the {*question*}.

☞ We emphasize the input span containing the *new fact*.

Metrics: we evaluate metrics following [Meng et al., 2022a]: (a) **Efficacy score** (ES) is the portion of cases for which the model has $P_{\mathrm{LLM}}(\text{new target}) > P_{\mathrm{LLM}}(\text{old target})$; (b) **Paraphrase score** (PS) is the same as ES but changes the *{question}* with a set of rephrased questions to assess the generalization

• *BiasBios* consists of professional biographies of non-famous people, originally introduced to investigate gender bias in occupations. Each example includes biographical context and a label of target occupation. The first sentence mentions the person's occupation, and subsequent sentences describe the individual's career history but may not be directly related to the prediction, potentially distracting the model attention. At the end of the context, we append the question: *{person} has the occupation of* __.

☞ We emphasize the first sentence, as it carries the most information about the occupation.

Metrics: following [Hernandez et al., 2023], we compute **Accuracy** by checking whether the probability assigned to the target occupation in the highest among the 28 candidate occupations.

For *Pronouns changing*, *CounterFact*, and *BiasBios*, we additionally measure **Fluency** as the average bi-gram and tri-gram entropy of generations, designed to be low for degenerated or repetitive texts [Meng et al., 2022a]. We filter out any results receiving a fluency below 3.0 (see full results including fluency in Appendix D).

Baselines. We compare PASTA to the following baselines:

• *Zero-shot prompting* is the most common approach to interact with LLMs, in which a user feeds models a prompt containing background context and a user instruction or question.

• *Marked prompting* alters the prompts used in zero-shot prompting by surrounding user-specified input spans with emphasis markers, e.g. asterisks, as is done in markdown files for emphasis, or quotes, as is done in natural languages.

• *Few-shot prompting* includes demonstrations (example inputs and target outputs) at the beginning of the prompt fed to the LLM. Few-shot prompting often improves performance in new tasks, but increases the computational cost of inference due to the increased prompt length, particularly when demonstrations are lengthy [Dong et al., 2023]; here we use 3 demonstrations in context.

PASTA settings We study PASTA in 2 settings: *multi-task* and *task-agnostic*. In the multi-task setting, the evaluation task j is included for profiling, whereas in the task-agnostic setting, the evaluation task is excluded (instead, we profile on the 3 datasets besides j). The multi-task setting improves performance but requires labeled training samples for the task which is evaluated, which can be difficult to obtain in practice.

Table 1: Main results of LLAMA-7B to demonstrate that PASTA can improve the model ability to (i) follow user instruction (*JSON Format* and *Prons. Changing*); (ii) interpret contextual information (*BiasBios*); (iii) resolving knowledge conflicts (*CounterFact*). For all scores, higher is better. The best results are in **bold**.

	Method	**JSON Format**	**Prons. Changing**	**BiasBios**	**CounterFact**	**All**
		F. Acc / P. Acc	Acc / A.Acc	Acc	ES / PS	Ave.
Prompting	Zero-shot	60.00 / 54.94	71.84 / 66.28	87.36	58.50 / 52.03	67.29
	∗-marked	18.55 / 12.71	39.14 / 35.17	90.62	57.74 / 50.52	49.38
	“"-marked	4.56 / 4.20	20.55 / 18.19	89.82	58.14 / 51.70	42.15
	Few-shot	84.85 / 73.58	59.06 / 55.27	88.79	87.45 / 49.82	73.45
PASTA	Task-agnostic	88.16 / 49.08	83.65 / 81.31	93.54	98.82 / 99.03	85.89
	Multi-task	**96.64 / 85.09**	**96.42 / 95.84**	**95.28**	**99.60 / 99.57**	**95.46**

Table 2: Main results of GPT-J to demonstrate that PASTA can improve the model ability to (i) follow user instruction (*JSON Format* and *Prons. Changing*); (ii) interpret contextual information (*BiasBios*); (iii) resolving knowledge conflicts (*CounterFact*). For all scores, higher is better. The best results are in **bold**.

	Method	**JSON Format**	**Prons. Changing**	**BiasBios**	**CounterFact**	**All**
		F. Acc / P. Acc	Acc / A.Acc	Acc	ES / PS	Ave.
Prompting	Zero-shot	28.83 / 25.09	39.88 / 36.19	72.76	42.14 / 42.02	44.96
	∗-marked	4.44 / 4.10	41.25 / 37.57	74.14	44.50 / 45.09	40.63
	“”-marked	8.81 / 5.62	6.12 / 5.72	78.64	45.54 / 41.84	33.87
	Few-shot	84.15 / **72.65**	35.77 / 32.08	72.98	68.34 / 38.23	59.65
PASTA	Task-agnostic	46.68 / 34.71	91.62 / 88.60	80.84	**99.54 / 99.57**	77.80
	Multi-task	**91.50** / 18.63	**92.96 / 91.34**	**94.96**	98.62 / 98.79	**85.22**

Empirically, we find that PASTA is not sensitive to the scaling coefficient α (see Section 5.3) and fix it to 0.01 in our experiments. We select 1000 training samples from each of the 4 tasks above for model profiling. After model profiling, we select k from $\{300, 400, 500\}$ for LLAMA-7B to have the number of steered heads $|\mathcal{H}|$ as $\{25, 53, 86\}$. We find that PASTA achieves the best performance on LLAMA-7B when $50 \leq |\mathcal{H}| \leq 100$, i.e., $k = 400$ or $k = 500$. For GPT-J, we select k from $\{250, 275, 300, 350\}$ to have $|\mathcal{H}|$ as $\{52, 72, 111, 153\}$. For every task, we split data into train/validation/test sets following [Hernandez et al., 2023] (See Appendix B) and select $|\mathcal{H}|$ by cross validation when evaluating on test sets. For all tasks, model outputs are generated with greedy search.

5 Results

5.1 Main result: PASTA improves model generation

Tables 1 and 2 present the main results for PASTA applied to LLAMA-7B and GPT-J respectively. Few-shot prompting is the strongest baseline, and task-agnostic PASTA outperforms it on the main metric for each task for all settings except JSON Formatting with GPT-J. Multi-task PASTA outperforms all baselines across all settings.

PASTA can improve LLM instruction following. The results from JSON Formatting and Pronouns Changing tasks indicate that, by highlighting the user instruction at the end of inputs, PASTA effectively steers models to focus on user intentions, thereby biasing their generation to fulfill specific requirements or formats. For example, while GPT-J only achieves 39.9% of its zero-shot generations complying the user requirement on the Pronouns Changing task, PASTA yields a remarkable 53% accuracy improvement by emphasizing the instruction. Moreover, PASTA achieves an impressive 96.64% format accuracy and 85.09% prediction accuracy when applied to LLAMA-7B on the JSON Formatting task. This performance exceeds that of few-shot prompting by 11%, even though few-shot prompting explicitly provides the model with correct JSON examples through additional demonstrations. Table 3 presents a few examples generated by LLAMA-7B when applying PASTA.

PASTA can help models capture crucial contextual information. In the case of BiasBios and CounterFact tasks, we apply PASTA to emphasize specific context spans for LLMs. Consequently,

Table 3: Generation examples of LLAMA-7B on JSON Formatting and Pronouns Changing tasks.

Task	Prompt	Zero-shot Generation	PASTA Generation
JSON Format	Winnie is an American photographer living in New York. Specialized in fashion photography and portrait, she applies her talent on both humans and animals . . . {instruction}	✗ Winnie is a fashion and portrait photographer living in New York, specialized in fashion photography and portrait, who applies her talent on both humans and animals.	✓ {"name": "Winnie", "occupation": "photographer" }
Prons. Change	Mary is an Associate Professor in the Department of Curriculum Instruction at St. John University; she holds a doctorate in Reading/Writing/Literacy from the University of Pennsylvania. She recently published an article in "Urban Education" with . . . {instruction}	✗ Mary is a professor in the Department of Curriculum Instruction at St. John University; **she** holds a doctorate in Reading/Writing/Literacy from the University of Pennsylvania. **She** recently published an article in "Urban Education" with . . .	✓ Mary is an associate professor in the department of curriculum and instruction at St. John's University; **they** hold a doctorate in reading/writing/literacy from the University of Pennsylvania. **They** recently published an article in . . .

the models are guided to pay close attention to the specific contextual information or new facts within contexts. The results from these two tasks illustrate that PASTA can direct the models to interpret the crucial information or resolve the knowledge conflicts within contexts, yielding significant improvement in prediction performance of both tasks. For example, PASTA achieves a prediction accuracy of 94.96% for GPT-J on the BiasBios task, which is 16.32% higher than the best baseline.

Tables 1 and 2 also suggest that marked prompting, a baseline that highlights specific texts akin to human writers, struggles to effectively convey emphasis to LLMs. One possible reason is that these emphasis markers rarely appear in the massive pre-training data. In contrast, few-shot prompting sometimes leads to improvements in model performance. However, a drawback of few-shot prompting is its instability, i.e. its performance exhibits high variance across different samples in the demonstration (See Appendix E).

5.2 PASTA can mitigate the sensitivity of prompts

Table 4: Results about sensitivity of model performance to prompt rephrasing on the JSON Formatting task. Given rephrased instructions in prompt template, PASTA can imporve zero-shot performance for all prompts.

Instruction	**Method**	**LLAMA-7B** JSON Format F. Acc / P. Acc	 Prons. Changing Acc / A. Acc	**GPT-J** JSON Format F. Acc / P. Acc	 Prons. Changing Acc / A. Acc	**Average**
Original	Zero-shot	60.0 / 54.9	71.8 / 66.3	28.8 / 25.1	39.9 / 36.2	47.9
	PASTA	96.6 / 85.1	96.4 / 95.8	91.5 / 18.6	93.0 / 91.3	83.5
Shortened	Zero-shot	36.0 / 32.4	49.2 / 42.6	25.4 / 17.1	56.5 / 54.8	39.3
	PASTA	87.4 / 65.9	89.0 / 86.9	54.1 / 37.0	94.0 / 93.7	76.0
Rephrased	Zero-shot	57.9 / 54.2	82.3 / 79.6	63.3 / 50.3	76.0 / 72.8	67.1
	PASTA	97.1 / 87.1	89.6 / 89.0	77.5 / 68.1	94.8 / 92.3	86.9

It is well-known that the the performance of LLMs can be sensitive to minor changes in prompts, such as rephrasing and reformatting, even when these prompts convey the same meaning [Reynolds and McDonell, 2021, Liu et al., 2021]. We find that PASTA can alleviate the sensitivity of model performance to varying prompts. Specifically, Table 4 evaluates the performance of LLAMA-7B and GPT-J on JSON Formatting and Pronouns Changing task given different instructions in the prompt template, all of which convey the same meaning (see precise prompts in Appendix C.1). The results show that zero-shot performance is sensitive to different prompts and can significantly deteriorate with poorly crafted templates. In contrast, PASTA consistently improves model performance over zero-shot prompting for all prompts, effectively mitigating sensitivity to variations in the prompts.

5.3 Analysis and Ablations

In this section, we investigate different hyperparameter choices and modeling decisions that affect the performance of PASTA.

Model profiling Figure 2 presents the results on the importance of model profiling introduced in Section 3.2. We compare PASTA when steering the selected heads versus other reasonable choices: steering (i) all heads, (ii) entire layers, or (iii) individual heads on the JSON Formatting task (See

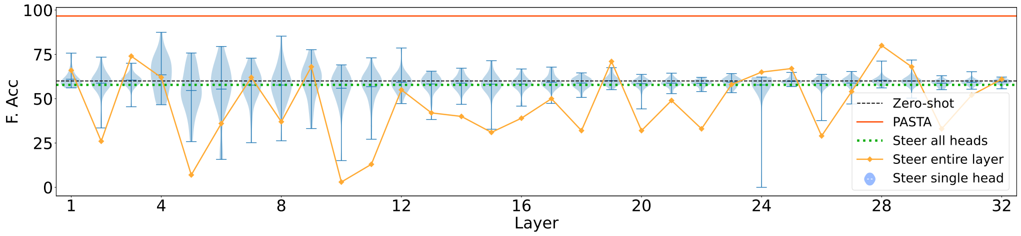


Figure 2: The performance of LLAMA-7B on the JSON Formatting task when we steer (i) all heads (green); (ii) an entire layer (yellow); and (iii) an individual head within a layer (blue violin plot). The performance varies dramatically across layers and across heads of a layer.

Appendix F for comparisons on the remaining tasks). Selecting heads via model profiling in PASTA (red line) significantly outperforms other approaches. Steering all heads (dashed green line) degrades performance compared to the baseline zero-shot performance (dashed black line). This is likely because steering all heads over-amplifies the user-specified information at the expense of other essential information required for effective generation and prediction. Interestingly, we find that the performance varies significantly when steering different layers (yellow) or heads (blue violin plot). As mentioned in Section 1, attention heads play distinct roles in encoding diverse semantic and syntactic information [Tenney et al., 2019]. When steering heads, which are appropriately involved in encoding of user-specified information, the model can be guided to capture and reinforce these specific signals. Conversely, modifying the attention of unrelated heads not only fails to emphasize the desired information but also interferes with their original functions, resulting in performance deterioration. Therefore, it is important to identify the effective heads through model profiling prior to applying the steering.

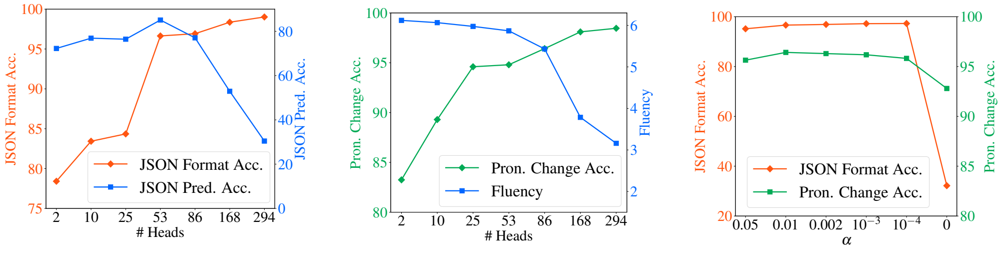
Varying strategies for selecting heads during profiling. As described in Sec. 5.3, our model profiling selects the *Intersection* of the top-k performing heads to steer across multiple tasks. Alternatively, when evaluating on task j, we can select heads for steering with different strategies: (i) *Task-specific* – steer the top-k_2 performing heads of only the task j, i.e., $R^{(j)}_{1:k_2}$; or (ii) *Union* – the union of these heads across multiple tasks, i.e., $\cup_{i=1}^{m} R^{(i)}_{1:k_2}$. Table 5 compares their performance. Using task-specific heads rather than intersection-selected heads sometimes yields improved performance, but requires selecting a different set of heads for each new task.

Table 5: Varying head selection strategies between top *task-specific* heads, *union* across multiple tasks, and intersection (the default used in PASTA).

	PASTA	**JSON Format**	**Prons. Changing**	**BiasBios**	**CounterFact**	**All**
		F. Acc / P. Acc	Acc / A.Acc	Acc	ES / PS	Avg.
LLAMA	Task-specific	95.56 / 86.83	98.52 / 98.02	97.62	99.18 / 99.24	96.57
	Union	88.42 / 74.49	92.12 / 91.44	96.36	99.24 / 99.35	92.22
	Intersection	96.64 / 85.09	96.42 / 95.84	95.28	99.60 / 99.57	95.46
GPT-J	Task-specific	85.71 / 79.39	94.74 / 92.54	97.64	99.26 / 99.34	93.29
	Union	72.61 / 64.89	89.68 / 87.76	95.56	99.82 / 99.83	88.21
	Intersection	91.50 / 18.63	92.96 / 91.34	94.96	98.62 / 98.79	85.22

Varying the number of heads to be steered. Figures 3a and 3b illustrate the performance of PASTA when steering different number of heads on two tasks. The results suggest that as more heads are included for steering, the model follows the user ever more closely, achieving higher efficacy (JSON Format Acc. and Pron. Change Acc.). However, at some point, this it results in a decrease in the metrics reflecting the generation quality (JSON Pred. Acc and Fluency). Thus, there is a trade-off between emphasizing efficacy and generation quality, requiring choosing the number of heads during model profiling.

Varying the scaling coefficient α. Figure 3c presents the performance of PASTA on two tasks when we change the scaling coefficient α. The results indicate that PASTA is fairly robust to this hyperparameter; in practice, we fix it as 0.01. Notice that setting α to zero should be avoided, as this



(a) JSON Format

(b) Prons. Change

(c) Varying α

Figure 3: The performance of applying PASTA to LLAMA-7B on JSON Formating and Pronouns Changing tasks when varying the number of steered heads $|\mathcal{H}|$ (3a,3b); and changing the scaling coefficient α (3c).

leads to the complete removal of other crucial contexts at the steered heads, resulting in performance degeneration.

6 Related work

The primary method for controlling LLMs has been through prompting, often yielding impressive improvements in performance [Brown et al., 2020b, Liu et al., 2021, Wei et al., 2022] and spurring a line of work aiming to make prompting easier, e.g. [Strobelt et al., 2022, Bach et al., 2022, Shin et al., 2020, Deng et al., 2022, Singh et al., 2023b]. However, LLMs remain extremely sensitive to nuances in prompts [Webson and Pavlick, 2021, Lu et al., 2021]; PASTA complements these approaches by making it easier for a user to specify a prompt in difficult scenarios.

Another line of work aims to make LLMs more amenable to prompting by modifying them during training. Most prominent among these approaches are instruction finetuning [Wei et al., 2021, Chung et al., 2022], Reinforcement Learning from Human Feedback [Ziegler et al., 2019, Ouyang et al., 2022], and other related methods, e.g. [Lee et al., 2023]. There are also a few methods for directly specifying which parts on an input are important during training, e.g. [Ross et al., 2017, Rieger et al., 2019, Schramowski et al., 2020, Krishna et al., 2023]. PASTA can be used in addition to these approaches to improve some aspects of model steerability (e.g. instruction following).

PASTA is related to variety of methods for adapting to new tasks, including LoRA [Hu et al., 2021a], AdaLoRA [Zhang et al., 2023], QLoRA [Dettmers et al., 2023], and TOAST [Shi et al., 2023b]. PASTA is also related to a variety of research on model editing, e.g. ROME [Meng et al., 2022a], MEMIT [Meng et al., 2022b], MEND [Mitchell et al., 2022], and REMEDI [Hernandez et al., 2023]. Unlike these works, PASTA preserves an LLMs ability to transfer to new tasks using prompts and human-selected info, rather than using new labeled examples.

Finally, PASTA is also motivated by works which have aimed to mechanistically understand attention scores, e.g. by studying them through feature importance [Jain and Wallace, 2019, Wiegreffe and Pinter, 2019, Deb et al., 2023], through probing [Conneau et al., 2018, Liu and Avci, 2019], through visualization [Karpathy et al., 2015, Olah et al., 2017], localizing knowledge [Meng et al., 2022a, Dai et al., 2021], categorizing directions in representation space [Kim et al., 2017, Schwettmann et al., 2021], or through natural-language explanations [Bills et al., 2023, Singh et al., 2023a].

7 Conclusion

In this study, we propose PASTA, a novel approach aimed at enabling LLMs to move beyond the limitations of plain text and effectively perceive user guidance embodied as highlighted parts of prompts. By making precise adjustments to attention scores in selected heads, PASTA directs the model's focus to the relevant context, mirroring the way humans benefit from textual cues. Unlike traditional fine-tuning methods, PASTA is applied at inference time and requires neither parameter updates nor gradient computation; PASTA requires only selecting which attention heads to apply the re-weighting to, a one-time profiling operation for a LLM. Experimental results show that PASTA can significantly improve model performance on a variety of tasks. In the future, we plan to integrate PASTA with various other methods, such as few-shot in-context learning, aiming to highlight effective examples to enhance its stability.

References

S. H. Bach, V. Sanh, Z.-X. Yong, A. Webson, C. Raffel, N. V. Nayak, A. Sharma, T. Kim, M. S. Bari, T. Fevry, et al. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*, 2022.

S. Bills, N. Cammarata, D. Mossing, H. Tillman, L. Gao, G. Goh, I. Sutskever, J. Leike, J. Wu, and W. Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2023.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020b.

H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://aclanthology.org/W19-4828.

A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018.

D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

M. De-Arteaga, A. Romanov, H. Wallach, J. Chayes, C. Borgs, A. Chouldechova, S. Geyik, K. Kenthapadi, and A. T. Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128, 2019.

M. Deb, B. Deiseroth, S. Weinbach, P. Schramowski, and K. Kersting. Atman: Understanding transformer predictions through memory efficient attention manipulation. *arXiv preprint arXiv:2301.08110*, 2023.

M. Deng, J. Wang, C.-P. Hsieh, Y. Wang, H. Guo, T. Shu, M. Song, E. P. Xing, and Z. Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *arXiv preprint arXiv:2205.12548*, 2022.

T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui. A survey on in-context learning, 2023.

E. Hernandez, B. Z. Li, and J. Andreas. Inspecting and editing knowledge representations in language models, 2023.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021a.

J. E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint abs:2106.09685*, 2021b.

S. Jain and B. C. Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.

S. Krishna, J. Ma, D. Slack, A. Ghandeharioun, S. Singh, and H. Lakkaraju. Post hoc explanations of language models can improve language models. *arXiv preprint arXiv:2305.11426*, 2023.

H. Lee, S. Phatale, H. Mansoor, K. Lu, T. Mesnard, C. Bishop, V. Carbune, and A. Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

F. Liu and B. Avci. Incorporating priors with feature attribution on text classification. *arXiv preprint arXiv:1906.08286*, 2019.

N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts, 2023.

P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022a.

K. Meng, A. S. Sharma, A. Andonian, Y. Belinkov, and D. Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022b.

P. Michel, O. Levy, and G. Neubig. Are sixteen heads really better than one? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c601ad9d2ff9bc8b282670cdd54f69f-Paper.pdf.

E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning. Fast model editing at scale, 2022.

C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

OpenAI. Gpt-4 technical report, 2023.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019.

L. Reynolds and K. McDonell. Prompt programming for large language models: Beyond the few-shot paradigm, 2021.

L. Rieger, C. Singh, W. J. Murdoch, and B. Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. *arXiv preprint arXiv:1909.13584*, 2019.

A. S. Ross, M. C. Hughes, and F. Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

P. Schramowski, W. Stammer, S. Teso, A. Brugger, F. Herbert, X. Shao, H.-G. Luigs, A.-K. Mahlein, and K. Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

S. Schwettmann, E. Hernandez, D. Bau, S. Klein, J. Andreas, and A. Torralba. Toward a visual concept vocabulary for gan latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6804–6812, 2021.

T. Shen, R. Jin, Y. Huang, C. Liu, W. Dong, Z. Guo, X. Wu, Y. Liu, and D. Xiong. Large language model alignment: A survey, 2023.

B. Shi, S. Gai, T. Darrell, and X. Wang. Toast: Transfer learning via attention steering. *arXiv preprint abs:2305.15542*, 2023a.

B. Shi, S. Gai, T. Darrell, and X. Wang. Refocusing is key to transfer learning. *arXiv preprint arXiv:2305.15542*, 2023b.

T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

C. Singh, A. R. Hsu, R. Antonello, S. Jain, A. G. Huth, B. Yu, and J. Gao. Explaining black box text modules in natural language with language models. *arXiv preprint arXiv:2305.09863*, 2023a.

C. Singh, J. X. Morris, J. Aneja, A. M. Rush, and J. Gao. Explaining patterns in data with language models via interpretable autoprompting, 2023b.

N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. J. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano. Learning to summarize from human feedback. *arXiv preprint abs:2009.01325*, 2020.

H. Strobelt, A. Webson, V. Sanh, B. Hoover, J. Beyer, H. Pfister, and A. M. Rush. Interactive and visual prompt engineering for ad-hoc task adaptation with large language models, 2022.

I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, July 2019. URL https://aclanthology.org/P19-1580.

B. Wang and A. Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

A. Webson and E. Pavlick. Do prompt-based models really understand the meaning of their prompts? *arXiv preprint arXiv:2109.01247*, 2021.

J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

S. Wiegreffe and Y. Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Z. Yao, R. Y. Aminabadi, O. Ruwase, S. Rajbhandari, X. Wu, A. A. Awan, J. Rasley, M. Zhang, C. Li, C. Holmes, Z. Zhou, M. Wyatt, M. Smith, L. A. Kurilenko, H. Qin, M. Tanaka, S. Che, S. L. Song, and Y. He. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. *arXiv preprint abs:2308.01320*, 2023.

Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen, and T. Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=lq62uWRJjiY`.

D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix

B Dataset details

In the following table, we provide the detailed statistics of datasets in our experiments.

Table 6: Statistics of datasets.

Task	Train	Valid	Test
CounterFact	1000	1000	5000
BiasBios	1000	1000	5000
JSON Formatting	1000	1000	5000
Pronouns Changing	1000	1000	5000

C Experimental Details

We implement all algorithms using `PyTorch` [Paszke et al., 2019] and `Huggingface` [Wolf et al., 2019] and run experiments on NVIDIA V100 GPUs and NVIDIA A6000 GPUs.

C.1 Detailed prompt templates of each task

For each task, the prompt templates in our results are as follows:

- **JSON Formatting**:
  - (Original) *{context}. Answer the occupation of {person} and generate the answer as json format. Here is an example: {"name": , "occupation": ,}. Now generate the answer.*
  - (Shortened one in Section 5.2) *{context}. Answer the occupation of {person} and generate the answer as json format.*
  - (Rephrased one in Section 5.2) *Answer the occupation of {person} and generate the answer as json format. Here is an example: {"name": , "occupation": ,}. {context}. Now generate the answer.*
- **Pronouns Changing:**
  - (Original): *{context}. For the aforementioned text, substitute 'she' and 'he' with 'they' and generate the occupation of {person} after changing pronouns.*
  - (Shortened one in Section 5.2): *{context}. Change 'she' and 'he' with 'they' and answer the occupation of {person} after replacing the pronouns*
  - (Rephrased one in Section 5.2): *{context}. For the aforementioned descriptions, replace 'she' and 'he' with 'they' in the aformentioned text and generate the new text after replacing the pronouns.*
- **BiasBios:** *{context}. {person} has the occupation of.*
- **CounterFact:** *Previously, {old fact}. Currently, {new fact}. {question}*

C.2 The evaluation details of PASTA

We present the number of heads to be steered by PASTA for LLAMA-7B and GPT-J-6B on every task in the following table.

Table 7: The number of heads to be steered by PASTA.

Task	LLAMA-7B	GPT-J-6B
JSON Formatting	53	153
Pronouns Changing	86	72
BiasBios	86	111
CounterFact	86	52

D Extended results with fluency

In this section, we include extended results, including fluency metrics. Fluency score is the average bi-gram and tri-gram entropy of generations, designed to be low for degenerated or repetitive texts [Meng et al., 2022a]. This metric can be regarded as the reference metric of generation quality. Typically, the generations of language models are reliable as long as their fluency score is not too low. Here, we filter out any results receiving a fluency score below 3.0. Table 8, 9 and 10 include all results and fluency evaluation.

Table 8: Main results of LLAMA-7B to demonstrate that PASTA can improve the model ability to (i) follow user instruction (*JSON Format* and *Prons. Changing*); (ii) interpret contextual information (*BiasBios*); (iii) resolving knowledge conflicts (*CounterFact*). For all scores, higher is better. The best results are in **bold**.

	Method	**JSON Format**	**Prons. Changing**	**BiasBios**	**CounterFact**
		F. Acc / P. Acc	Acc / A.Acc / Flue.	Acc / Flue.	ES / PS /Flue.
Prompting	Zero-shot	60.00 / 54.94	71.84 / 66.28 / **6.10**	87.36 / 3.98	58.50 / 52.03 / 4.96
	∗-marked	18.55 / 12.71	39.14 / 35.17 / 6.03	90.62 / 3.89	57.74 / 50.52 / 5.12
	“"-marked	4.56 / 4.20	20.55 / 18.19 / 5.13	89.82 / 3.97	58.14 / 51.70 / 5.13
	Few-shot	84.85 / 73.58	59.06 / 55.27 / 5.95	88.79 / **4.19**	87.45 / 49.82 / **5.68**
PASTA	Task-agnostic	88.16 / 49.08	83.65 / 81.31 / 4.62	93.54 / 3.03	98.82 / 99.03 / 4.78
	Multi-task	**96.64 / 85.09**	**96.42 / 95.84** / 5.43	**95.28** / 4.05	**99.60 / 99.57** / 4.89

Table 9: Main results of GPT-J to demonstrate that PASTA can improve the model ability to (i) follow user instruction (*JSON Format* and *Prons. Changing*); (ii) interpret contextual information (*BiasBios*); (iii) resolving knowledge conflicts (*CounterFact*). For all scores, higher is better. The best results are in **bold**.

	Method	**JSON Format**	**Prons. Changing**	**BiasBios**	**CounterFact**
		F. Acc / P. Acc	Acc / A.Acc / Flue.	Acc / Flue.	ES / PS /Flue.
Prompting	Zero-shot	28.83 / 25.09	39.88 / 36.19 / 5.91	72.76 / **5.06**	42.14 / 42.02 / 5.01
	∗-marked	4.44 / 4.10	41.25 / 37.57 / 4.76	74.14 / 5.01	44.50 / 45.09 / 5.22
	“”-marked	8.81 / 5.62	6.12 / 5.72 / 5.43	78.64 / 4.96	45.54 / 41.84 / 5.16
	Few-shot	84.15 / **72.65**	35.77 / 32.08 / **6.46**	72.98 / 4.82	68.34 / 38.23 / **5.67**
PASTA	Task-agnostic	46.68 / 34.71	91.62 / 88.60 / 3.00	80.84 / 4.92	**99.54 / 99.57** / 5.11
	Multi-task	**91.50** / 18.63	**92.96 / 91.34** / 4.91	**94.96** / 4.87	98.62 / 98.79 / 5.11

Table 10: Varying head selection strategies between top top *task-specific* heads, *union* across multiple tasks, and intersection (the default used in PASTA).

	PASTA	**JSON Format**	**Prons. Changing**	**BiasBios**	**CounterFact**
		F. Acc / P. Acc	Acc / A.Acc / Flue.	Acc / Flue.	ES / PS /Flue.
LLAMA	Task-specific	95.56 / 86.83	98.52 / 98.02 / 5.92	97.62 / 4.18	99.18 / 99.24 / 4.93
	union	88.42 / 74.49	92.12 / 91.44 / 4.88	96.36 / 4.13	99.24 / 99.35 / 4.53
	intersection	96.64 / 85.09	96.42 / 95.84 / 5.43	95.28 / 4.05	99.60 / 99.57 / 4.89
GPT-J	Task-specific	85.71 / 79.39	94.74 / 92.54 / 5.07	97.64 / 5.06	99.26 / 99.34 / 4.94
	Union	72.61 / 64.89	89.68 / 87.76 / 3.92	95.56 / 5.02	99.82 / 99.83 / 5.03
	Intersection	91.50 / 18.63	92.96 / 91.34 / 4.91	94.96 / 4.87	98.62 / 98.79 / 5.11

E Extended results

E.1 The variance of few-shot performance

Few-shot prompting sometimes leads to improvements in model performance. as explicitly providing the examples in additional demonstrations. However, a drawback of few-shot prompting is its instability, meaning its performance exhibits high variance across different samples in the demonstratio.

In this section, we present the results to show that the performance of few-shot prompting displays high variance in terms of sampling different few-shot demonstrations.

Table 11: The few-shot performance (Acc. / A. Acc. / Fluency) on the Pronouns Changing task.

Few-shot examples	LLAMA-7B	GPT-J-6B
Demonstration 1	84.87 / 90.09 / 4.74	43.82 / 40.36 / 6.43
Demonstration 2	57.24 / 53.98 / 6.22	40.68 / 37.86 / 6.44
Demonstration 3	57.08 / 53.22 / 6.02	33.13 / 29.21 / 6.48
Demonstration 4	52.26 / 48.30 / 6.42	25.47 / 20.89 / 6.44
Demonstration 5	43.86 / 40.78 / 6.43	11.90 / 8.63 / 6.51

F Model Profiling Results

F.1 Steering performance on more datasets

In this Section, we provide more results of the performance of LLAMA-7B on all of tasks when steering: (i) all heads; (ii) entire layer; (iii) a individual head of a layer.

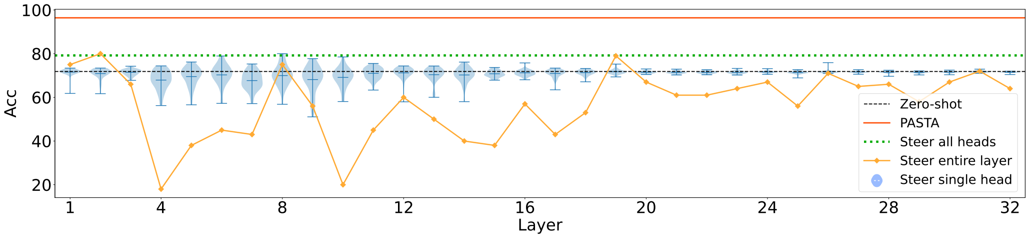


Figure 4: The performance of LLAMA-7B on Pronouns Changing task when we steer (i) all heads (green); (ii) entrie layer (yellow); and (iii) individual head with a layer (blue violin plot). The performance varies dramatically across layers and across heads of a layer.

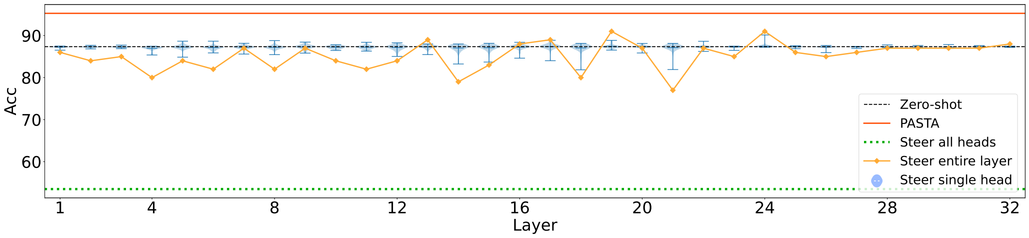


Figure 5: The performance of LLAMA-7B on BiasBios task when we steer (i) all heads (green); (ii) entrie layer (yellow); and (iii) individual head with a layer (blue violin plot). The performance varies dramatically across layers and across heads of a layer.

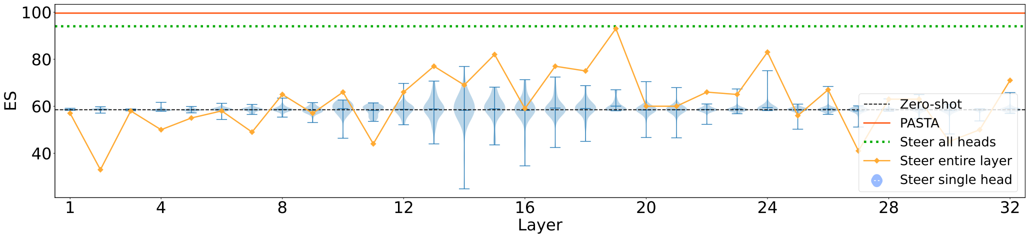


Figure 6: The performance of LLAMA-7B on CounterFact task when we steer (i) all heads (green); (ii) entrie layer (yellow); and (iii) individual head with a layer (blue violin plot). The performance varies dramatically across layers and across heads of a layer.