Correcting Negative Bias in Large Language Models through Negative Attention Score Alignment

Sangwon Yu[1*] **Jongyoon Song**[1*] **Bongkyu Hwang**[2] **Hoyoung Kang**[2] **Sooah Cho**[2]
Junhwa Choi[2] **Seongho Joe**[2] **Taehee Lee**[2] **Youngjune Gwon**[2] **Sungroh Yoon**[1,3†]
[1]Department of Electrical and Computer Engineering, Seoul National University
[2]Samsung SDS, Korea
[3]AIIS, ASRI, INMC, ISRC, and IPAI, Seoul National University
{dbtkddnjs96, coms1580, sryoon}@snu.ac.kr
{bongkyu.hwang, hoyoung.kang, sooah.cho, jh496.choi, drizzle.cho, taehee23.lee, gyj.gwon}@samsung.com

Abstract

A binary decision task, like yes-no questions or answer verification, reflects a significant real-world scenario such as where users look for confirmation about the correctness of their decisions on specific issues. In this work, we observe that language models exhibit a negative bias in the binary decisions of complex reasoning tasks. Based on our observations and the rationale about attention-based model dynamics, we propose a negative attention score (NAS) to systematically and quantitatively formulate negative bias. Based on NAS, we identify attention heads that attend to negative tokens provided in the instructions as answer candidate of binary decisions, regardless of the question in the prompt, and validate their association with the negative bias. Additionally, we propose the negative attention score alignment (NASA) method, which is a parameter-efficient fine-tuning technique to address the extracted negatively biased attention heads. Experimental results from various domains of reasoning tasks and large model search space demonstrate that NASA significantly reduces the gap between precision and recall caused by negative bias while preserving their generalization abilities.

1 Introduction

Recent advancements in large language models (LLMs) have enabled complex reasoning tasks executed by understanding user instructions (Ouyang et al., 2022; Touvron et al., 2023; Achiam et al., 2023; Jiang et al., 2023). As the capabilities of LLMs have expanded rapidly, research has intensified to analyze their characteristics and inherent issues. One of the major issues is the generation of factually incorrect content, known as "hallucination", which significantly degrades the reliability of LLM-based services (Zhang et al., 2023; Xu et al., 2024; Huang et al., 2023).

The hallucination problem in LLMs can be attributed to factors such as parametric knowledge, overconfidence, and biases (Zhang et al., 2023). Studies have been conducted to understand the decision-making mechanisms in LLMs from a parameter perspective. For instance, the "logit lens" technique allows for the interpretation of the model's reasoning process along the layers, exploiting the hidden representations from intermediate layers (Belrose et al., 2023; Ferrando et al., 2023). This technique has been utilized to analyze model characteristics in various scenarios such as in-context learning. On the other hand, Yuan et al. (2024) focus on analyzing hallucination phenomenon based on attention heads. However, the causes and mechanisms of hallucination in LLMs vary by type of task, indicating that extensive further study is still necessary.

In this paper, we aim to identify biases that emerge when LLMs respond to questions requiring a binary decision, such as affirmation or negation. We observe a negative bias in LLMs in yes-no question-answering (QA) and answer verification tasks that demand complex reasoning, such as mathematical or logical reasoning. Specifically, we observe that LLMs generally exhibit high precision but low recall in binary decision tasks. Additionally, we find that the prediction confidence for negative decisions is significantly higher than for positive decisions. This suggests that *LLMs tend to be overly cautious when making positive decisions, leading to a phenomenon where the model excessively favors negative decisions as a form of shortcut*. This negative bias, which indicates a discrepancy between the model's reasoning ability and accuracy, may ultimately reduce the reliability of its predictions.

From the rationale that the model allocates higher attention to the negative candidates among the binary answer options presented in the user prompt, we propose a **n**egative **a**ttention **s**core

*Equal Contribution
† Corresponding author

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 9979–10001

(**NAS**) to systematically probe these negatively biased attention heads in LLMs. Based on NAS, we detect attention heads that attend to tokens associated with negation in an instruction. Our findings also confirm the existence of attention heads that predominantly contribute to manifesting this negative bias, regardless of the given query. To address the identified attention heads, we introduce a parameter-efficient fine-tuning technique named **NAS a**lignment (**NASA**).

In the NASA framework, we first construct a probing set in the form of a binary decision task to extract negative attention heads. This probing set is derived from an existing short-answer QA dataset. Next, we apply an incremental fine-tuning process to the negative attention heads identified based on NAS. During this process, each head undergoes fine-tuning using the probing set, while NASA includes periodic monitoring of NAS, which automatically schedules early stopping and update cancellations. This pipeline ensures that tuning is conducted only up to an appropriate extent and in the correct sequence, effectively mitigating negative bias without compromising the model's existing capabilities.

For the evaluation of NASA, we measure the performance of four LLMs: LLaMA3-8B-Instruct (Touvron et al., 2023), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Gemma-1.1-7b-it (Team et al., 2024), and Qwen2-7B-Instruct (Yang et al., 2024a) as the search space. We conduct a comprehensive evaluation across multiple reasoning tasks, including multi-hop QA: StrategyQA (Geva et al., 2021), MuSiQue (Trivedi et al., 2022), mathematical reasoning: GSM8k-Rephrased and MATH-Rephrased subset in MetaMATH (Yu et al., 2023), and logical reasoning: AR-LSAT (Zhong et al., 2021).

Experimental results demonstrate that our method NASA significantly reduces the gap between precision and recall while maintaining or even improving accuracy and F1 score. This indicates that the NASA method successfully mitigates negative bias without compromising overall performance. Additionally, we confirm that the NASA-tuned model effectively preserves general reasoning performance beyond binary decision-making and exhibits improvements in calibration. Furthermore, NASA demonstrates robust performance across various prompting formats, including few-shot settings.

Our contributions can be summarized as follows:

- We observe that LLMs exhibit a negative bias in binary decision tasks requiring complex reasoning and demonstrate its association with attention heads.
- We propose negative attention score (NAS), a metric and framework that allows for systematic probing of attention heads involved in negative bias regardless of the input query.
- Based on NAS, we introduce NAS alignment (NASA), a parameter-efficient fine-tuning technique that effectively addresses negative bias by tuning the extracted negative attention heads in a proper order.

2 Related Work

Hallucination problems in complex reasoning tasks The emergent abilities of LLMs, such as the chain-of-thoughts, have significantly enhanced the performance of complex reasoning (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2022; Madaan et al., 2023). However, the increased complexity of tasks and pipelines has made the hallucination problem in complex reasoning tasks challenging to analyze and resolve, which remains an active area of research (Zhang et al., 2023). With the advent of benchmarks vulnerable to hallucinations, recent LLMs are laying the groundwork for analyzing the causes of LLM hallucinations based on substantial data which requires complex reasoning abilities (Geva et al., 2021; Trivedi et al., 2022;Cobbe et al., 2021).

Analysis on roles and knowledge in model parameters As the capacity and complexity of LLMs increase, the need to understand and interpret these models is becoming more crucial. In particular, there is active research into analyzing parametric knowledge and model behavior that influence the phenomenon of model hallucination. Meng et al. (2022) propose a framework to specify the location where the knowledge of the model is stored, while Belrose et al. (2023) and Yang et al. (2024b) research to interpret the inference process of the model. Yuan et al. (2024) suggest a method to detect model parameters involved in the false premise hallucination phenomenon of LLMs.

Recent studies have shown that attention heads in language models play a variety of roles such as knowledge recalling (Zheng et al., 2024b). For example, Jin et al. (2024) find the existence of attention heads in LLMs that either recall parametric

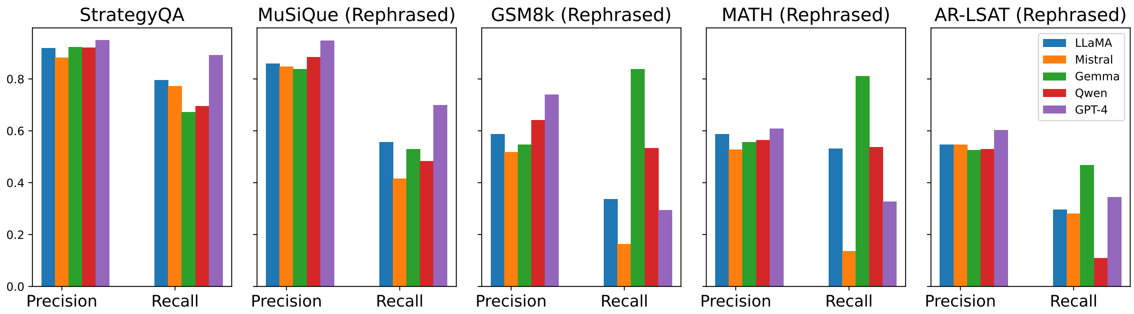

Figure 1: Precision and recall of LLMs in complex reasoning tasks.

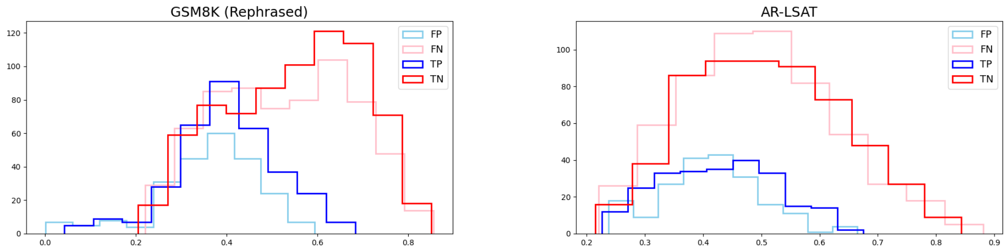

Figure 2: Confidence histograms of LLaMA3-8B-Instruct model for the rephrased GSM8K and AR-LSAT samples.

knowledge or retrieve information from external contexts and García-Carrasco et al. (2024) propose a method to identify attention heads that increase vulnerability to adversarial attacks.

In this paper, we identify the negative bias in LLMs that leads to hallucinations in binary decision tasks requiring complex reasoning. We focus on the attention heads in LLMs and propose a framework to detect and address the attention heads responsible for the negative bias.

3 Negative Bias in LLMs

A binary decision task, such as yes-no QA or answer verification, represents a major real-world scenario where users seek confirmation on whether their decisions regarding specific problems are correct. For instance, a user may inquire whether a statement formed through various contexts is true. Also, a user may seek validation of the correctness of their proposed solutions to problems requiring complex logical reasoning. Such fundamental interactions, represented as yes-no decisions, act as the basic block of the high-dimensional reasoning tasks that need step-by-step sub-tasks. Despite this significance of the binary decision task, many complex reasoning benchmark datasets are not structured in a binary decision format. In this section, we first transform existing complex reasoning tasks into yes-no binary decision tasks. We then conduct statistical observations and quantitative analyses to examine how LLMs perform in these binary decision scenarios, observing a meaningful negative bias in their decisions. Following this, we propose a systematic framework, the **N**egative **A**ttention **S**core (**NAS**), to formulate and quantify the negative bias exhibited by LLMs.

3.1 Statistical Observation

3.1.1 Setup for Binary Decision Task

Transformation to binary decision task We transform general QA reasoning tasks into binary decision tasks by modifying each sample into a *positive* or *negative* sample where the answer is “Yes” or “No”, respectively. In this section, we focus on the case of short-answer QA datasets. We refer to Appendix D for the multiple choice datasets.

For the positive sample, we perform a rule-based transformation on general queries by appending a confirmatory query. For example, the query "What is 1+1?" is transformed into "What is 1+1? Is the answer 2?". This approach is straightforward for the cases where the correct answer is “Yes”. Formally, given the question and its label, we use Prompt I to transform the prompt into the positive sample:

Prompt I (Positive Sample Transformation)
You are given a question and you MUST answer Yes or No. Question: $\{question\}$ Is the answer $\{label\}$? Answer:

To transform into the negative sample, a wrong label is necessary based on the query and answer which will be placed in *label* in Prompt I. For this, we obtain suitable incorrect answers by prompting GPT-4 (OpenAI, 2023). We present GPT-4 prompts for the wrong label generation in Appendix D.

Datasets and models We utilize StrategyQA, MuSiQue, subsets of MetaMATH (GSM8k-Rephrased and MATH-Rephrased), and AR-LSAT for evaluation, covering three domain tasks: multi-hop open-domain QA, mathematical reasoning, and logical reasoning. StrategyQA is a yes-no QA dataset, MuSiQue and the rephrased versions of GSM8k and MATH are short-answer QA datasets, and AR-LSAT is a multiple-choice dataset. Samples are in Table 14. To assess reasoning capabilities in binary decision tasks, we convert these datasets into binary decision sets using the techniques above, except for StrategyQA. AR-LSAT is converted to the yes-no QA formats, while the others are rephrased for answer verification formats. Details on the conversion can be found in Appendix D. We sample a portion of the entire dataset, and detailed statistics can be found in Appendix A. For the search space of LLMs, we explore LLaMA3-8B-Instruct (LLaMA), Mistral-7B-Instruct-v0.3 (Mistral), Gemma-1.1-7b-it (Gemma), Qwen2-7B-Instruct (Qwen), and GPT-4.

3.1.2 Observation of Negative Bias

Fig 1 presents the precision and recall results on the transformed datasets. We use FP, FN, TP, and TN to denote false positive, false negative, true positive, and true negative, respectively. In most cases, a significant drop in recall compared to precision is observed. According to the definition, a high precision implies that a significant proportion of the samples predicted as positive by the model (TP+FP) are true positives. Low recall means that among all the positive samples (TP+FN), the model rarely responds correctly with positive answers (TP). The phenomenon where existing models exhibit high precision but low recall can be explained to that *the model is overly cautious in outputting positive responses*. Conversely, this implies that the model outputs negative responses indiscriminately, indicating that the trustworthiness of negative responses and positive responses is not balanced.

Fig 2 shows a histogram of the response confidence of the LLaMA in mathematical and logical reasoning tasks. We observe that the overall frequency and confidence of negative responses (FN and TN) tend to be higher than those of positive responses (FP and TP). This indicates that *the model tends to output negative responses more frequently and with greater confidence*.

In summary, we find that LLMs typically show 1) over-cautiousness to the positive response, and 2) more frequent and confident negative responses. It is evident that *large language models possess a bias towards negative responses in the binary decision-making process of complex reasoning tasks*. A case study of the negative responses about the negative bias can be found in Appendix G.

3.2 Formulation of Negative Bias

3.2.1 NAS: Negative Attention Score

In this section, we formulate negative bias from the perspective of model intrinsic properties. Specifically, we focus on the model's internal attention patterns. To obtain an answer for a binary decision, a user generally provides answer candidates like "Yes" or "No" as instructions to the model before posing the query prompt (c.f., Prompt I). The model follows these instructions and responds to the given query in the manner provided in the instructions. During this response process, the model attends to the given instructions, and due to the operational characteristics of attention-based models, the candidate with the larger attention weight generally appears in the response. In this context, the negative bias of the LLM can be seen as originating from assigning greater attention weight to negative answer candidates during the reasoning process. Fig 3 illustrates an example of negative attention head. To validate this rationale, we define Negative Attention Score (NAS) using the attention weights applied to negative answer candidate tokens like "No", and positive answer candidate tokens like "Yes", observed in a attention head.

Let L_x be the length of the sample x, t_{Yes} and t_{No} the positions of the "Yes" and "No" tokens within the instruction, and L_I the length of the instruction in x. For the attention weight inferred by the h-th attention head in the l-th layer, denoted as $A^{l,h} \in \mathbb{R}^{L_x \times L_x}$, the NAS is defined as:

$$\text{NAS}_x^{l,h} := \sum_{i=L_I}^{L_x} \left(A_{i,t_{Yes}}^{l,h} + A_{i,t_{No}}^{l,h} \right) * \log \left(\frac{A_{i,t_{No}}^{l,h}}{A_{i,t_{Yes}}^{l,h}} \right), \quad (1)$$

where we omit the input x in the attention function for brevity. The NAS term consists of two

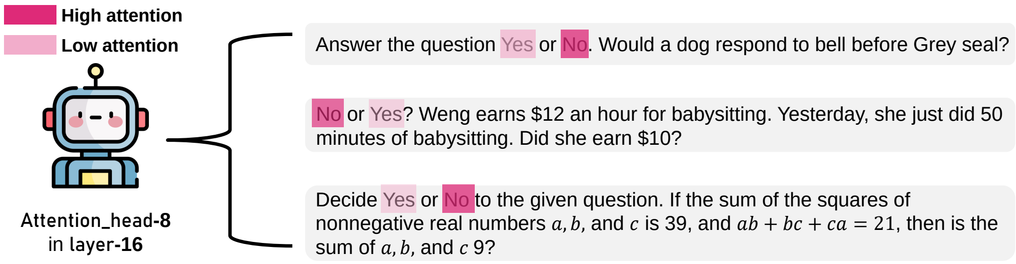

Figure 3: An example of the negative attention head. Three queries in the figure are sampled from StrategyQA (Geva et al., 2021), rephrased GSM8K and MATH datasets (Yu et al., 2023).

Table 1: Correlations of NAS and the negative confidence (-1 to 1).

Model	LLaMA	Mistral	Gemma	Qwen
Pearson	0.5274	0.3625	0.6322	0.4806
Spearman	0.5201	0.3563	0.6725	0.5166

Table 2: Overlapping proportion of top-100 negative heads for various datasets (0 to 1).

LLaMA	Mistral	Gemma	Qwen
0.74	0.76	0.80	0.74

factors: the sum of the attention weights applied to the "Yes" and "No" tokens, which aims to find attention heads focusing significantly on the tokens representing answer options; and the logarithm of the ratio of attention weights applied to these tokens, identifying heads that preferentially attend to the "No" token over the "Yes" token.

NAS can have a value for each attention head on a single sample. Based on this, we introduce two types of NAS variants: **single head NAS** and **model NAS**. Single head NAS represents the average NAS of a specific attention head across the sample(s). It is used to quantify the negative bias of a single attention head. On the other hand, model NAS represents the total NAS across all attention heads for the sample(s). This is used to approximate the model's negative bias. Formally, given a sample set X, we define the single head NAS of the h-th attention head in the l-th layer and model NAS as follows:

$$\mathrm{NAS}(X, l, h) := \frac{1}{|X|} \sum_{x_i \in X} \mathrm{NAS}_{x_i}^{l,h} \ \textbf{(single head)}$$

$$\mathrm{NAS}(X, L, H) := \sum_{l \in L} \sum_{h \in H} \mathrm{NAS}(X, l, h) \ \textbf{(model)},$$

where L and H are the sets of all layer and attention head indices, respectively.

3.2.2 Empirical Studies

NAS and Negative Confidence To demonstrate that the previously defined NAS is an effective indicator of negative bias, we measure the correlation between model NAS and the negative confidence observed in model responses. For this measurement, we construct a test set consisting of a total of 1,500 samples, drawing 500 samples each from StrategyQA, rephrased GSM8K, and rephrased AR-LSAT. Table 1 shows the Pearson correlation and Spearman's rank correlation coefficient measured between model NAS and negative response confidence for each model. In all cases, we confirm that model NAS has a positive correlation with negative confidence. This validates that NAS is an effective indicator of a model's negative bias.

Overlapping Negative Attention Heads We conduct a study on negative attention heads defined through NAS. Using our previously constructed test set, which spans three different domains, we investigate whether the negative attention heads extracted from each domain subset overlap with one another. For each sample x_i in a domain subset X, we first define the tuple list P^i consisting of the layer and head indices of the top-k attention heads with the highest single head NAS:

$$P_X^i = \text{Top-}k_{(l,h)} \, \mathrm{NAS}(\{x_i\}, l, h). \tag{2}$$

Among these extracted heads, we select those that are consistently included in over 90% of the tuple lists, which we denote as C_X:

$$C_X = \{(l, h) \mid (l, h) \text{ appears in at least } 0.9 \times |X| \text{ of } P_X^i\}. \tag{3}$$

Finally, we extract the list of top N attention heads P_X, sorted by their single head NAS values on X.

$$P_X = \text{Top-}N_{(l,h) \in C} \, \mathrm{NAS}(X, l, h). \tag{4}$$

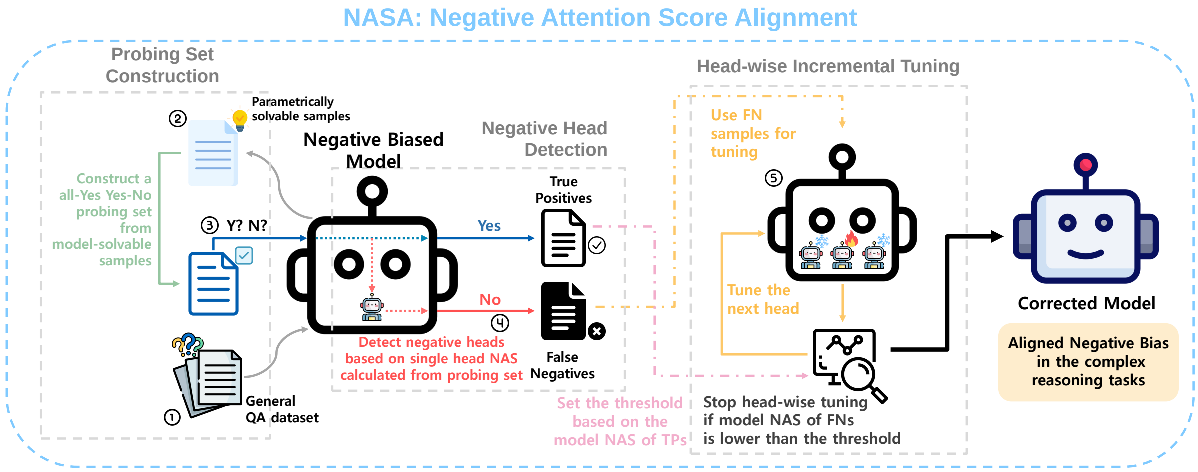

Figure 4: Overall framework of negative attention score alignment method.

In this process, we set k to 200 and N to 100.

Keeping the model type constant, we extract such negative attention heads from each domain subset and then measure the extent of their overlap. The overlapping rates are displayed in Table 2, and surprisingly, we find that negative heads extracted through different domain subsets significantly overlap. This indicates the existence of *query-agnostic negative attention heads* that represent a common underlying cause of the model's negative bias.

4 NASA: NAS Alignment Framework

Motivated by previous observations, we propose **NAS A**lignment (**NASA**), a framework designed to address attention heads that induce negative bias in a parameter-efficient manner. In NASA, we start by constructing a probing set to identify negative attention heads, followed by a selection process to determine which of these heads will be fine-tuned. Our framework fine-tunes the target attention heads in the order based on single head NAS.

4.1 Probing Set Construction

To construct the probing set, we use a short-answer QA dataset and select the parametric samples where the model can correctly answer the question. A detailed explanation of the parametric sample selection can be found in Appendix E. Next, we convert the selected parametric samples into binary decision making format using Prompt I in Section 3.1.1. Note that in the probing set, the label is uniformly set to "Yes". The probing set is designed to have two properties: i) the model possesses knowledge about the question, and ii) the model should respond "Yes" to the converted question. We assume that attention heads with a high single head NAS for the converted samples that meet the two conditions are strongly associated with negative bias and should be addressed.

In Section 3.2.2, we observe that the negative bias attention heads exhibit query-agnostic properties. Based on this observation, we employ another multi-step reasoning dataset, HotpotQA (Yang et al., 2018), to construct the probing set. We presume that HotpotQA, being less challenging compared to recent datasets, has a higher proportion of samples that can be answered using the parametric knowledge of LLMs.

4.2 Negative Attention Head Probing

Using the negative attention head detection process described in Section 3.2.2, we extract the attention heads to be fine-tuned by utilizing the probing set X. In this process, we set k to 100 and N to 30.

At this point, we further categorize the probing set X into cases where the model correctly answers $TP(X)$ (i.e., the true positive set) and cases where it does not $FN(X)$ (i.e., the false negative set). During the fine-tuning phase of the model (See Section 4.3), we use the false negative set as the fine-tuning set D. This approach is aimed at inducing the model to generate positive answers for samples that it fails to answer due to a negative bias. Meanwhile, the true positive set is used to determine the threshold τ for early halting of the fine-tuning process.

$$\tau := \min_{x \in TP(X)} \text{NAS}(\{x\}, L, H) \tag{5}$$

In other words, during the training process aimed at reducing the model NAS, if the model NAS on the validation set (i.e., subset of $FN(X)$) falls below the minimum model NAS of true positive set, training is halted to minimize the bias towards positives.

4.3 Head-wise Incremental Tuning

Training setup As mentioned in Section 4.2, we utilize the FNs in the probing set for fine-tuning. Because the model fails to generate positive answers for the samples although the model possesses parametric knowledge, it is expected that fine-tuning with these samples can effectively address the negative bias. We do not provide contexts containing related facts to the model because the model already has the corresponding knowledge.

Similar to a standard supervised fine-tuning objective, we target only the answer token (e.g., "Yes") for training. Since the fine-tuning set contains only short answers, training the model to generate the end of a sequence token could introduce a bias towards short responses. To prevent this side effect, we exclude the end of a sequence token in a loss calculation.

Training strategy We fine-tune the query and key projection modules of each attention head sequentially in order of decreasing single head NAS which is measured during the probing stage.

Since we fine-tune with data where the target answer is positive, there is a risk of inducing a positive bias. Furthermore, updating a single attention head might change the NAS of another attention head. To prevent these, we set aside a portion of the dataset as a validation set and apply *early stopping* and *update cancellation* schemes for each attention head tuning and *early halting* scheme for the whole fine-tuning process.

For early stopping, we set criteria based on the model NAS and single head NAS. For the former criteria, we check for a decrease in the value after each epoch. For the latter one, we verify that i) the value decreases and ii) remains above a specific threshold ρ after each epoch. Fine-tuning of the current attention head is stopped if either of the three conditions is not met.

For the update cancellation, after fine-tuning each attention head, we calculate the difference between the single head NAS and the model NAS before and after fine-tuning. If any of these values have increased compared to before training, we revert the parameter update of that attention head to exclude attention heads that reinforce the negative bias from the training target.

For the early halting, we utilize the model NAS of each sample within the $TP(X)$, τ, as mentioned in Section 4.2. This threshold setting is based on the rationale that if the model NAS in a $FN(X)$ is lower than in a $TP(X)$, it may induce a positive bias. After training each negative attention head, we measure the model NAS on the validation set. If this value falls below τ, training is halted to prevent falling into local optima. Algorithm for the entire process can be found in Appendix F.

5 Experiments

5.1 Experimental Setup

We use LLaMA, Mistral, Gemma, and Qwen for our experiments. We use those supervised fine-tuned models before NASA-tuning as baselines. We use LLaMA-Factory (Zheng et al., 2024a)[1] and HuggingFace Transformers (Wolf et al., 2020)[2] for all the experiments. The models are evaluated using binary decision datasets, which are described in Section 3, measuring accuracy, precision, recall, F1, and the model NAS. We set ρ to 0.5, the initial learning rate to 1e$-$6, the batch size to 32, and the maximum number of epochs to 30. We reference the Alpaca repository[3], setting the weight decay to 0 and the warmup ratio to 0.03.

5.2 Main Results

Tables 3 and 4 show the performance of our method across various binary decision tasks. Generally, we observe an improvement in accuracy and a significant reduction in the precision-recall gap. As noted in Section 3, high precision coupled with low recall indicates that the model is excessively cautious in providing positive responses, resulting in a degradation in the trustworthiness of negative responses. From this perspective, the negative bias can be mitigated by reducing the gap between precision and recall while maintaining accuracy, which balances the trustworthiness of the model's responses and ultimately improves the model.

Additionally, based on the improvement in the F1 score, which is the harmonic mean of precision and recall, we demonstrate that the balance between precision and recall has been achieved in the direction of improving the model's capability for reasoning. We also observe that our method effectively reduces the model NAS. In Appendix K,

[1] https://github.com/hiyouga/LLaMA-Factory
[2] https://github.com/huggingface/transformers
[3] https://github.com/tatsu-lab/stanford_alpaca

Table 3: Model evaluation results on multi-step reasoning datasets.

Dataset	Model	Accuracy	Precision	Recall	F1	NAS
StrategyQA	LLaMA / + NASA	0.871 / **0.877**	**0.919** / 0.864	0.795 / **0.875**	0.852 / **0.869**	160.6 / **55.0**
	Mistral / + NASA	**0.842** / 0.830	**0.882** / 0.802	0.772 / **0.861**	0.823 / **0.830**	114.1 / **34.3**
	Gemma / + NASA	**0.778** / 0.771	**0.923** / 0.907	0.676 / **0.690**	0.777 / **0.784**	107.8 / **71.8**
	Qwen / + NASA	0.829 / **0.844**	**0.921** / 0.907	0.694 / **0.742**	0.792 / **0.816**	209.6 / **107.2**
MuSiQue (Rephrased)	LLaMA / + NASA	0.720 / **0.757**	**0.859** / 0.815	0.555 / **0.683**	0.675 / **0.743**	199.8 / **64.0**
	Mistral / + NASA	0.649 / **0.707**	**0.848** / 0.802	0.416 / **0.594**	0.558 / **0.682**	131.2 / **110.7**
	Gemma / + NASA	**0.518** / 0.503	**0.837** / 0.834	0.529 / **0.555**	0.648 / **0.667**	259.8 / **96.2**
	Qwen / + NASA	0.621 / **0.639**	**0.883** / 0.858	0.482 / **0.552**	0.624 / **0.672**	418.8 / **282.9**

Table 4: Model evaluation results on mathematical and logical reasoning datasets.

Dataset	Model	Accuracy	Precision	Recall	F1	NAS
GSM8K (Rephrased)	LLaMA / + NASA	0.537 / **0.545**	**0.587** / 0.531	0.336 / **0.838**	0.428 / **0.650**	218.2 / **74.7**
	Mistral / + NASA	0.501 / **0.512**	**0.518** / 0.509	0.163 / **0.782**	0.247 / **0.617**	115.5 / **77.0**
	Gemma / + NASA	**0.533** / 0.524	**0.546** / 0.541	0.837 / **0.887**	0.661 / **0.672**	159.6 / **66.2**
	Qwen / + NASA	0.538 / **0.576**	**0.640** / 0.614	0.534 / **0.738**	0.582 / **0.670**	210.9 / **82.9**
MATH (Rephrased)	LLaMA / + NASA	**0.563** / 0.549	**0.587** / 0.534	0.530 / **0.874**	0.557 / **0.663**	183.3 / **60.8**
	Mistral / + NASA	0.504 / **0.526**	**0.527** / 0.522	0.136 / **0.685**	0.216 / **0.592**	117.9 / **109.6**
	Gemma / + NASA	**0.546** / 0.543	**0.556** / 0.553	0.810 / **0.871**	0.659 / **0.677**	185.0 / **64.0**
	Qwen / + NASA	0.564 / **0.581**	**0.661** / 0.617	0.537 / **0.715**	0.593 / **0.662**	224.0 / **103.1**
AR-LSAT (Rephrased)	LLaMA / + NASA	0.513 / **0.518**	**0.547** / 0.537	0.297 / **0.444**	0.385 / **0.486**	178.1 / **63.6**
	Mistral / + NASA	0.511 / **0.530**	0.547 / **0.553**	0.280 / **0.447**	0.370 / **0.494**	108.7 / **80.02**
	Gemma / + NASA	0.509 / **0.510**	**0.525** / 0.523	0.468 / **0.534**	0.495 / **0.528**	151.7 / **68.0**
	Qwen / + NASA	0.492 / **0.496**	**0.529** / 0.528	0.110 / **0.182**	0.182 / **0.270**	317.6 / **207.3**

we demonstrate that the components of our method contribute effectively to tackling negative bias.

Meanwhile, the slight decrease in precision might superficially appear as a performance decline. However, it can be interpreted as enhancing the model's trustworthiness by adjusting the alignment between the model's actual knowledge and its responses. This means that precision might slightly decrease because some samples incorrectly categorized as true negatives due to negative bias might shift to false positives. Specifically, even if the model makes incorrect inferences or remains uncertain about a question, in a binary decision scenario, negative bias could lead to it being classified as a true negative. An analysis of the shifts in negative responses due to NASA can be found in Appendix H.

6 Analysis

We analyze the impact of NASA on the general reasoning capability of LLMs beyond the binary decision task which is the primary focus of this work. After that, we examine the effect of NASA on model calibration. Additionally, regarding performance improvement and negative bias, we investigate the relationship between few-shot examples and NASA in few-shot prompting scenarios. We present an instruction following analysis and an ablation study in Appendices I, J and K.

6.1 General Reasoning Abilities

We evaluate the performance of LLMs fine-tuned with NASA on general QA reasoning tasks. We utilize general reasoning benchmarks such as MuSiQue, GSM8K, and AR-LSAT without the transformation. For short-answer QA, we first verify whether each model-generated prediction includes an answer in these benchmarks and then utilize GPT-4 to double-check whether the prediction is semantically correct. To generate predictions from the model, we utilize Prompt II in Section 4.1. The content of the GPT-4 prompt for prediction verification can be found in Table 15.

As shown in Table 5, models fine-tuned with NASA show enhanced or maintained general reasoning abilities while effectively tackling negative bias. Since negative bias is a task-specific factor contributing to hallucination in binary decision tasks, its influence on general reasoning QA, like short-answer QA, is relatively low compared to other factors contributing to hallucination such as parametric knowledge. Therefore, improving reasoning ability in binary decision tasks does not necessarily enhance reasoning capabilities in gen-

Table 5: General QA reasoning ability of baseline (left of "/") and ours (right of "/").

Model	MuSiQue	GSM8K	AR-LSAT
LLaMA	0.594 / 0.596	0.329 / 0.325	0.235 / 0.227
Mistral	0.415 / 0.408	0.126 / 0.133	0.174 / 0.174
Gemma	0.139 / 0.137	0.257 / 0.251	0.125 / 0.131
Qwen	0.541 / 0.541	0.478 / 0.481	0.220 / 0.221

Table 6: Expected calibration error of baseline and ours.

Dataset	Mistral	+ NASA
StrategyQA	0.120	**0.116**
MuSiQue (Rephrased)	0.278	**0.212**
GSM8K (Rephrased)	0.301	**0.269**
MATH (Rephrased)	0.353	**0.260**
AR-LSAT (Rephrased)	0.399	**0.355**

eral reasoning QA. This means that addressing bias in binary decision tasks while maintaining performance in general reasoning QA can be considered an advancement in the model's overall reasoning capability, not overfitted to the binary decision task.

6.2 Enhanced Model Calibration

As Fig 2 illustrates, the negative bias of LLMs is also associated with the model's confidence in its predictions. This issue regarding the gap between prediction confidence and actual accuracy is related to research on calibration, which aims to enhance the trustworthiness of LLMs (Kadavath et al., 2022). Table 6 shows that the expected calibration error (He et al., 2022) is reduced when NASA is applied to Mistral. This indicates that NASA effectively mitigates model bias, better aligning prediction confidence and accuracy.

6.3 Analysis in a Few-shot Scenario

We compare the F1 scores of the baselines and our models in a 4-shot setting on StrategyQA, rephrased GSM8K / AR-LSAT. As shown in Table 7, we first observe that the F1 score gap between the baselines and our models decreases compared to the zero-shot scenario, as reported in Tables 3 and 4. Given that few-shot prompting has been reported to improve model calibration (Kadavath et al., 2022), these results suggest that the inclusion of few-shot examples enhances model calibration, thereby partially reducing negative bias. Nevertheless, our model consistently outperforms the baselines in terms of F1 score across all cases. This indicates that NASA remains effective in mitigating negative bias in few-shot scenarios.

Table 7: Precision, recall, and F1 scores in a 4-shot scenario (baseline / NASA).

Model	Dataset	Precision	Recall	F1
LLaMA	StrategyQA	**0.899** / 0.872	0.812 / **0.851**	0.853 / **0.862**
	GSM8K	**0.543** / 0.532	0.720 / **0.870**	0.619 / **0.661**
	AR-LSAT	**0.562** / 0.557	0.431 / **0.590**	0.488 / **0.573**
Mistral	StrategyQA	**0.877** / 0.832	0.824 / **0.889**	0.850 / **0.859**
	GSM8K	**0.548** / 0.522	0.763 / **0.885**	0.638 / **0.656**
	AR-LSAT	**0.552** / 0.532	0.589 / **0.720**	0.570 / **0.612**
Gemma	StrategyQA	**0.903** / 0.897	0.654 / **0.659**	0.758 / **0.760**
	GSM8K	0.588 / **0.590**	0.876 / **0.894**	0.704 / **0.711**
	AR-LSAT	**0.550** / 0.535	0.366 / **0.405**	0.439 / **0.461**
Qwen	StrategyQA	**0.901** / 0.894	0.770 / **0.805**	0.830 / **0.847**
	GSM8K	**0.596** / 0.569	0.765 / **0.827**	0.670 / **0.674**
	AR-LSAT	**0.535** / 0.546	0.194 / **0.303**	0.285 / **0.390**

7 Discussion

Through our experiments, we identify two key characteristics of negative bias.

Model NAS may not negatively correlate with F1 score during fine-tuning As shown in Table 19, we observe instances where freezing certain parameters in the attention module yields higher F1 scores despite an increase in NAS. These results suggest that a phase during fine-tuning exists where the negative correlation between the F1 score and NAS is disrupted. To mitigate performance degradation, we introduce thresholds such as early stopping and halting within our framework.

Accurate identification of negative attention heads is crucial, and NAS facilitates this process Our ablation study in Table 20 indicates that fine-tuning randomly selected attention heads leads to suboptimal results. In some cases, we observe abnormal increases or decreases in model NAS, implying the importance of fine-tuning problematic attention heads rather than unrelated parameters. Leveraging NAS and the probing set, we successfully detect negative attention heads, leading to consistent improvements in the F1 score.

8 Conclusion

We identify a critical issue where large language models exhibit negative bias in binary decision tasks requiring complex reasoning. To address this issue, we propose a negative attention score and employ it to discover query-agnostic negative heads. By performing parameter-efficient tuning on these heads, we introduce the NASA method, which effectively mitigates the bias problem. Our method not only enhances the performance of the model but also serves as a useful analytical framework from the perspective of interpretability.

Limitations

In this work, we focus on understanding the relationship between negative biases exhibited in fine-tuned LLMs and attention heads. However, further research is still needed to comprehend the causes and mechanisms behind the occurrence of negative biases in LLMs, and we anticipate that our observations and experimental results will lay the groundwork for future work. Additionally, we have adopted a scheme of fine-tuning a small number of attention heads individually. In future work, it may be possible to explore more time-efficient training methods. While this study focuses on understanding the characteristics of negative bias attention heads, future work could involve a more integrated research approach connecting various elements.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A3B1077720, No. 2022R1A5A708390811), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)], the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University, Samsung SDS Co., Ltd, and a grant from the Yang Young Foundation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Jorge García-Carrasco, Alejandro Maté, and Juan Trujillo. 2024. Detecting and understanding vulnerabilities in language models via mechanistic interpretability. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 385–393. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Guande He, Jianfei Chen, and Jun Zhu. 2022. Preserving pre-trained features helps calibrate fine-tuned language models. In *The Eleventh International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1193–1215, Bangkok, Thailand. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *CoRR*.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. 2024b. Do large language models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Longhui Yu, Weisen Jiang, Han Shi, YU Jincheng, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*.

Hongbang Yuan, Pengfei Cao, Zhuoran Jin, Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao. 2024. Whispers that shake foundations: Analyzing and mitigating false premise hallucinations in large language models. *arXiv preprint arXiv:2402.19103*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024a. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

Zifan Zheng, Yezhaohui Wang, Yuxin Huang, Shichao Song, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024b. Attention heads of large language models: A survey. *arXiv preprint arXiv:2409.03752*.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2021. Ar-lsat: Investigating analytical reasoning of text. *Preprint*, arXiv:2104.06598.

Table 8: Data statistics of datasets utilized in our work.

Data	# of Positives	# of Negatives	# of total
StrategyQA	1071	1219	2290
MuSiQue (Rephrased)	1208	1208	2416
GSM8K (Rephrased)	1001	999	2000
MATH (Rephrased)	1000	1000	2000
AR-LSAT (Rephrased)	820	777	1597

Table 9: Results of multiple regression alongside a correlation for multiple comparisons. r^2 denotes the effect size.

Model	Coefficient	Corrected p-value	r^2
LLaMA	-3.04	$7.18E-01$	0.271
Mistral	19.57	$1.22E-03$	0.060
Gemma	29.61	$1.23E-21$	0.377
Qwen	126.03	$4.65E-69$	0.265

A Data Statistics

Table 8 shows the statistics of dataset utilized in our work.

B A Statistical Analysis of the Relationship between Negative Confidence and NAS

To analyze the impact of negative confidence on model NAS, we conduct a multiple regression analysis by including positive confidence as a variable, as shown in Table 9. In all cases except for LLaMA, where the corrected p-value is high, negative confidence is observed to have a statistically significant positive correlation with model NAS. Meanwhile, considering the effect sizes, future work could investigate additional factors influencing model NAS.

C Instructions for NASA

Table 12 presents the instructions utilized for NASA process in our work and Table 11.

D Details of Binary Decision Data Construction

To convert general QA datasets into binary decision tasks, we create positive and negative examples where the answers are “Yes” and “No”, respectively. Note that StrategyQA is a yes-no QA dataset and does not require a transformation.

MuSiQue, GSM8K, and MATH are short-answer QA datasets, and AR-LSAT is a multiple-choice dataset. For short-answer QA datasets, we use Prompt I in Section 3.1.1 to generate positive examples. AR-LSAT, a multiple-choice dataset, utilizes the positive question generation prompt from Table 13 to create questions with the answer "Yes." Consequently, StrategyQA and AR-LSAT (Rephrased) are yes-no QA datasets, while MuSiQue (Rephrased), GSM8K (Rephrased), and MATH (Rephrased) is the form of answer verification.

D.1 GPT-4 Prompts for Negative Example Construction

To construct negative examples, we utilize GPT-4. In short-answer QA (i.e., MuSiQue, GSM8K, and MATH), we use the wrong label generation prompt from Table 13 to obtain incorrect answers. We then replace the label in Prompt I with the incorrect answer to generate a negative example.

In multiple choice (i.e., AR-LSAT), we use the negative question generation prompt from Table 13 to create questions with the answer “No”. An example of a rephrased AR-LSAT example can be found in Table 14.

E Details of Parametric Sample Selection

As mentioned in Section 4.1, we construct the probing and fine-tuning sets using the HotpotQA dataset, which is a short-answer QA dataset. From the HotpotQA dataset, we use samples that can be correctly answered solely with parametric knowledge, without any context, as our probing samples. Specifically, we suppose that the probing samples must meet the condition: *the question should be answerable using the model’s parametric knowledge.* To determine whether the question in the sample meets the condition, we construct the prompt as follows:

Prompt II (Inquiring Parametric Knowledge)
You MUST answer shortly the given question based on your knowledge. Question: $\{question\}$ Answer:

This prompt is designed to identify attention heads that attend to the negative token in the instruction even when a positive answer is expected. We assume that if attention heads attend to the negative token when the model knows the correct answer and the target answer is “Yes”, it is generated by a negative bias in the model. This prompt effectively selects samples that belong to the model’s parametric knowledge while minimizing the influence of the model’s negative bias. We assume that samples meet the condition if the model returns the correct answers for the prompt, and we use them as the probing set. We utilize GPT-4 to verify whether the prediction matches the label.

During the probing sample selection process, we first input the question into the target model to extract a prediction. Since the HotpotQA dataset can contain various forms of answers, we use GPT-4 to determine if the prediction corresponds to the label. Specifically, we input the prompt from Table 15 into GPT-4 and only include samples in the probing set if the response is "Yes".

F Algorithm of Incremental Head Tuning

Algorithm 1 shows the details of head-wise incremental tuning of NASA.

Algorithm 1 Head-wise Incremental Tuning

Input: Fine-tuning set D, validation set V, model ψ, single head NAS threshold ρ, list of negative attention heads to be fine-tuned P, early halting threshold τ, maximum training epoch T, set of all layer and head indices L and H, respectively.
Output: NASA-tuned model.

for each $(l, h) \in P$ **do**
  $\psi_{init} = \psi$
  $\alpha_{init} = \text{NAS}(V, l, h)$
  $\beta_{init} = \text{NAS}(V, L, H)$
  $\alpha = \beta = \infty$
  for epoch $= 1$ to T **do**
    Update ψ by fine-tuning h-th attention head in the l-th layer using D.
    $\alpha' = \text{NAS}(V, l, h)$
    $\beta' = \text{NAS}(V, L, H)$
    if $\alpha' > \alpha$ **or** $\alpha' < \rho$ **or** $\beta' > \beta$ **then**
      break ▷ Early stopping
    $\alpha \leftarrow \alpha', \beta \leftarrow \beta'$
  if $\alpha' > \alpha_{init}$ or $\beta' > \beta_{init}$ **then**
    $\psi \leftarrow \psi_{init}$ ▷ Update cancellation
  if $\beta' < \tau$ **then**
    break ▷ Early halting
return ψ

G Case Study for Negative Responses

In this section, we classify negative responses based on the following criteria:

Firstly, in a general QA reasoning task, a model's response to a specific question can be broadly categorized into below two cases.

- Deterministic Case (Det): The model provides a certain answer regardless of its correctness.
- Non-Deterministic Case (Non-Det): The model fails to provide a certain answer, indicating that the question is unanswerable due to insufficient information from the given context or its knowledge base.

Also, the deterministic case can be further classified based on whether the certain answer provided by the model is correct (True) or incorrect (False). Considering this, a model's response to a general question requiring a short answer can be categorized into the following three types.

- True Deterministic (Det-T): The model provides a correct answer.
- False Deterministic (Det-F): The model provides an incorrect answer.
- Non-Deterministic (Non-Det): The model indicates that the question is unanswerable.

Additionally, beyond the extrinsic responses of the model, we can classify the model's responses based on the intrinsic factor of confidence. Specifically, if the model exhibits low confidence in the response, it can be considered closer to not knowing the answer to the question. We calculate the entropy of the output distribution for the first token of the model's response and use the median value of these entropies across all responses to classify them as high-confidence (low-entropy) or low-confidence (high-entropy). This classification is performed based on the original model's responses before applying NASA method.

We apply this classification to cases where the model gives a negative response in a binary decision task. Negative responses can be classified as true negative (TN) or false negative (FN), with our focus in this discussion on FN. In this, an FN response classified as high-confident Det-T is the case where the model knows the answer is correct but says "No", while an FN response classified as Non-Det is the case where the model doesn't know the correct answer and says "No". For high-confident Det-F responses, the model does not indicate unanswerability but rather holds incorrect knowledge. This issue falls outside the scope of our methodology, which focuses on handling negative bias in binary decision tasks rather than general knowledge-intensive tasks.

Although it is clear that the case where the model knows the answer is correct but says "No" is the most problematic, we argue that in a binary decision task, the case where the model doesn't know the correct answer and says "No" can be related to the issue of negative bias. In a binary decision task where the model must answer "Yes" or "No", instances where the model is uncertain about the correct answer should result in an equal frequency of positive (yes) and negative (no) responses. However, our analysis indicates that existing models tend to output negative responses more confidently and frequently (as shown in Figures 1 and 2 in the paper). This suggests that when the model is uncertain, it predominantly outputs negative responses, which negatively impacts the trustworthiness of these negative responses. Therefore, for models exhibiting such negative bias, Non-Det and low-confident responses should also be partially adjusted.

Table 16 shows the compositions of FN responses from the original models according to the classifications mentioned. It is evident that across all types of reasoning tasks examined in our work, including multi-hop, mathematical, and logical reasoning, FN responses encompass two distinct cases: (A) instances where the model recognizes the correct answer but incorrectly responds with "No", and (B) instances where the model lacks knowledge of the correct answer and responds with "No".

Meanwhile, for Gemma, we observe that the model tends to generate overly cautious responses when performing multi-hop QA. Specifically, instead of selecting the "unanswerable" option for multiple samples, Gemma rephrases its responses in an alternative long-form manner. As a result, instances that should have originally been classified as Non-Det are instead categorized as Det-F. This should be taken into account when interpreting the results in Table 16.

H Analysis of Negative Responses Shifting in NASA

In this section, we analyze the ratios of samples which are originally responded to as negative by the original model and subsequently changed to positive by the NASA model. We categorize the ratio cases by each classification group explained in section G.

Table 17 illustrates the ratios of samples shifted from FN to TP. This demonstrates that the Det-T corresponding to the case where the model knows the answer is correct but says "No" generally has the highest rate of change. This finding indicates that our methodology does not indiscriminately add positive bias to the model but rather enhances the model's binary decision reasoning ability effectively.

Table 10: Accuracy and F1 (left and right of "/", respectively) scores on various positive and negative tokens.

Type	Dataset	LLaMA	+ NASA
True/False	StrategyQA	0.855 / 0.828	**0.866** / **0.853**
	MATH	0.559 / 0.452	**0.574** / **0.583**
	AR-LSAT	**0.512** / 0.306	0.506 / **0.405**
Correct/Wrong	StrategyQA	**0.852** / 0.838	0.846 / **0.840**
	MATH	**0.545** / 0.660	0.544 / **0.675**
	AR-LSAT	**0.523** / 0.540	0.522 / **0.590**

Table 18 illustrates the ratios of samples shifted from TN to FP. As discussed in the section on main results, some samples that are incorrectly categorized as TN due to negative bias might shift to FP (Det-F), and this case does not correspond to the model degradation. Actually, among the TN samples, those classified as Det-T with high confidence exhibit a lower shift ratio compared to their Det-F counterparts. This indicates that the model tends to maintain predictions for samples that it can accurately classify. In conclusion, it can be interpreted that NASA has improved reasoning capability in binary decision tasks while maintaining general reasoning capability.

I Generalization to Universal Binary Decision

We analyze the generalization ability for binary decisions by replacing the positive and negative vocabulary used in our fine-tuning set. Our fine-tuning set consists of "Yes" and "No" in the instruction, and we test with the tokens "True" and "False", as well as "Correct" and "Wrong". Table 10 presents the results of experiments with LLaMA on three datasets transformed for the binary decision task. We observe that models fine-tuned with NASA achieve semantic-level generalization without overfitting to the specific vocabulary.

J Transferability across Various Instructions

We fix the instruction for binary decision-making throughout the application of our method. In this section, we evaluate the robustness of our method when presented with instructions in the inference

Table 11: Accuracy and F1 (left and right of "/", respectively) scores on various instruction types.

It. Type	Dataset	LLaMA	+ NASA
A	MuSiQue	0.686/0.641	**0.710/0.700**
	GSM8K	0.534/0.472	**0.543/0.575**
	AR-LSAT	**0.508**/0.306	0.507/**0.370**
B	MuSiQue	0.708/0.692	**0.736/0.756**
	GSM8K	**0.536**/0.611	0.531/**0.665**
	AR-LSAT	0.512/0.382	**0.518/0.500**

stage that are different forms of content. We assess the binary decision performance using the LLaMA with and without NASA on benchmarks: MuSiQue, GSM8K, and AR-LSAT using paraphrased instruction types A and B. Note that the datasets are transformed for the binary decision task. Details of instruction types can be found in Appendix C. As shown in Table 11, models fine-tuned with NASA still demonstrate superior performance compared to the baseline in both prompt types. This supports our claim of instruction robustness for our method. Fig 11 in Appendix C shows the content of instructions.

K Ablation Study

NASA updates the parameters related to the inference of attention weights of negative attention heads (i.e., query and key projection weights). To demonstrate the effectiveness of NASA, we conduct two ablation studies: i) updating only the query projection weight and ii) random attention head tuning.

K.1 Freezing Key Projection Weight

We perform fine-tuning using the same pipeline as before, except that we freeze the key projection weight. Note that strategies such as early stopping and halting, as well as other hyperparameters, remain the same as those used in the NASA settings.

The experimental results of NASA and freezing key projection weights are shown in Table 19. In general, both the F1 score and NAS are relatively better for NASA, while the trend in accuracy is not distinctly observable. An exceptional case is Qwen at AR-LSAT, where NASA's accuracy and F1 score are significantly lower than those of its counterpart.

K.2 Random Attention Head Tuning

In another ablation study, we experiment by randomly selecting attention heads to be fine-tuned. As shown in Table 20, NAS typically outperforms most others in terms of F1 score and expected calibration error. Notably, during random attention head tuning, model NAS largely diverges in the case of LLaMA and Mistral. These results imply that carefully selecting attention heads is critical and our framework effectively detects and addresses negatively biased attention heads.

L Confidence Histogram Analysis

To analyze the confidence of binary decisions between the baseline and our model, we plot confidence histograms for the LLaMA model, similar to Fig 2. In Fig 5, we observe that NASA influences the prediction confidence of LLMs in binary decision tasks. In the baseline model, except for StrategyQA, we frequently observe the return of negative responses with high confidence. On the other hand, after incorporating NASA, the model demonstrates a reduction in the frequency and confidence gap between positive and negative responses.

Meanwhile, for datasets of mathematical reasoning, there has been a significant increase in the frequency of positive responses, even though F1 scores also increased. As discussed in Section 5.2, future research could explore additional regularization during the fine-tuning process.

M Samples of Attention Heads

Fig 6 presents the real examples of attention weights of heads in the LLaMA model for a GSM8K rephrased sample. Fig 7 presents the real examples of attention weights of heads in the LLaMA model with NASA for the same GSM8K rephrased sample.

N Computation

For all fine-tuning experiments, we use 2 NVIDIA A40 GPUs for approximately 3 hours.

O Licenses

StrategyQA, MuSiQue, MetaMATH, and AR-LSAT are under the license of MIT license, CC-BY-4.0 license, MIT License, and MIT license, respectively. LLaMA3-8B-Instruct, Mistral-7B-Instruct-v0.3, Gemma-1.1-7b-it, and Qwen2-7B-Instruct, and GPT-4 are under the license of META LLAMA 3 COMMUNITY LICENSE AGREEMENT, Apache License 2.0, Gemma, Apache License 2.0, and OpenAI, respectively.

P Usage of AI Writing Assistance

This paper received linguistic assistance from the AI assistant GPT-4, which provided services including paraphrasing, spell-checking, and refinement of the original content authored. No additional help was utilized beyond this support.

Table 12: Instructions used NASA process in our work and Table 11.

Instruction in NASA `You are given a question and you MUST answer Yes or No.`
Instruction type A `You are asked a question that demands a clear Yes or No answer.`
Instruction type B `A question is posed to you, and you are obligated to answer either Yes or No.`

Table 13: GPT-4 prompts used in wrong label or question generation for binary decision sample construction.

Wrong Label Generation (MuSiQue, GSM8K, and MATH)

```
[System]
Generate the short wrong answer word for the given question. You MUST refer the given context
about the question.
[User]
Question: {question}
Context: {context}
Answer: {label}
Wrong answer:
```

Positive / Negative Question Generation (AR-LSAT)

```
[System]
Convert the given question to the binary Yes-No question based on the context about the given
question and the answer of the given question. The answer of the converted question must be YES.
Do NOT omit the condition in the given question like ‘If ...’ or ‘Suppose ...’. You MUST include
the entire contents of the given question to the converted question.
[User]
Context: {context}
Question: {question}
Answer: {label / wrong label}
Converted Question:
```

Table 14: Examples of original datasets and rephrased AR-LSAT. Examples are sampled from StrategyQA (Geva et al., 2021), MuSiQue (Trivedi et al., 2022), GSM8K-Rephrased and MATH-Rephrased (Yu et al., 2023), and AR-LSAT (Zhong et al., 2021).

StrategyQA
Context: Depression is caused by low levels of serotonin, dopamine and norepinephrine. Monoamine Oxidase breaks down neurotransmitters and lowers levels of serotonin, dopamine and norepinephrine. **Question**: Would a Monoamine Oxidase candy bar cheer up a depressed friend? **Answer**: No
MuSiQue (Original)
Context: [Title: President of the Confederate States of America] The president was indirectly elected by the people through the Electoral College to a six - year term and was one of only two nationally elected Confederate officers ... **Question**: When did the president of the Confederate States of America end his fight in the Mexican-American war? **Answer**: 1848
GSM8K (Original)
Question: What is the total cost of purchasing equipment for all sixteen players on the football team, considering that each player requires a \$25 jersey, a \$15.20 pair of shorts, and a pair of socks priced at \$6.80? **Answer**: 1500
MATH (Original)
Question: What is the sum of all positive integer values of n for which $\frac{n+6}{n}$ is an integer? **Answer**: 23
AR-LSAT (Original)
Context: Hannah spends 14 days, exclusive of travel time, in a total of six cities. Each city she visits is in one of three countries—X, Y, or Z. Each of the three countries has many cities. Hannah visits at least one city in each of the three countries. She spends at least two days in each city she visits. She spends only whole days in any city. If the city of Nomo is in country X, and if Hannah spends as many days as possible in Nomo and as few days as possible in each of the other cities that she visits **Question**: Which one of the following must be true? (A) Hannah cannot visit any other cities in country X. (B) Hannah can visit four cities in country Y. (C) Hannah can spend six days in Nomo. (D) Hannah cannot spend more than four days in country Z. (E) Hannah can visit, at most, a total of four cities in countries Y and Z. **Answer**: (B) Hannah can visit four cities in country Y.
AR-LSAT (Rephrased)
Question: Hannah spends 14 days, exclusive of travel time, in a total of six cities. Each city she visits is in one of three countries—X, Y, or Z. Each of the three countries has many cities. Hannah visits at least one city in each of the three countries. She spends at least two days in each city she visits. She spends only whole days in any city. If the city of Nomo is in country X, and if Hannah spends as many days as possible in Nomo and as few days as possible in each of the other cities that she visits can Hannah visit four cities in country Y? **Answer**: Yes

Table 15: GPT-4 prompt used in prediction verification for the general reasoning task.

```
[System]
Can the prediction be considered as the same meaning as the answer to the question? You must
answer only yes or no.
[User]
Question: {context}
Prediction: {model prediction}
Answer: {label}
```

Table 16: Composition of false negative samples of original models in the rephrased datasets. High denotes high-confident and Low denotes low-confident.

Dataset	Model	Det-T		Det-F		Non-Det	
		High	Low	High	Low	High	Low
MuSiQue	LLaMA	0.21	0.182	0.141	0.261	0.04	0.166
	Mistral	0.159	0.103	0.218	0.27	0.095	0.155
	Gemma	0.009	0.022	0.546	0.42	0	0.003
	Qwen	0.25	0.174	0.139	0.327	0.007	0.104
GSM8K	LLaMA	0.209	0.092	0.286	0.414	0	0
	Mistral	0.076	0.03	0.407	0.486	0.001	0
	Gemma	0.097	0.065	0.378	0.459	0	0
	Qwen	0.194	0.121	0.278	0.407	0	0
AR-LSAT	LLaMA	0.057	0.06	0.303	0.312	0.158	0.11
	Mistral	0.058	0.078	0.385	0.437	0.002	0.04
	Gemma	0.011	0.025	0.49	0.474	0	0
	Qwen	0.068	0.108	0.306	0.413	0.092	0.013

Table 17: Response shift ratio by NASA on false negative samples (FN $\rightarrow$ TP ratio) in the rephrased datasets. High denotes high-confident and Low denotes low-confident.

Dataset	Model	Det-T		Det-F		Non-Det	
		High	Low	High	Low	High	Low
MuSiQue	LLaMA	0.364	0.39	0.268	0.326	0.136	0.197
	Mistral	0.432	0.449	0.265	0.28	0.203	0.163
	Gemma	0.185	0	0.029	0	0	0
	Qwen	0.122	0.15	0.074	0.093	0	0.032
GSM8K	LLaMA	0.662	0.694	0.629	0.715	N/A	N/A
	Mistral	0.817	0.667	0.746	0.638	1	N/A
	Gemma	0.14	N/A	0.254	N/A	N/A	N/A
	Qwen	0.315	0.586	0.389	0.386	N/A	N/A
AR-LSAT	LLaMA	0.5	0.432	0.298	0.33	0.228	0.281
	Mistral	0.585	0.189	0.289	0.195	0	0
	Gemma	0.143	0.158	0.146	0.131	N/A	N/A
	Qwen	0.22	0.098	0.079	0.095	0.062	0.111

Table 18: Response shift ratio by NASA on true negative samples (TN → FP ratio) in the rephrased datasets. High denotes high-confident and Low denotes low-confident.

Dataset	Model	Det-T		Det-F		Non-Det	
		High	Low	High	Low	High	Low
MuSiQue	LLaMA	0.057	0.06	0.144	0.112	0	0.06
	Mistral	0.073	0.075	0.097	0.08	0.119	0.032
	Gemma	0	N/A	0.007	N/A	0	N/A
	Qwen	0.009	0.018	0.015	0.026	0	N/A
GSM8K	LLaMA	0.587	0.632	0.516	0.606	N/A	N/A
	Mistral	0.784	0.65	0.736	0.652	N/A	N/A
	Gemma	0.271	N/A	0.188	N/A	N/A	N/A
	Qwen	0.213	0.208	0.234	0.303	N/A	N/A
AR-LSAT	LLaMA	0.229	0.349	0.306	0.247	0.176	0.21
	Mistral	0.208	0.204	0.248	0.162	0	0.174
	Gemma	0.133	0.083	0.152	0.134	N/A	N/A
	Qwen	0.109	0.041	0.13	0.095	0	0

Table 19: Ablation results of updated parameters (NASA / query projection only).

Dataset	Model	Accuracy	Precision	Recall	F1	NAS	ECE
StrategyQA	LLaMA	0.877 / **0.877**	0.864 / **0.895**	**0.875** / 0.836	**0.869** / 0.864	**55.0** / 96.2	**0.102** / 0.128
	Mistral	0.830 / **0.849**	0.802 / **0.846**	**0.861** / 0.837	0.830 / **0.841**	**34.3** / 71.9	0.116 / **0.114**
	Gemma	0.771 / **0.774**	0.907 / **0.907**	0.690 / **0.692**	0.784 / **0.785**	**71.8** / 79.2	0.162 / **0.161**
	Qwen	**0.844** / 0.836	0.907 / **0.915**	**0.742** / 0.716	**0.816** / 0.804	**107.2** / 194.9	**0.106** / 0.115
MuSiQue (Rephrased)	LLaMA	**0.757** / 0.741	0.815 / **0.844**	**0.683** / 0.609	**0.743** / 0.707	**64.0** / 118.6	**0.039** / 0.055
	Mistral	**0.707** / 0.704	0.802 / **0.821**	**0.594** / 0.558	**0.682** / 0.664	**110.7** / 185.9	**0.212** / 0.217
	Gemma	0.503 / **0.510**	0.834 / **0.836**	**0.555** / 0.552	**0.667** / 0.665	**96.2** / 107.1	0.276 / **0.274**
	Qwen	**0.639** / 0.622	0.858 / **0.873**	**0.552** / 0.500	**0.672** / 0.636	**282.9** / 405.7	**0.200** / 0.215
GSM8K (Rephrased)	LLaMA	0.545 / **0.558**	0.531 / **0.551**	**0.838** / 0.688	**0.650** / 0.612	**74.7** / 123.6	0.170 / **0.105**
	Mistral	**0.512** / 0.511	0.509 / **0.515**	**0.782** / 0.463	**0.617** / 0.487	**77.0** / 127.4	**0.269** / 0.294
	Gemma	**0.524** / 0.517	0.541 / **0.542**	0.887 / **0.888**	0.672 / **0.673**	**66.2** / 74.7	**0.346** / 0.349
	Qwen	**0.576** / 0.560	0.614 / **0.634**	**0.738** / 0.606	**0.670** / 0.620	**82.9** / 141.6	0.115 / **0.085**
MATH (Rephrased)	LLaMA	0.549 / **0.572**	0.534 / **0.562**	**0.874** / 0.740	**0.663** / 0.639	**60.8** / 103.6	0.175 / **0.118**
	Mistral	0.526 / **0.533**	0.522 / **0.549**	**0.685** / 0.384	**0.592** / 0.452	**109.6** / 152.8	**0.260** / 0.299
	Gemma	**0.543** / 0.539	0.553 / **0.555**	**0.871** / 0.867	**0.677** / 0.677	**64.0** / 73.0	0.340 / **0.338**
	Qwen	**0.581** / 0.549	0.617 / **0.639**	**0.715** / 0.611	**0.662** / 0.624	**103.1** / 187.5	**0.162** / 0.168
AR-LSAT (Rephrased)	LLaMA	**0.518** / 0.510	**0.537** / 0.536	**0.444** / 0.337	**0.486** / 0.414	**63.6** / 109.2	**0.227** / 0.245
	Mistral	**0.530** / 0.526	0.553 / **0.556**	**0.447** / 0.382	**0.494** / 0.453	**80.02** / 121.3	**0.355** / 0.382
	Gemma	0.510 / **0.514**	0.523 / **0.526**	0.534 / **0.551**	0.528 / **0.539**	**68.0** / 75.6	**0.372** / 0.383
	Qwen	0.496 / **0.559**	0.528 / **0.643**	0.182 / **0.574**	0.270 / **0.607**	207.3 / **204.4**	**0.359** / 0.384

Table 20: Ablation results of attention head selection (NASA / random attention heads).

Model	Dataset	Accuracy	Precision	Recall	F1	NAS	ECE
LLaMA	StrategyQA	**0.877** / 0.870	0.864 / **0.924**	**0.875** / 0.789	**0.869** / 0.851	55.0 / $-\mathbf{507.6}$	**0.102** / 0.176
	MuSiQue	**0.757** / 0.722	0.815 / **0.862**	**0.683** / 0.547	**0.743** / 0.669	64.0 / $-\mathbf{956.7}$	**0.039** / 0.076
	GSM8K	0.545 / **0.554**	0.531 / **0.570**	**0.838** / 0.482	**0.650** / 0.522	74.7 / $-\mathbf{433.8}$	0.170 / **0.109**
	MATH	0.549 / **0.562**	0.534 / **0.573**	**0.874** / 0.561	**0.663** / 0.567	60.8 / $-\mathbf{390.1}$	0.175 / **0.095**
	AR-LSAT	0.518 / 0.518	0.537 / **0.570**	**0.444** / 0.249	**0.486** / 0.346	63.6 / $-\mathbf{817.5}$	**0.227** / 0.259
Mistral	StrategyQA	0.830 / **0.849**	0.802 / **0.888**	**0.861** / 0.784	0.830 / **0.833**	**34.3** / 750.8	0.116 / **0.112**
	MuSiQue	**0.707** / 0.679	0.802 / **0.850**	**0.594** / 0.483	**0.682** / 0.616	**110.7** / 1405.1	**0.212** / 0.244
	GSM8K	0.512 / **0.518**	0.509 / **0.539**	**0.782** / 0.287	**0.617** / 0.374	**77.0** / 1055.0	**0.269** / 0.339
	MATH	0.525 / **0.528**	0.522 / **0.563**	**0.685** / 0.266	**0.592** / 0.361	**109.6** / 1042.2	**0.260** / 0.354
	AR-LSAT	**0.530** / 0.514	**0.553** / 0.551	**0.447** / 0.290	**0.494** / 0.380	**80.0** / 1015.0	**0.355** / 0.396
Gemma	StrategyQA	0.771 / **0.780**	0.907 / **0.922**	**0.690** / 0.678	**0.784** / 0.782	**71.8** / 104.7	**0.162** / 0.165
	MuSiQue	0.503 / **0.532**	**0.834** / 0.829	**0.555** / 0.534	**0.667** / 0.650	**96.2** / 123.8	**0.276** / 0.279
	GSM8K	0.524 / **0.533**	**0.541** / 0.540	**0.887** / 0.855	**0.672** / 0.662	**66.2** / 103.5	**0.346** / 0.357
	MATH	0.543 / **0.547**	0.553 / **0.554**	**0.871** / 0.835	**0.677** / 0.666	**64.0** / 105.4	0.340 / **0.331**
	AR-LSAT	0.510 / **0.513**	0.523 / **0.529**	**0.534** / 0.489	**0.528** / 0.508	**68.0** / 98.6	**0.372** / 0.374
Qwen	StrategyQA	**0.844** / 0.832	0.907 / **0.917**	**0.742** / 0.705	**0.816** / 0.797	**107.2** / 202.0	**0.106** / 0.116
	MuSiQue	**0.639** / 0.627	0.858 / **0.880**	**0.552** / 0.505	**0.672** / 0.642	**282.9** / 392.8	**0.200** / 0.221
	GSM8K	**0.576** / 0.540	0.614 / **0.637**	**0.738** / 0.562	**0.670** / 0.597	**82.9** / 195.7	0.115 / **0.091**
	MATH	**0.581** / 0.563	0.617 / **0.652**	**0.715** / 0.556	**0.662** / 0.600	**103.1** / 201.3	**0.162** / 0.168
	AR-LSAT	**0.496** / 0.493	0.528 / **0.530**	**0.182** / 0.117	**0.270** / 0.192	**207.3** / 299.4	**0.359** / 0.392

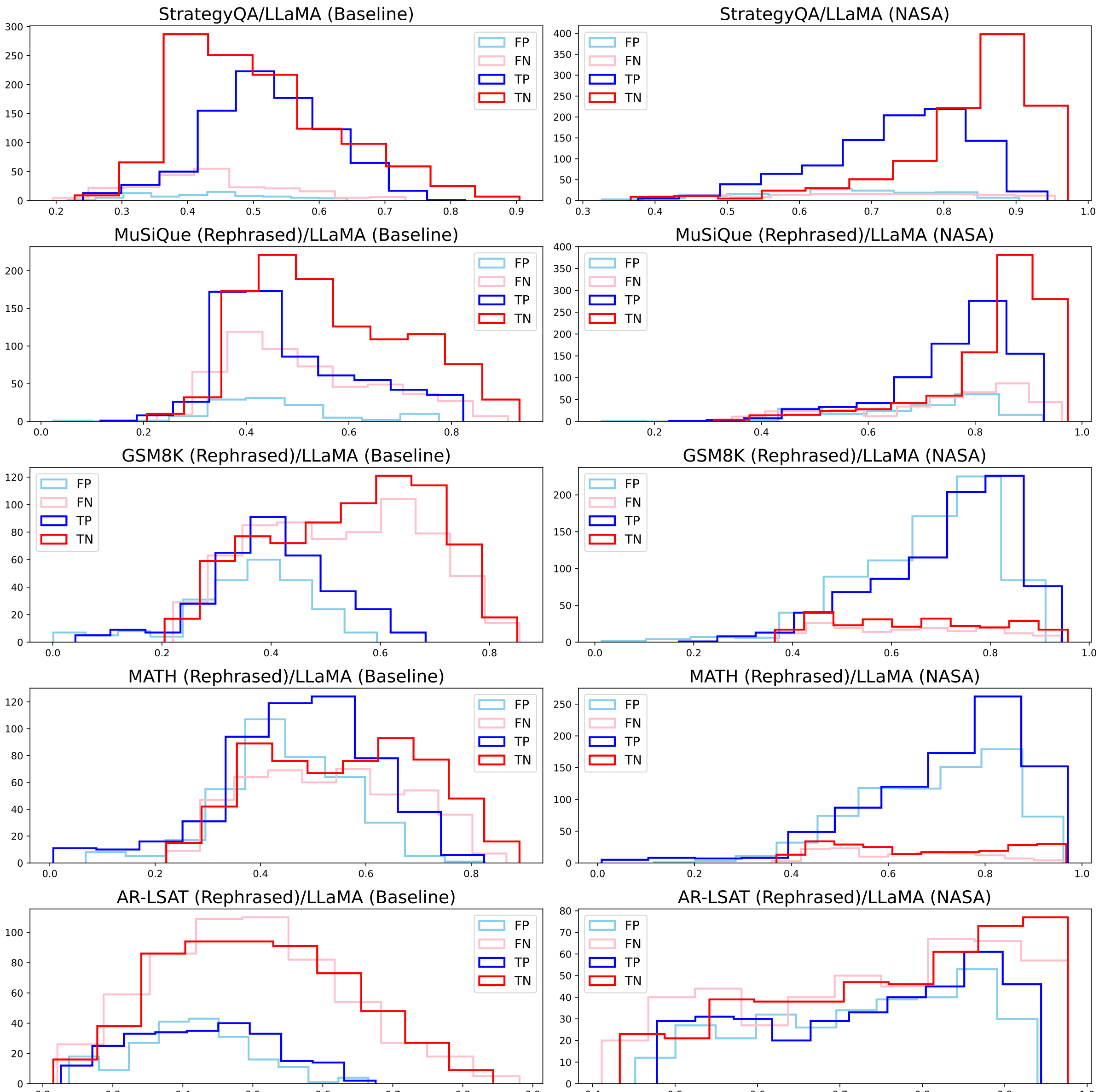

Figure 5: Confidence histograms of LLaMA before and after NASA.

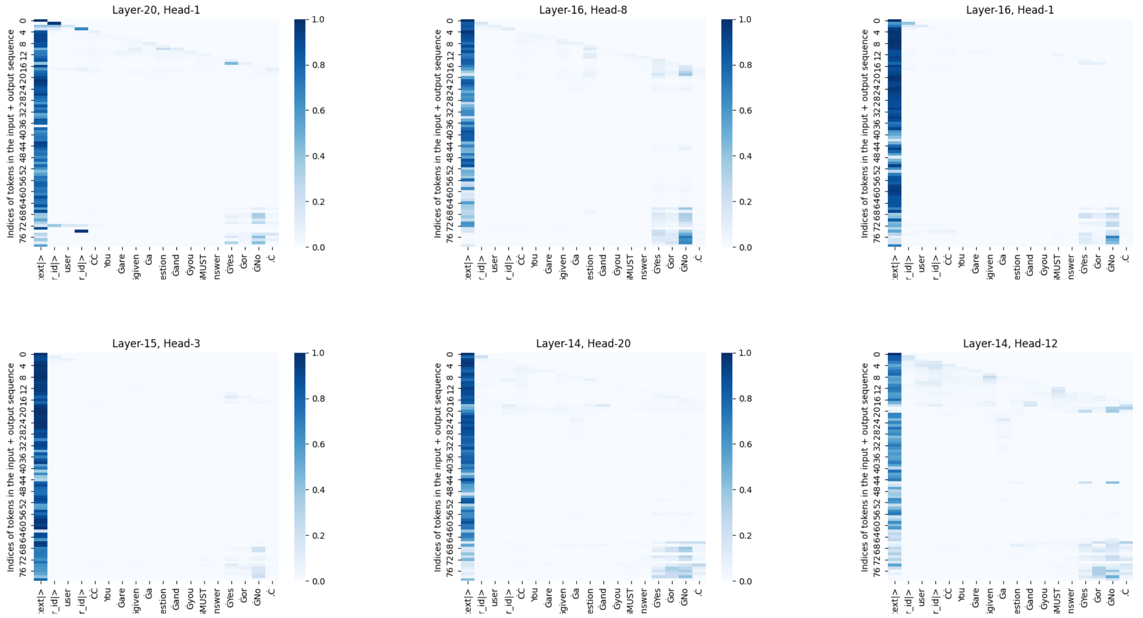

Figure 6: Examples of attention heads of LLaMA for a GSM8K dataset sample.

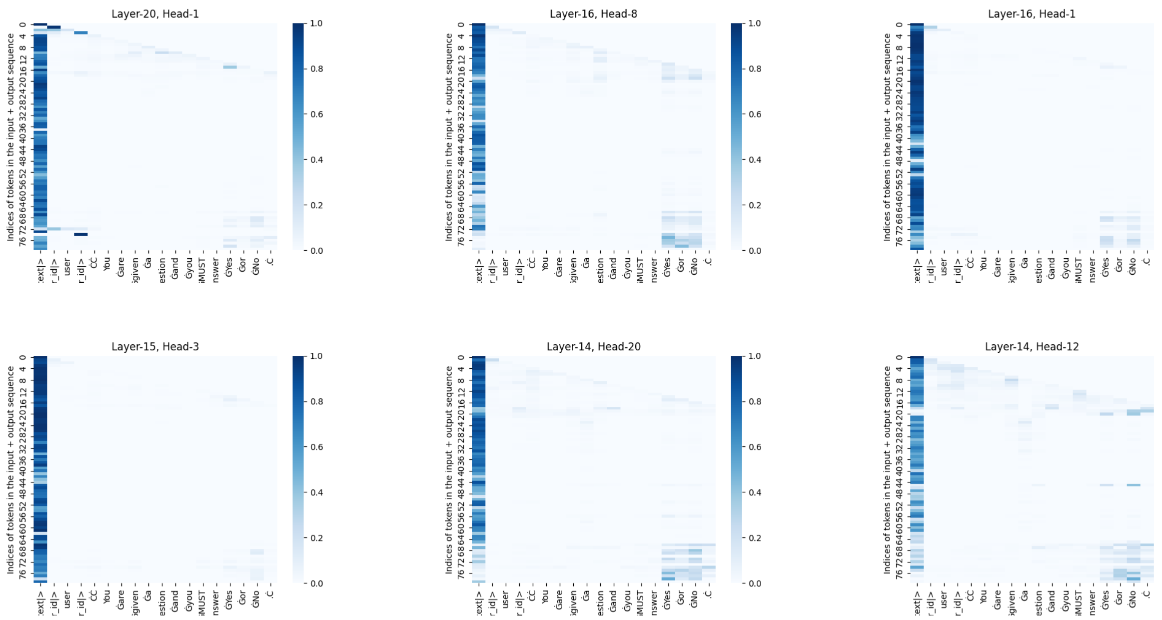

Figure 7: Examples of attention heads of LLaMA model with NASA for a GSM8K dataset sample.