

Persona is a Double-edged Sword: Enhancing the Zero-shot Reasoning by Ensembling the Role-playing and Neutral Prompts

Anonymous ACL submission

Abstract

Recent studies demonstrate that prompting an appropriate role-playing persona to an LLM improves its reasoning capability. However, assigning a proper persona is difficult since an LLM’s performance is extremely sensitive to assigned prompts; therefore, personas sometimes hinder LLMs and degrade their reasoning capabilities. In this paper, we propose a novel framework, Jekyll & Hyde, which ensembles the results of role-playing and neutral prompts to eradicate performance degradation via unilateral use of role-playing prompted LLM and enhance the robustness of an LLM’s reasoning ability. Specifically, Jekyll & Hyde collects two potential solutions from both role-playing and neutral prompts and selects a better solution after cross-checking via an LLM evaluator. However, LLM-based evaluators tend to be affected by the order of those potential solutions within the prompt when selecting the proper solution; thus, we also propose a robust LLM evaluator to mitigate the position bias. The experimental analysis demonstrates that role-playing prompts distract LLMs and degrade their reasoning abilities in 4 out of 12 datasets, even when using GPT-4. In addition, we reveal that Jekyll & Hyde improves reasoning capabilities by selecting better choices among the potential solutions on twelve widely-used reasoning datasets. We further show that our proposed LLM evaluator outperforms other baselines, proving the LLMs’ position bias is successfully mitigated.

1 Introduction

Recent studies have exhibited that assigning specific characteristics and roles into prompts can activate the role-playing ability of Large Language Models (LLMs), thereby improving their reasoning capabilities (Shanahan et al., 2023). Specifically, some studies have proposed utilizing a hand-crafted persona or investigating various types of jobs and relationships to find the most optimal persona to

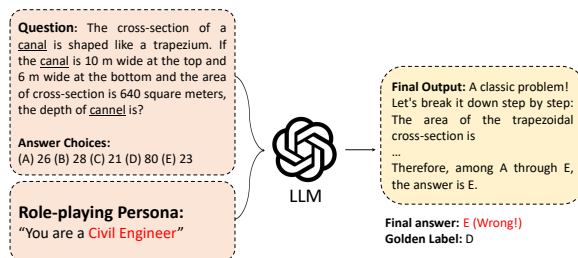


Figure 1: **Persona is a Double-edged Sword.** The example exhibits that prompting the persona to an LLM occasionally leads to deriving the wrong answer. Given a mathematical problem related to civil engineering, the following example uses "Civil Engineer" as a persona, leading the LLM to derive the wrong answer.

enhance the model’s reasoning capability (Kong et al., 2024; Zheng et al., 2023).

However, despite the benefits of enhancement via role-playing persona, persona prompting can sometimes confuse LLMs (Zheng et al., 2023; Gupta et al., 2023); thereby, LLMs provide incorrect solutions to reasoning problems. As shown in Figure 1, an LLM often answers the given question incorrectly due to the assigned persona. The given example exhibits that the role-playing persona is guessed as "Civil Engineer" due to the given situation illustrated in the question; however, since the given question is a math problem, the LLM ends up getting the wrong answer. This phenomenon is crucially problematic when an LLM without a persona can correctly answer the given question. For deeper insights, we first conduct an experiment that compares the LLM’s correctness under the condition of whether the persona is assigned or not. Table 1 presents a confusion matrix of an experimental result for executing an LLM with persona and without persona on the AQuA dataset. It exhibits that 15.75% of the reasoning problems become correct when using an LLM with persona compared to without persona. On the other hand, 13.78% of the problems are answered incorrectly when using an

Method	Persona Solver (w/ Persona)		
	AQuA	Wrong	Right
Neutral Solver (w/o Persona)	Wrong	33.07%	15.75%
	Right	13.78%	37.40%

Table 1: **Confusion matrix between Neutral Solver (w/o Persona) and its Persona Solver (w/ Persona) on AQuA dataset.** we compute each model’s correctness, exhibiting the result in a confusion matrix form. Neutral Solver and Persona Solver means an LLM without persona and an LLM with persona, respectively. More analysis for other datasets are included in Appendix D.

LLM with a persona rather than without a persona. This experiment reveals that assigning a persona to an LLM sometimes fails to enhance its reasoning ability.

To address the aforementioned limitation, we propose a novel framework called **Jekyll & Hyde** that ensembles the solutions of role-playing and neutral prompts to enhance the model’s reasoning ability for a given task. Specifically, we execute an LLM with a role-playing prompt and a non-persona prompt to obtain each solution, respectively, and then utilize another LLM evaluator to judge which one is the better solution. We use an LLM-generated persona for the role-playing prompt to enhance efficacy and efficiency, which is more effective than using a handcrafted persona.

In addition, we also propose a novel robust position bias mitigation method that can be used in other LLM evaluation cases. Previous studies show several issues with using an LLM as an evaluator, and the most challenging problem is the existence of position bias that occurs by the order of solutions within the prompt (Zheng et al., 2024; Li et al., 2023; Wang et al., 2023). Our method alternately inserts two sequences of solutions (in forward and reverse orders) into the evaluation prompt and executes the LLM until both evaluation results are identical without exceeding a pre-defined number of attempts.

Experimental results demonstrate that Jekyll & Hyde significantly outperforms the case of using only a single role-playing LLM or a neutral LLM for three widely-used LLMs. For example, Jekyll & Hyde outperforms the baselines by an average of 9.98% accuracy across twelve datasets when using GPT-4 as a backbone model. In addition, we show that utilizing an LLM-generated persona is better than using a handcrafted persona in terms of the stability of the LLM’s performance. We also show

that using the same LLM for generating a persona and solving reasoning questions improves the performance in reasoning tasks than using different LLMs. Furthermore, our method for mitigating position bias outperforms the existing methods, requiring execution trials that are comparable to those of the others. To the best of our knowledge, this work is the first to provide a systematic investigation of the impact of incorporating LLMs with personative and neutral perspectives.

2 Related Works

2.1 Role-playing abilities of LLMs

Large language models have demonstrated significant eligibility in personating various roles, which shows the power of the role-playing capabilities of LLMs (Kong et al., 2024; Zheng et al., 2023). From this consensus, several studies have tried to investigate the positive effect of role assignment on improving the LLM’s performance. Zheng et al. (2023) have dissected the impact of role assignment towards the LLM by assigning various types of persona, including job names and relationship keywords. This study has concluded that using a domain-specific persona related to the given question enhances the performance of the LLM.

However, although the role-playing ability enables the LLM to generate multiple viewpoints of the model, a persona may also generate bias, which distracts the model and causes performance degradation. Gupta et al. (2023) have analyzed how assigning persona combined with social demographical details brings bias toward the LLM, significantly dropping the performance on reasoning tasks. Deshpande et al. (2023) have investigated the toxicity score for each persona combined with specific entities such as age, sexual orientation, etc., and found out that using particular names and adding specific entities into the persona generates a toxic and biased response, deteriorating the LLM’s performance.

2.2 Position bias in LLM Evaluators

After the advent of LLMs, there have been attempts to use LLMs as evaluators for selecting a better solution generated from two different LLMs. However, LLM evaluators are exposed to a significant problem, position bias, which hinders LLMs from evaluating faithfully and finally degrades their performance. Position bias is attributed to the given order of two solutions in prompts, and LLMs tend to

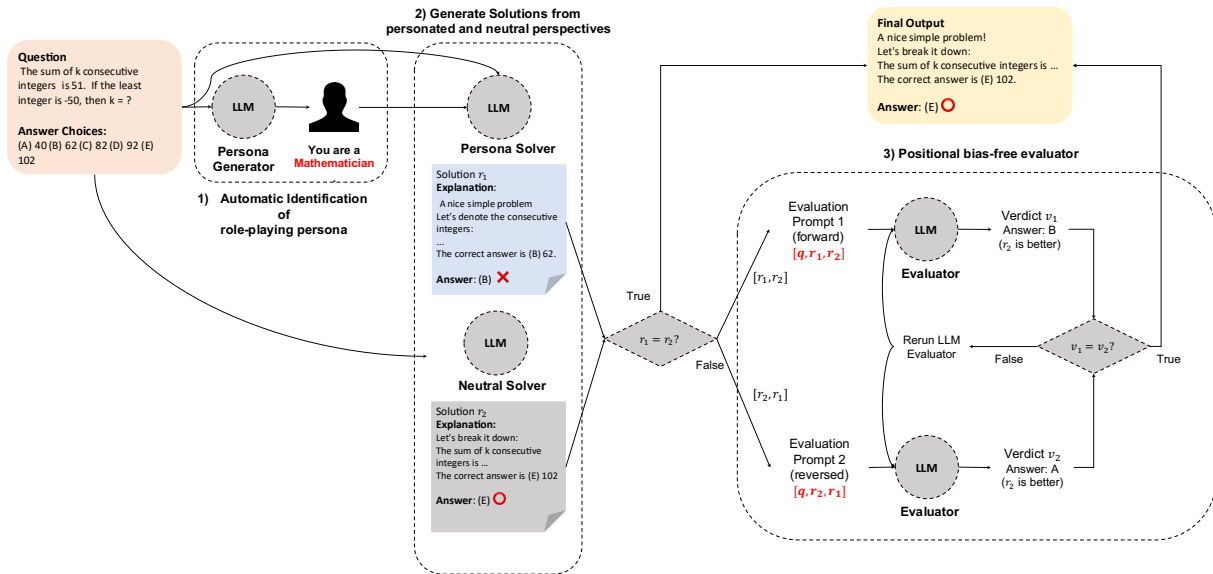


Figure 2: **The architecture of Jekyll & Hyde.** Jekyll & Hyde utilizes not only persona-assigned LLM (**Persona Solver**) but also LLM without prompting (**Neutral Solver**), which provides a dual perspective towards the given question. This structure improves the model to gain potentially high performance. After executing both LLMs, a robust Evaluator whose positional bias is mitigated selects a better solution between the two solutions.

157 be affected by the order and dominantly select the
 158 solution that appears in a specific position. There-
 159 fore, prior studies have tried to solve the problem
 160 using various methods to mitigate position bias. Li
 161 et al. (2023) have introduced a technique called
 162 "Portia," which splits the given two solutions into
 163 several chunks and inserts the chunks alternately
 164 into the evaluation prompt, mitigating the infor-
 165 mation about the order of the solutions inside the
 166 prompt. However, it has failed to completely miti-
 167 gate position bias since the order of the solutions
 168 still remains in each chunk. Wang et al. (2023)
 169 have proposed a method to evaluate solutions, com-
 170 puting the score in a particular scale and running
 171 the evaluation twice by ordering two solutions re-
 172 versely to mitigate position bias. They have aver-
 173 aged to aggregate the two scores to use it as the
 174 final score. On the other hand, in this process,
 175 the scales of scores and the score's relative differ-
 176 ence may vary between two different evaluation
 177 prompts. Therefore, simply computing the aver-
 178 age of the scores and adopting the assistant with
 179 the higher average score may derive inconsistent
 180 results.

181 3 Methods

182 In this section, we demonstrate the process of Jekyll
 183 & Hyde. Specifically, Jekyll & Hyde consists of
 184 three different LLM modules: **Persona generator**,
 185 **Solver**, and **Evaluator**. The pipeline of Jekyll &

186 Hyde is the following: First, the Persona generator
 187 generates an appropriate persona based on a given
 188 question. Then, two different LLM solvers (i.g.,
 189 Persona Solver and Neutral Solver) are executed
 190 simultaneously to get two solutions, respectively.
 191 Finally, the Evaluator compares two solutions and
 192 derives the final prediction based on the selection
 193 from the evaluator. Figure 2 describes the entire
 194 framework of Jekyll & Hyde.

195 3.1 Automatic Identification of Persona

196 The common practice of role-playing prompting
 197 prepends a persona role (e.g., Mathematician) into
 198 the prompt that contains the question. While these
 199 conventional role-playing methods work properly,
 200 prior works have also noticed that persona often
 201 brings bias to LLMs when the question is not
 202 strongly related to the role assignment (Gupta et al.,
 203 2023; Deshpande et al., 2023). In addition, prior
 204 works have manually assigned roles to LLMs; thus,
 205 those methods are labor-intensive to designate an
 206 appropriate role for each data instance. To address
 207 these drawbacks, we modify the common practice
 208 of manually inferring a specialized role for a given
 209 question. Specifically, we utilize an LLM (Per-
 210 sona generator) to guess an appropriate role for the
 211 given question by utilizing an instruction-following
 212 prompt that guides the LLM to automatically gen-
 213 erate a persona that could possibly solve the given
 214 question. More details about the effect of leverag-

ing LLM-generated persona are discussed in section 4.3.

3.2 Generating personated and neutral perspective solutions

After identifying a proper persona, it is formatted as a role-playing prompt and inserted inside the input query for an LLM. This role-playing prompt typically enhances the performance of LLMs effectively. However, using a persona prompt can be a double-edged sword since some data instances may not be appropriately represented by the persona. Therefore, we propose to ensemble two different LLM Solvers, specified as **Persona Solver** and **Neutral Solver**. Persona Solver is an LLM that uses role-playing prompting, utilizing the persona by inserting it inside the query. Neutral Solver does not allow persona prompting, which directly inserts the query into the LLM. This dual execution approach provides two different perspectives on solving the question and derives two discriminative responses. By recalling table 1, if we execute two solvers (i.g., Persona Solver and Neutral Solver) and ideally choose the correct answer between two responses, we are able to achieve better performance than using a single solver via correctly answering the question that is contained in first, second, and the third quadrant of the confusion matrix. We follow the identical implementation introduced in Kong et al. (2024) when utilizing the Neutral Solver. In the case of implementing the Persona Solver, we use a prompt in the format of "You are a \$persona", inserting a generated persona (described in the Section 3.1) to the "\$persona" part.

3.3 Aggregating solutions of two Solvers

Two solutions generated from Neutral Solver and Persona Solver are inserted into the evaluation prompt, which works as the Evaluator’s query. Specifically, two solutions are formatted to the evaluation prompt, creating an order between the solutions. The format of the evaluation prompt can be found in Appendix A. Then, the Evaluator returns its verdict, which is denoted in "A" if the first-ordered solution is better or "B" if the second-ordered solution is better. Note that LLM Evaluator requires two hyper-parameters, involving the temperature of the LLM τ , and the max number of attempt k which is going to be explained in section 3.4.

3.4 Robust evaluation via mitigating bias

As introduced in Section 3.3, the Evaluator returns the final output between two solutions; however, this method may be exposed to position bias, which degrades the total performance of the framework. According to previous studies, position bias occurs due to the order of the solutions (Zheng et al., 2024; Li et al., 2023; Wang et al., 2023). Therefore, we run the Evaluator model twice by inserting the solutions into the evaluation prompt and reversing the order of the solutions to mitigate the following bias. Formally, given a question q , the solution of the Neutral Solver r_1 , and the solution of the Persona Solver r_2 , we depict the process of the position bias mitigation as the following:

$$\begin{aligned} v_1 &= \operatorname{argmax}_v \mathcal{P}(v|q, r_1, r_2) \\ v_2 &= \operatorname{argmax}_v \mathcal{P}(v|q, r_2, r_1) \end{aligned} \quad (1)$$

where \mathcal{P} is the Evaluator, and v_1 and v_2 are verdicts derived from the double executions of the Evaluator, respectively. Note that every time when evaluations are executed for generating their decision, we count the number of trials t until it reaches the maximum trial k defined as a hyper-parameter. Then, the framework compares two decisions, whether they are consistent or not. The process finally reaches the end when these two decisions are the same as the following formula:

$$v_{final} = \begin{cases} v_1 & \text{if } v_1 = v_2 \text{ and } t < k \\ \text{Can't answer} & \text{if } t \geq k \end{cases} \quad (2)$$

If t gets bigger than k , we conclude that the Evaluator is significantly exposed to position bias for two solutions. Therefore, Jekyll & Hyde returns "Can't answer" as a final output since it is significantly dangerous to narrow down only one solution in this case.

4 Experiments

4.1 Experimental setup

Datasets. We conduct our experiments across twelve datasets used in prior research (Kong et al., 2024; Kojima et al., 2022) categorized in 4 categories: (1) **Arithmetic**, including MultiArith (Roy and Roth, 2015), GSM8K (Cobbe et al., 2021), AddSub (Hosseini et al., 2014), AQUA-RAT (Ling et al., 2017), SingleEq (Koncel-Kedziorski et al., 2015), and SVAMP (Patel et al., 2021) (2) **Commonsense reasoning**, including CSQA (Talmor

Models	Method	Arithmetic						Average
		Multiarith	GSM8K	AddSub	AQuA	SingleEq	SVAMP	
GPT-4 (220B*8)	Base	98.44	92.97	97.13	68.24	98.56	91.00	91.06
	Persona	97.78	94.06	97.55	74.80	98.56	90.90	92.28
	Jekyll & Hyde	98.00	95.27	97.72	76.90	98.95	92.03	93.15
GPT-3.5-turbo (20B)	Base	95.72	81.40	90.97	62.60	97.83	80.17	84.78
	Persona	96.50	83.27	93.08	64.44	97.31	84.13	86.45
	Jekyll & Hyde	97.56	85.01	92.91	67.98	98.03	84.77	87.71
llama3 (8B)	Base	98.56	78.59	87.76	47.38	94.23	82.30	81.47
	Persona	97.22	81.05	87.17	52.23	91.27	84.97	82.32
	Jekyll & Hyde	98.17	83.02	89.03	54.07	94.62	86.50	84.23

Table 2: **Main results for Arithmetic datasets.** We report accuracy for six arithmetic datasets computed with a Neutral solver (Base), Persona solver (Persona), and Jekyll & Hyde. Bold values mean the best performance among the three methods. We execute each model three times and average their performance.

et al., 2019) and StrategyQA (Geva et al., 2021) (3) **Symbolic reasoning**, including Last Letter Concatenation and Coin Flip (Wei et al., 2022) (4) **Others**, including Date Understanding and Tracking Shuffled Objects from BIG-bench (Srivastava et al., 2022). More details about dataset configuration can be found in Appendix C.

Models. We utilize two black box large language models released from OpenAI, which are GPT-4 (gpt-4-0613) and GPT-3.5-turbo (gpt-3.5-turbo-0125) (OpenAI, 2023), and one open source large language model llama3 (AI@Meta, 2024). These models are used as a backbone model of our framework.

Implementation details. To evaluate Jekyll & Hyde, we testify three cases for each dataset: (1) **Base**, which is the case of only utilizing a Neutral solver where persona is not assigned to LLMs, (2) **Persona**, a case that only uses Persona solver which is an LLM assigned with persona, and (3) **Jekyll & Hyde**, our proposed framework. Specifically, we investigate the model’s performance by computing accuracy with the provided label for each dataset. When using the LLM evaluator in Jekyll & Hyde, the hyper-parameters are set as follows: the max attempt k to 5 and temperature τ to 0.7. Details for determining the hyper-parameters are shown in section 4.6.

4.2 Jekyll & Hyde enhances the reasoning abilities of LLMs

Table 2 shows the performance of different methods on Arithmetic datasets. In addition, Table 3

exhibits the performance of the other datasets (i.g., Commonsense reasoning, Symbolic reasoning, and Others) in the accuracy metric. The result possesses several points. 1) Persona does not always improve the performance of an LLM. According to the result, several datasets show that performance is better when using a Neutral solver rather than a persona-assigned LLM. 2) Using the Jekyll & Hyde framework improves the model performance. By leveraging the LLM evaluator, performance has increased in most datasets, regardless of the model type. This desired behavior shows that the ensemble process aids the task in choosing the correct answers that could not be correctly answered when using only a single perspective LLM.

4.3 Automatic persona generation ensures the robust reasoning ability

In section 3.1, we choose to use an automatically generated persona rather than a manually handcrafted persona for Jekyll & Hyde. We additionally conduct an experiment to reveal that the automated persona generation is not only labor-efficient and flexible but also performs more robustly than handcrafted persona, using Aqua and Object tracking datasets. Using llama3-8B as a backbone model, we run the model three times for each dataset and compute the average performance and standard deviation for each dataset to verify the method’s robustness. For the persona generation, we generate solutions by sampling each word-piece from the LLM’s output probability distribution; thus, it causes the generation of various personas. For the handcrafted persona, we utilize three different hu-

Models	Method	Common Sense		Symbolic Reasoning		Other Tasks		Average
		CSQA	Strategy	Letter	Coin	Date	Object	
GPT-4 (220B*8)	Base	79.91	76.42	19.80	66.93	79.22	45.96	61.37
	Persona	80.89	75.71	92.80	75.93	78.41	58.76	77.08
	Jekyll & Hyde	81.11	77.00	93.00	80.27	82.38	61.69	79.24
GPT-3.5-turbo (20B)	Base	77.31	68.75	18.67	47.53	67.84	34.67	52.46
	Persona	75.40	69.75	45.67	59.20	76.15	40.22	61.07
	Jekyll & Hyde	77.50	70.00	48.93	64.00	76.78	42.22	63.24
llama3 (8B)	Base	74.50	69.21	86.40	95.80	77.42	44.76	74.68
	Persona	72.29	71.21	86.07	95.33	74.44	47.60	74.49
	Jekyll & Hyde	74.97	70.54	86.47	98.67	79.04	48.58	76.38

Table 3: **Main results for Common Sense, Symbolic Reasoning, and Other Tasks Datasets.** We report accuracy for six datasets, including Common Sense, Symbolic Reasoning, and Other tasks. Bold values mean the best performance among the three methods. We execute each model three times and average their performance.

man handcrafted personas for each run to check the robustness of the performance. Specifically, we manually select three appropriate handcrafted personas for the Aqua dataset: *Math teacher*, *Mathematician*, and *Math Tutor*. For the Object tracking dataset, we leverage *Observer*, *Recorder*, and *Logical Reasoner*. As shown in table 5, although the two methods show comparable performance on accuracy, we observe that the standard deviation highly increases when using handcrafted personas. This implies that performance could vary depending on the manually designated persona when using a handcrafted persona. Thus, we believe that using an LLM-generated persona stabilizes the model and gives it a smaller margin of error than a manually handcrafted persona.

4.4 Reasoning ability increases when using the same LLM for each module consistently

Due to the impact of role-playing prompting towards LLMs, assigning a persona enhances the reasoning capability of LLMs. In Jekyll & Hyde, we adopt a process that generates an appropriate persona for the given question by utilizing three types of LLMs. For the detailed analysis, we conduct an additional experiment to reveal whether a persona generated from different LLM affects the LLM solver’s performance. Specifically, we investigate the relationship between Persona Generator and Persona Solver by substituting the backbone model of Persona Solver with another LLM and computing the average accuracy for each dataset. We use llama3-8B as the persona generator fixedly

and run three LLMs as our backbone model for the persona solver, involving llama3-8B, GPT-3.5-turbo, and GPT-4. We utilize five different datasets: Aqua, AddSub, Coin, Date understanding, and Object tracking, and the results can be found in table 6. As shown, we notice that using the same LLM as a backbone model reveals the optimal performance, while using different LLMs for the Persona Solver degrades the performance.

4.5 Jekyll & Hyde effectively mitigates position bias

To enhance the performance of the task, it is necessary to mitigate position bias from Evaluator. For further analysis of the framework’s mitigation process, we compare the performance of Jekyll & Hyde with the existing two position bias mitigation methods: (1) Portia (Li et al., 2023) and (2) MEC+BPC (Wang et al., 2023). For better investigation, we also observe the idealistic case when the Evaluator always picks the desirable gold answer among the answers of Neutral Solver and Persona Solver, namely the Oracle Evaluator. We use six different datasets to test the general use case. Position bias mitigation experimental results can be found in table 4. According to the result, we reveal that the evaluator within Jekyll & Hyde derives the best performance among the other methods from most datasets, regardless of the type of the backbone model. The details for implementation of Portia and MEC+BPC is shown in Appendix E.

Models	Method	AddSub	AQuA	SingleEq	SVAMP	Coin	Date
GPT-4	Oracle Evaluator	97.72	81.10	99.41	95.20	88.80	82.66
	Portia ($k = 3$)	97.47	74.41	98.82	91.80	74.40	80.76
	MEC+BPC ($k = 3$)	97.22	74.41	98.43	91.20	74.00	79.95
	Jekyll & Hyde (w/o mitigation)	97.72	78.35	98.43	92.20	78.20	80.22
	Jekyll & Hyde	97.72	77.56	99.02	92.60	79.80	81.57
GPT-3.5-turbo	Oracle Evaluator	95.19	74.41	99.21	87.10	60.80	80.22
	Portia ($k = 3$)	91.14	62.60	98.23	81.80	57.80	72.63
	MEC+BPC ($k = 3$)	89.37	62.60	97.64	80.20	57.60	75.61
	Jekyll & Hyde (w/o mitigation)	92.15	62.60	97.83	82.50	56.60	72.63
	Jekyll & Hyde	93.16	64.17	98.23	83.00	59.60	72.63
llama3 (8B)	Oracle Evaluator	92.41	63.39	96.06	90.20	99.00	84.55
	Portia ($k = 3$)	88.35	51.97	93.31	86.10	96.40	78.86
	MEC+BPC ($k = 3$)	88.10	55.91	91.73	84.50	95.40	81.03
	Jekyll & Hyde (w/o mitigation)	90.38	51.18	94.29	86.10	97.00	79.95
	Jekyll & Hyde	91.14	53.54	95.67	86.80	98.40	79.95

Table 4: **Mitigating positional bias.** We report that the LLM Evaluator used for Jekyll & Hyde outperforms other existing methods in most datasets. Despite the marginal increase when using the LLM Evaluator from Jekyll & Hyde, the Evaluator aids the LLM to nearly approach the performance of an Oracle Evaluator, which is the optimal performance for the given datasets.

Model	Datasets	Methods	Average Accuracy (\uparrow)	Standard Deviation (\downarrow)
llama3 (8b)	AQuA	handcrafted persona	51.71	6.11
		expert persona	50.66	2.08
	Object	handcrafted persona	44.31	8.02
		expert persona	46.71	3.06

Table 5: **Standard deviation of handcrafted persona LLM and LLM generated persona LLM** We compute the standard deviation for each dataset after running three times in order to check the stability of the model output. As is shown, using an expert persona generates a smaller value of standard deviation for two datasets, resulting in utilizing an LLM-generated persona consistently yields robust output.

4.6 Hyper-parameter experiments for the Evaluator

The number of max attempts (k). We experiment with each hyper-parameter to examine their impact on the framework’s performance. For the number of max attempts of the Evaluator, we compare four different values of $k \in \{1, 2, 5, 10\}$, by computing the framework’s performance. For the experiment, we utilize four datasets, namely MultiArith, SingleEq, Aqua, and Date Understanding. As shown in figure 3-(a), we compare the experimental results executed from llama3-8B as a backbone model and reveal the performance of the

Persona generator	Datasets	Persona Solver		
		llama3-8B	GPT-3.5-turbo	GPT-4
llama3-8B	AQuA	53.15	52.36	53.54
	AddSub	88.35	81.77	82.53
	Coin	95.00	90.20	92.80
	Date	74.80	71.54	72.63
	Object	49.07	46.93	50.93
	Average	72.07	68.56	70.49

Table 6: **Comparison of average accuracy of Persona Solver when using different LLMs for Persona Generator.** We use the fixed Persona generator (i.g., llama3-8B) and evaluate the performance of three different Persona Solvers (i.g., llama3-8B, GPT-3.5-turbo, and GPT-4) for the Jekyll & Hyde framework. Bold values mean the highest performance among different LLMs.

framework increases as the number of attempts increases. Furthermore, we could identify that Jekyll & Hyde could outperform the single perspective LLM when the max attempt k is bigger than 1. Despite the framework’s improvement, increasing the number of attempts leads to a high-cost burden on the framework. Since the enhancement of the framework is getting smaller as the number of the max attempts increases, we decide to use $k = 5$ as our default setting, which can balance the framework’s performance and prevent fining the model excessively.

Dataset	Question (q)	Neutral Solver(r_1)	Persona Solver(r_2)	Evaluator[q, r_1, r_2] result	Evaluator[q, r_2, r_1] result	Final prediction
AQuA	Two ants are standing side-by-side. One ant, which is 4 inches tall, casts a shadow that is 10 inches long. The other ant is 6 inches tall. Compute, in inches, the length of the shadow that the taller ant casts. Answer Choices: (A) 36 (B) 28 (C) 42 (D) 15 (E) 20	explanation: A classic problem! Let's use the concept of similar triangles to solve this. We can draw a diagram: In this diagram, the ratio of the height of the ant to the length of its shadow is the same for both ants. Let's call this ratio "k". ... we can multiply the height by the ratio k to get: Shadow length = 6 x (2/5) = 12 inches. So, the correct answer is: (C) 42. Therefore, among A through E, the answer is C persona: None answer: C	explanation: A classic problem! Let's use the concept of similar triangles to solve this. We can draw a diagram: In this diagram, the ratio of the height of the ant to the length of its shadow is the same for both ants. ... The correct answer is (D) 15. Therefore, among A through E, the answer is D persona: Mathematician answer: D	verdict: r_2	verdict: r_2	D
Strategy	Did anyone in the 1912 election take a majority of the popular vote?	explanation: In the 1912 United States presidential election, Woodrow Wilson, the Democratic candidate, won the majority of the popular vote, receiving 41.8% of the total votes cast. However, ... Theodore Roosevelt won the majority of the electoral votes. Therefore, the answer (Yes or No) is no persona: None answer: no	explanation: "In the 1912 presidential election, Woodrow Wilson, the Democratic candidate, won the majority of the popular vote, receiving 41.8% ... So, to answer your question, yes, Woodrow Wilson did take a majority of the popular vote in the 1912 election. Therefore, the answer (Yes or No) is yes persona: Historical Election Analyst answer: yes	verdict: r_1	verdict: r_1	no
Object	Alice, Bob, and Claire are holding a white elephant gift exchange. At the start of the event, they are each holding a present of a different color: Alice has an orange ball, Bob has a purple present, and Claire has a blue present. As the event progresses, pairs of people swap gifts. First, Claire and Alice swap their gifts. Then, Alice and Bob swap their gifts. Finally, Bob and Claire swap their gifts. At the end of the event, Alice has the Which choice is true? Answer Choices: (A) orange ball. (B) purple present. (C) blue present.	explanation: Let's follow the gift-swapping process: 1. Claire and Alice swap gifts: Claire gets the orange ball, and Alice gets the blue present. 2. ... At the end of the event, Alice has the purple present. Therefore, the correct answer is: (B) purple present. Therefore, among A through C, the answer is B persona: None answer: B	explanation: A fun gift exchange problem! Let's break it down step by step: 1. Claire and Alice swap gifts: Claire gets the orange ball, and Alice gets the blue present. 2. ... So, the correct answer is: (A) orange ball. Therefore, among A through C, the answer is A persona: Gift Exchange Analyst answer: A	verdict: r_1	verdict: r_1	B

Table 7: **Qualitative analysis.** We conduct a qualitative analysis of Jekyll & Hyde on three datasets. Jekyll & Hyde’s Persona Solver attains dynamic persona based on the Persona Generator (e.g., Classical Mythology Scholar). Additionally, examples reveal that Evaluator is working properly via selecting the correct answer among different solutions from each Solver.

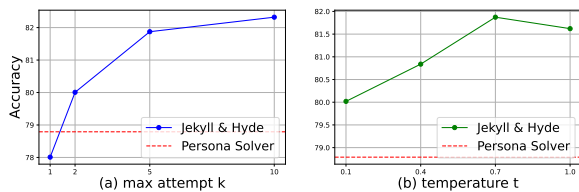


Figure 3: **Hyper-parameters Experiments.** Variation of averaged accuracy with a (a) various number of max attempt k and (b) temperature of the LLM τ used in LLM evaluator. X and Y axes correspond to each hyper-parameter setting and accuracy, respectively.

The temperature of the Evaluator (τ). We further investigate the impact of the Evaluator’s token generation temperature by comparing the framework’s performance. Specifically, we utilize four different temperatures $\tau \in \{0.1, 0.4, 0.7, 1.0\}$ to examine how the generation diversity affects the performance of the Evaluator. Figure 3-(b) shows that temperature $\tau = 0.7$ exhibits the optimal performance among others.

4.7 Qualitative analysis

We qualitatively analyze our method on three different datasets, namely Aqua, StrategyQA, and Object tracking. Table 7 exhibits the result of qualitative analysis. These results reveal that the persona is dynamically generated based on the question from the LLM, stating different personas (i.g., Mathematician, Gift Exchange Analyst) for different arithmetic questions. Additionally, running the proposed evaluator eventually gives a consistent verdict, which mostly ends up deriving the correct output for the given question. This meticulous process aids the framework in mitigating position bias that

exists inside the LLM when using it as an evaluator, increasing the performance of the framework.

5 Conclusion

In this paper, we propose Jekyll & Hyde, a novel framework that solves the reasoning problem by ensembling personated and neutral perspectives. Evaluations across twelve representative reasoning benchmark datasets present that our framework surpasses both cases when the persona is assigned or not on most datasets. In addition, our framework’s novel method of mitigating position bias has shown better performance when compared with existing methods. These results highlight the impact of utilizing LLM with and without persona to improve performance in reasoning tasks. Overall, this work sets the initial stage for further investigation in combining solutions from different viewpoints within the LLM community, a promising research direction for improving reasoning abilities.

6 Limitations

Although existing methods require additional computation cost and Jekyll & Hyde is comparable in efficiency to them, our method still requires more computation cost than single perspective LLMs since it runs the LLM at least twice for a single instance. In practice, users can leverage Jekyll & Hyde by setting the maximum attempt of the model into 2, which still outperforms using a single perspective LLM. In addition, the performance of Jekyll & Hyde is upper-bounded since questions that both perspective LLMs answered incorrectly can not be answered correctly using Jekyll

513	& Hyde. These aspects of exploration are left to	Split and merge: Aligning position biases in large	566
514	future works.	language model based evaluators. <i>arXiv preprint</i>	567
		<i>arXiv:2310.01432</i> .	568
515	References		
516	AI@Meta. 2024. Llama 3 model card.	Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blun-	569
517	Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian,	som. 2017. Program induction by rationale genera-	570
518	Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias	tion: Learning to solve and explain algebraic word	571
519	Plappert, Jerry Tworek, Jacob Hilton, Reiichiro	problems. In <i>Proceedings of the 55th Annual Meet-</i>	572
520	Nakano, et al. 2021. Training verifiers to solve math	<i>ing of the Association for Computational Linguistics</i>	573
521	word problems. <i>arXiv preprint arXiv:2110.14168</i> .	(<i>Volume 1: Long Papers</i>), pages 158–167, Vancouver,	574
		Canada. Association for Computational Linguistics.	575
522	Ameet Deshpande, Vishvak Murahari, Tanmay Rajpuro-	OpenAI. 2023. Gpt-4 technical report.	576
523	hit, Ashwin Kalyan, and Karthik Narasimhan. 2023.		
524	Toxicity in chatgpt: Analyzing persona-assigned lan-	Arkil Patel, Satwik Bhattamishra, and Navin Goyal.	577
525	guage models. In <i>Findings of the Association for</i>	2021. Are NLP models really able to solve simple	578
526	<i>Computational Linguistics: EMNLP 2023</i> , pages	math word problems? In <i>Proceedings of the 2021</i>	579
527	1236–1270.	<i>Conference of the North American Chapter of the</i>	580
		<i>Association for Computational Linguistics: Human</i>	581
528	Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot,	<i>Language Technologies</i> , pages 2080–2094, Online.	582
529	Dan Roth, and Jonathan Berant. 2021. Did aristotle	Association for Computational Linguistics.	583
530	use a laptop? a question answering benchmark with		
531	implicit reasoning strategies. <i>Transactions of the</i>	Subhro Roy and Dan Roth. 2015. Solving general arith-	584
532	<i>Association for Computational Linguistics</i> , 9:346–	metic word problems. In <i>Proceedings of the 2015</i>	585
533	361.	<i>Conference on Empirical Methods in Natural Lan-</i>	586
		<i>guage Processing</i> , pages 1743–1752, Lisbon, Portu-	587
534	Shashank Gupta, Vaishnavi Shrivastava, Ameet Desh-	gal. Association for Computational Linguistics.	588
535	pande, Ashwin Kalyan, Peter Clark, Ashish Sabhar-		
536	wal, and Tushar Khot. 2023. Bias runs deep: Implicit	Murray Shanahan, Kyle McDonell, and Laria Reynolds.	589
537	reasoning biases in persona-assigned llms. <i>arXiv</i>	2023. Role play with large language models. <i>Nature</i> ,	590
538	<i>preprint arXiv:2311.04892</i> .	623(7987):493–498.	591
539	Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren	Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao,	592
540	Etzioni, and Nate Kushman. 2014. Learning to solve	Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch,	593
541	arithmetic word problems with verb categorization.	Adam R Brown, Adam Santoro, Aditya Gupta,	594
542	In <i>Proceedings of the 2014 Conference on Empirical</i>	Adrià Garriga-Alonso, et al. 2022. Beyond the	595
543	<i>Methods in Natural Language Processing (EMNLP)</i> ,	imitation game: Quantifying and extrapolating the	596
544	pages 523–533, Doha, Qatar. Association for Com-	capabilities of language models. <i>arXiv preprint</i>	597
545	putational Linguistics.	<i>arXiv:2206.04615</i> .	598
546	Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yu-	Alon Talmor, Jonathan Herzig, Nicholas Lourie, and	599
547	taka Matsuo, and Yusuke Iwasawa. 2022. Large lan-	Jonathan Berant. 2019. CommonsenseQA: A ques-	600
548	guage models are zero-shot reasoners. <i>Advances in</i>	tion answering challenge targeting commonsense	601
549	<i>neural information processing systems</i> , 35:22199–	knowledge. In <i>Proceedings of the 2019 Conference</i>	602
550	22213.	<i>of the North American Chapter of the Association for</i>	603
		<i>Computational Linguistics: Human Language Tech-</i>	604
551	Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish	<i>nologies, Volume 1 (Long and Short Papers)</i> , pages	605
552	Sabharwal, Oren Etzioni, and Siena Dumas Ang.	4149–4158, Minneapolis, Minnesota. Association for	606
553	2015. Parsing algebraic word problems into equa-	Computational Linguistics.	607
554	tions. <i>Transactions of the Association for Computa-</i>		
555	<i>tional Linguistics</i> , 3:585–597.	Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu,	608
556	Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong	Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and	609
557	Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiao-	Zhifang Sui. 2023. Large language models are not	610
558	hang Dong. 2024. Better zero-shot reasoning with	fair evaluators. <i>arXiv preprint arXiv:2305.17926</i> .	611
559	role-play prompting. In <i>Proceedings of the 2024</i>		
560	<i>Conference of the North American Chapter of the</i>	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten	612
561	<i>Association for Computational Linguistics: Human</i>	Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le,	613
562	<i>Language Technologies (Volume 1: Long Papers)</i> ,	and Denny Zhou. 2022. Chain-of-thought prompt-	614
563	pages 4099–4113.	ing elicits reasoning in large language models. In	615
		<i>Advances in Neural Information Processing Systems</i> ,	616
564	Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan	volume 35, pages 24824–24837. Curran Associates,	617
565	Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2023.	Inc.	618
		Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan	619
		Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin,	620
		Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024.	621

Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Mingqian Zheng, Jiaxin Pei, and David Jurgens. 2023. Is "a helpful assistant" the best role for large language models? a systematic evaluation of social roles in system prompts. *arXiv preprint arXiv:2311.10054*.

A Prompt Design

In Jekyll & Hyde, we leverage three types of LLMs, namely **Persona Generator**, **Solver**, and **Evaluator**. Since each LLM has different roles, they also have different persona designs. Table 8, 9 shows the Persona Generator and Evaluator prompt, respectively. These prompt designs are followed by (Zheng et al., 2024), and we manually revise them to give better instructions for all LLM baselines.

SystemMessage:

You have a special ability in giving job recommendations that could sufficiently solve the given problem.

HumanMessage:

This is the user's question: {input}

According to the question, recommend a job that can sufficiently solve the user's question. Here are some rules you need to follow:

1. give a description of the job in JSON format with the following keys:
 - job: a specific job name
2. Do not give any reasons or preambles about your response

Output:

Table 8: The template for persona generator with one slot {input}. Based on the given template, the persona generator yields a unified occupation name (e.g. *Math teacher*)

B Solver mechanism

When running the LLM under the zero-shot setting, the response is not fixed in a certain format. To extract the answer from the response, we follow the technique of Zero-Shot CoT (Kojima et al., 2022). In detail, the technique consists of two steps, which first generates the response from the LLM based on role-playing prompting and the given question. Then, we concatenate the question, response from the previous step, and an answer trigger together and input them to the LLM, computing the extracting the final answer from the response. The entire

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below.

Your evaluation should ONLY consider correctness. You will be given assistant A's answer, and assistant B's answer.

Your job is to evaluate which assistant's answer is better. You should independently solve the user question step-by-step first

Then compare both assistants' answers with your answer. Identify and correct any mistakes.

Based on the given two solutions for the following question, you need to choose the best solution based on their explanation and answer

First, solve the problem step by step, and then identify errors and flaws from the given solutions if needed.

Please note that:

1. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision.
2. Do not allow the length of the responses to influence your evaluation.
3. Do not favor certain names of the assistants. Be as objective as possible.
4. Give reason for your choice between two solution.
5. You must output your final verdict by strictly following this format: "[[A]]" if assistant A is better, and "[[B]]" if assistant B is better

This is your user's question: {question}

assistant A's answer: {assistantA_answer}
assistant A's explanation: {assistantA_explanation}

assistant B's answer: {assistantB_answer}
assistant B's explanation: {assistantB_explanation}

Now, begin!
Final verdict:

Table 9: The evaluation template with five slots ({question}, {assistantA_answer}, {assistantA_explanation}, {assistantB_answer}, and {assistantB_explanation}). The final verdict output [[A]] or [[B]]

progress is shown in figure 4. The answer trigger sentences for various datasets are depicted in Table 10.

C Dataset Details

In this section, we briefly introduce twelve datasets spanning four categories below. Specific details are shown in Table 11

Arithmetic. We leveraged the following six datasets: MultiArith, GSM8K, AddSub, AQUA, SingleEq, and SVAMP. All questions in these datasets include a particular scenario and require reasoning based on mathematical knowledge.

Answer Format	Answer Trigger
arabic number	Therefore, the answer (arabic numerals) is
option (A-E)	Therefore, among A through E, the answer is
option (A-C)	Therefore, among A through C, the answer is
yes or no	Therefore, the answer (Yes or No) is
string	Therefore, the final answer is

Table 10: Answer trigger sentences for various answer formats.

1. Answer Generation

System: 'You are a \${Persona}'
User: [Question]

Assistant: [Answer1]

2. Answer Extraction

System: 'You are a \${Persona}'
User: [Question] + [Answer1] + [Answer trigger]

Assistant: [Answer2]

Figure 4: an entire process of how Solver works

Commonsense Reasoning. We employ CommonsenseQA and StrategyQA. Both of them require reasoning based on common sense.

Symbolic Reasoning. we utilize Last letter concatenation and Coin Flip. Last Letter Concatenation demands concatenation of the last letter of the given four words. Coin Flip gives a sequence of operations to flip a coin and asks for the final state of the coin. We utilized these two datasets following the approach of [Kojima et al. \(2022\)](#).

Other Reasoning Tasks. We use Date Understanding and Tracking Shuffled Objects from Bigbench([Srivastava et al., 2022](#)). Date Understanding requires date calculations. Tracking Shuffled Objects gives a sequence of object substitution operations and then asking the final location of a certain object.

D Confusion matrix for other datasets

As shown in Table 1, we reveal that some of the questions are correctly answered with LLMs without role-playing prompting, while getting wrong when using LLM with role-playing prompting. Here, we provide the result of a confusion matrix for other datasets, namely the StrategyQA, Coin Flip, and Object Tracking datasets. Table 12, exhibit the confusion matrix for each dataset, respec-

tively.

E Implementation details for Portia and MEC+BPC

In section 4.5, we conduct an experiment to compare the performance of mitigating position bias. Here, we employed two existing methods, specifically Portia and MEC+BPC.

Portia is introduced by [Li et al. \(2023\)](#), which mitigates position biases by slicing each given response into chunks and putting them alternately into the prompt, mitigating the information of the order between the given responses. We implemented this method by slicing the given response into chunks with fixed lengths and then inserting them alternately into the evaluation prompt.

MEC+BPC is introduced by [Wang et al. \(2023\)](#) to mitigate position bias in the LLM Evaluator. It utilizes two evaluation prompts with differently ordered sequences (in forward and reverse orders) of the response. This method executes each evaluation prompt to estimate the scores of two responses, respectively. After deriving scores for each response, it aggregates and computes the final scores of each response by averaging scores for the two sequences of solutions, respectively. We implemented MEC+BPC by preparing two evaluation prompts for the two sequences. Then, we ran the model and computed the score for each response. For robust answer generation, the model is run three times, and the average of the scores is computed.

Dataset	Answer Format	N_q	L_q	License
SingleEq	arabic number	508	27.4	No License
AddSub	arabic number	395	31.5	Unspecified
MultiArith	arabic number	600	31.8	Unspecified
GSM8K	arabic number	1319	46.9	MIT License
AQUA	option (A-E)	254	51.9	Apache-2.0
SVAMP	arabic number	1000	31.8	MIT License
CommonsenseQA	option (A-E)	1221	27.8	Unspecified
StrategyQA	yes or no	2290	9.6	Apache-2.0
Date Understanding	option (A-F)	369	35.0	Apache-2.0
Object Tracking	option (A-C)	750	91.1	Apache-2.0
Last Letters	string	500	15.0	-
Coin Flip	yes or no	500	37.0	-

Table 11: Relevant information of 12 datasets. N_q denotes the number of questions in each dataset. L_q denotes the average words of questions in each dataset.

Method	Persona Solver (w/ Persona)								
	StrategyQA			Coin Flip			Object Tracking		
	Wrong	Right		Wrong	Right		Wrong	Right	
Neutral Solver (w/o Persona)	Wrong	19.39%	12.31%	Wrong	4.60%	4.00%	Wrong	46.67%	18.13%
	Right	10.31%	57.99%	Right	18.00%	73.40%	Right	12.93%	22.27%

Table 12: Confusion matrix between Neutral Solver (w/o Persona) and its Persona Solver (w/ Persona) on StrategyQA dataset.