Persona is a Double-Edged Sword: Rethinking the Impact of Role-play Prompts in Zero-shot Reasoning Tasks

Anonymous ACL submission

Abstract

Recent studies have shown that prompting large language models (LLMs) with role-playing personas can enhance their reasoning capabilities. While the benefits of role-playing personas in reasoning tasks are widely recognized, it remains uncertain whether a persona aligned with the given dataset can consistently achieve these improvements. In this work, we empirically investigate the potential drawbacks of using dataset-aligned personas (referred to as **coarsely aligned personas**) and introduce Jekyll & Hyde, a novel framework that enhances reasoning robustness by ensembling solutions from both role-playing and neutral (non-persona) prompts. Jekyll & Hyde first predicts an instance-specific persona tailored to each query using an LLM, then generates answers with both persona and neutral prompts, and finally selects the superior output through an LLM-based evaluator. Experimental results claim that across twelve widely used natural language reasoning datasets and three backbone large language models, Jekyll & Hyde consistently outperforms single-perspective LLMs, achieving an average accuracy gain of **9.98%** on GPT-4. We further demonstrate that using instance-aligned personas yields more accurate and stable performance than using dataset-aligned personas.

1 Introduction

Recent studies have exhibited that assigning specific roles to prompts can activate the role-playing ability of Large Language Models (LLMs), improving their reasoning capabilities (Shanahan et al., 2023). Therefore, some studies have proposed using a handcrafted persona or domain expert persona aligned with the given dataset to enhance the reasoning performance of an LLM (Kong et al., 2024; Salewski et al., 2024). Although the benefits of using role-playing personas are empirically proven, since conventional role-playing personas

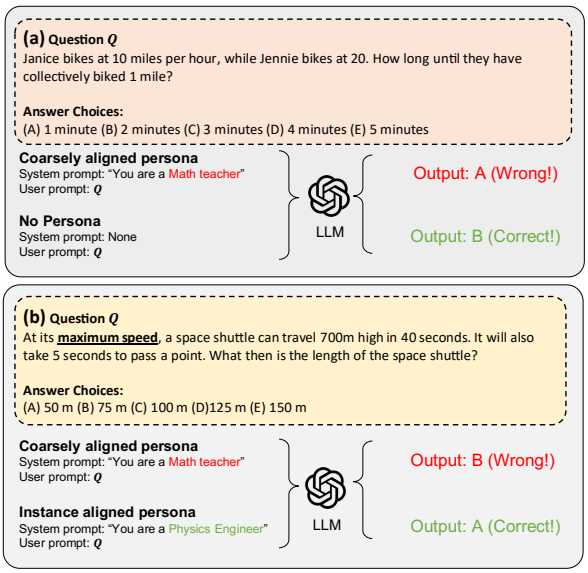

Figure 1: **Persona is a Double-edged Sword.** Case (a) shows that an LLM without a persona can sometimes outperform one with a persona, while case (b) highlights the effectiveness of role-playing persona when properly aligned with the given instance.

are broadly aligned to the given dataset, a deeper examination at the instance level reveals that personas are not universally effective. As shown in Figure 1, an LLM often produces incorrect answers on the AQuA dataset, influenced by the assigned persona. Figure 1-(a) illustrates the case where the role-playing persona inferred as *"Math teacher"*, while seemingly well-aligned for addressing mathematical problems in the dataset, ultimately leads to incorrect answers. In our paper, we refer to such a persona, which is handcrafted and broadly tailored to the given dataset, as a **coarsely aligned persona**. Unlike the case where LLM uses a coarsely aligned persona, the LLM without a persona provides correct answers. Moreover, in Figure 1-(b), the LLM provides the correct answer when the role-playing persona is aligned at the instance level, which is inferred as *"Physics Engineer"*. In both cases, although a coarsely aligned persona (e.g. *"Math*

Method	Dataset	Persona Solver (w/ Persona)		
			Wrong	Correct
Neutral Solver (w/o Persona)	AQuA	Wrong	33.07%	15.75%
		Correct	**13.78%**	37.40%
	Coin Flip		Wrong	Correct
		Wrong	4.60%	4.00%
		Correct	**18.00%**	73.40%

Table 1: **Confusion matrix between Neutral Solver (w/o Persona) and its Persona Solver (w/ Persona) on AQuA and Coin Flip dataset.** We calculate the model's correctness and present the result in a confusion matrix form. Neutral Solver and Persona Solver mean an LLM without persona and an LLM with persona, respectively. Appendix D includes more analysis for other datasets.

teacher") seems to be effective in solving the given mathematical dataset, it ultimately produces the wrong answers. This highlights the importance of considering whether the assigned persona is also aligned with the given instance. Furthermore, it also demonstrates that, in some cases, the correct answer can be achieved without using a persona.

For deeper insights into estimating the impact of LLM without a persona, we further experimentally compare the LLM's correctness based on whether a persona is assigned for two reasoning datasets. Table 1 shows the confusion matrices of an empirical result to run an LLM with persona and without persona on the AQuA and Coin Flip datasets. The AQuA dataset results show that 15.75% of the questions become correct when using an LLM with persona compared to without it. On the other hand, 13.78% of the questions are incorrectly answered when using an LLM with a persona rather than without it. This phenomenon could also be observed from the result of the Coin Flip dataset, stating that 18% of the questions are wrong when using persona and correct without it. It shows that assigning a persona to an LLM sometimes degrades its reasoning ability. Thus, instead of applying a role-playing prompt, it is crucial to distinguish whether a role-playing persona should be used based on the given query to improve the LLM's performance.

To address this limitation, we propose a novel framework called **Jekyll & Hyde** that automatically generates an instance-aligned role-playing prompt for the given query, thereafter ensemble the solutions of role-playing and neutral (non-persona) prompts to maximize the reasoning ability of the model. We execute an LLM with role-playing and neutral prompts to obtain each solution and then utilize an LLM evaluator to judge which solution is better. We demonstrate our method by comparing the LLM with and without a persona, showing that our method outperforms the single role-playing and neutral LLM across three widely used models: GPT-4, GPT-3.5-turbo, and Llama 3-8B model. For instance, Jekyll & Hyde outperforms the baselines by an average accuracy of 9.98% in 12 datasets when using GPT-4 as a backbone model. In addition, we demonstrated that using an instance-aligned persona is more effective than a coarsely aligned persona, and in some cases, better reasoning performance is achieved without a persona. To the best of our knowledge, this work systematically investigates the side effects of coarsely aligned personas on LLMs in reasoning tasks and proposes a novel framework to address this issue.

2 Related Works

2.1 Role-playing Abilities of LLMs

Large language models have demonstrated significant capabilities in personating various roles, which highlights the power of LLMs' role-playing capabilities. Based on this consensus, several studies have investigated the positive effect of role assignment on improving the performance of LLMs. Kong et al. (2024) have revealed the effect of using role-playing prompts in an LLM by handcrafting a specific prompt form for 12 different reasoning datasets and discovered that assigning a proper role to the LLM enhances its reasoning ability. Salewski et al. (2024) have shown the impact of role assignment on the LLM when using expertise impersonation related to the given dataset, by only including the occupation for the persona (e.g., *high school computer science expert*). Other studies have systematically benchmarked LLM role-playing abilities (Wang et al., 2023b), analyzed the cognitive biases induced by personas in Theory-of-Mind tasks (Yeo et al., 2025), and even explored representation-level control using "role vectors" (Potertì et al., 2025). Collectively, these works highlight the potential of persona-based prompting, establishing it as a promising yet nuanced approach to improving LLM reasoning.

2.2 Analysis on Role-playing Prompts

Role-playing has been widely adopted to improve LLMs' reasoning and problem-solving performance by conditioning on explicit personas; yet, recent studies reveal notable drawbacks to persona

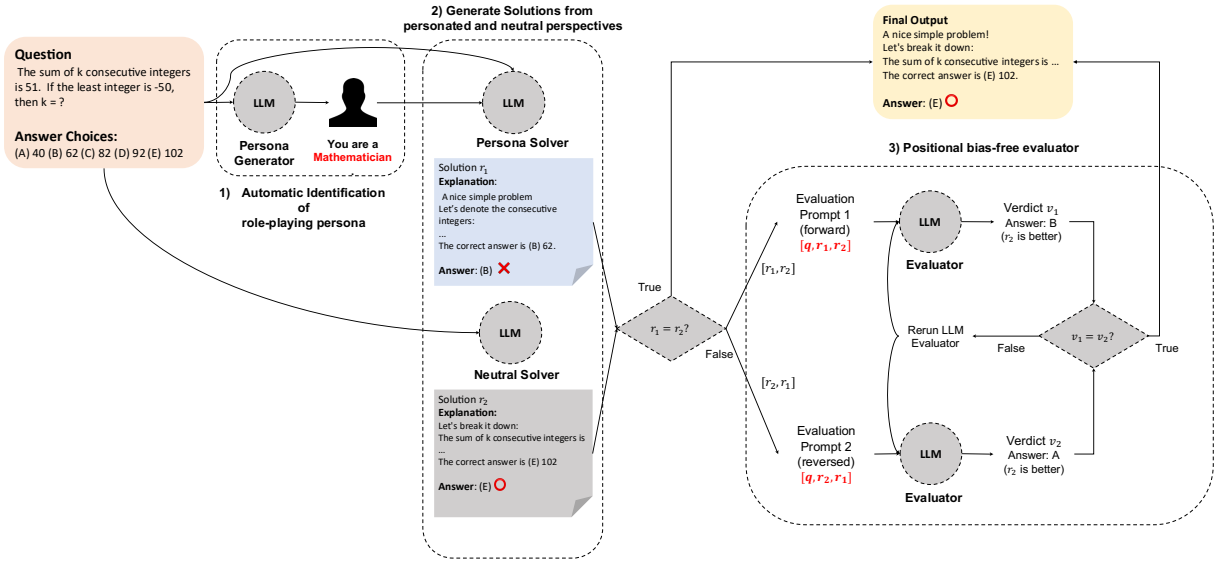

Figure 2: **The Architecture of Jekyll & Hyde.** Jekyll & Hyde utilizes a persona-assigned LLM (**Persona Solver**) and an LLM without a persona (**Neutral Solver**), which provides a dual perspective towards the given question. This structure improves the model to gain potentially high performance. After executing both LLMs, a robust Evaluator, designed to mitigate positional bias, selects a better solution between them.

assignment. Prior works show that adding socio-demographic traits (e.g., disability, race, sexual orientation) often induces bias or toxicity, degrading reasoning accuracy (Gupta et al., 2023; Deshpande et al., 2023). Benchmarks such as *BiasLens* demonstrate that persona conditioning systematically amplifies bias even when baseline prompts remain neutral (Li et al., 2024). Mitigation efforts include enforcing consistency through Persona-Aware Contrastive Learning (Ji et al., 2025) and replacing demographic cues with belief-seeded personas (Do et al., 2025), yet our work is the first to systematically analyze how coarsely aligned personas can still impair reasoning, underscoring the need for instance-specific persona alignment.

3 Methods

In this section, we demonstrate the process of Jekyll & Hyde. Specifically, Jekyll & Hyde consists of three different LLM modules: **Persona generator**, **Solver**, and **Evaluator**. Jekyll & Hyde's pipeline is as follows: First, the Persona generator generates an instance-specific persona aligned to the given query. Then, two different LLM solvers (i.e., Persona Solver and Neutral Solver) are executed simultaneously to generate solutions for the given question. Finally, the Evaluator compares two solutions and derives the final prediction. Figure 2 describes the overall framework of Jekyll & Hyde.

3.1 Automatic Identification of Persona

The common practice of role-playing prompting prepends a manually assigned persona role (e.g., Mathematician) into the prompt that contains the question. While these conventional role-playing methods are known to work well, the persona role that the user considers suitable for solving the problem may lead to performance fluctuation, as it only focuses on aligning with the dataset rather than the given specific instance. To address these drawbacks, we use an LLM (**Persona generator**) to guess a role that aligns with the given query using an instruction-following prompt. This prompt guides the LLM to automatically generate a persona that could solve the given question, generating different adequate personas that align well with each instance of the dataset. Appendix A details the instructions-following prompt.

3.2 Generating Personated and Neutral Perspective Solutions

After identifying an instance-aligned persona, it is formatted as a role-playing prompt and inserted into the input query for an LLM. While utilizing role-playing prompts typically improves the performance of an LLM, using a persona prompt can be a double-edged sword for two reasons. 1) persona assignment may not always align closely with the corresponding data instances. 2) An LLM without a persona may sometimes outperform one with a 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 means directly inserting 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.e., Persona and Neutral Solvers) and ideally choose the correct answer between two responses, we can 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. When implementing the Neutral Solver, we follow the identical implementation of (Kong et al., 2024). To estimate the impact of role-playing prompting, we utilized three types of prompt design and chose the most optimal format. 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. The detailed format of the prompt can be found in Appendix F.

3.3 Aggregating Solutions of Two Solvers

Two solutions generated from Neutral Solver and Persona Solver are inserted into the evaluation prompt. Specifically, two solutions are formatted to the evaluation prompt, establishing an order between the solutions. The format of the evaluation prompt can be found in Appendix A. Formally, given a question q and two solutions (r_n, r_p), we depict the process of the Evaluator as the following:

$$v_{n,p} = \underset{v}{\text{argmax}} \mathcal{P}(v|[\iota; q; r_n; r_p]) \quad (1)$$

where $v \in \{$*"A"*, *"B"*$\}$ is a verdict text and $\mathcal{P}$ is the LLM Evaluator. ι is an instruction for evaluation, and q is a given question. r_n and r_p indicate the solution of the Neutral and Persona Solver, respectively. $v_{n,p}$ means the verdict generated by the Evaluator based on the evaluation prompt, where ι, q, r_n, and r_p construct the evaluation prompt, as $P_{eval} = [\iota; q; r_n; r_p]$. The verdict $v_{n,p}$ takes one of two values,(*"A"* or *"B"*), indicating whether the first or second solution is judged superior, respectively. Note that $v_{n,p}$ is obtained by inserting two responses in the order of r_n and r_p; thus, we can also get $v_{p,n}$ by reversing the order of two solutions in the evaluation prompt, as $P_{eval} = [\iota; q; r_p; r_n]$.

3.4 Robust Evaluation via Consistency Verification

As introduced in Section 3.3, the Evaluator returns the verdict 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., 2023a). Therefore, we run the Evaluator model shown in Equation 1 twice by inserting the solutions into the evaluation prompt and reversing the order of the solutions to mitigate the position bias. Hence, we yield two verdicts, namely $v_{n,p}$ and $v_{p,n}$. When evaluations are executed to generate their verdict, we count the number of trials t until it reaches the maximum trial k, defined as a hyper-parameter. Then, the framework compares two verdicts, whether equal or not. The process ends when these two verdicts are identical, as in the following formula.

$$v_{final} = \begin{cases} v_{n,p} & \text{if } v_{n,p} = v_{p,n} \text{ and } t < k \\ \textit{"Can't answer"} & \text{if } t \geq k \end{cases} \quad (2)$$

where v_{final} is the final verdict obtained by considering the consistency of two verdicts. 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 the final output since it is risky to narrow to one solution in this case.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct our experiments across twelve datasets 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 et al., 2019) and StrategyQA (Geva et al., 2021) (3) **Symbolic reasoning**, including Coin Flip and Last Letter (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 LLMs released from OpenAI, GPT-4 (gpt-4-0613) and GPT-3.5-turbo (gpt-3.5-turbo-0125) (OpenAI, 2023), and

Models	Method	Arithmetic						
		Multiarith	GSM8K	AddSub	AQuA	SingleEq	SVAMP	**Average**
GPT-4	Base	**98.44**	92.97	97.13	68.24	98.56	91.00	91.06
	Fixed Persona	97.83	94.39	96.96	73.23	97.83	91.2	91.90
	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	Base	95.72	81.40	90.97	62.60	97.83	80.17	84.78
	Fixed Persona	**97.67**	81.35	91.64	64.57	96.85	84.3	86.06
	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**
Llama 3-8B	Base	**98.56**	78.59	87.76	47.38	94.23	82.30	81.47
	Fixed Persona	97.00	81.05	86.33	50.79	92.13	84.30	81.46
	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 the accuracy on six arithmetic datasets, evaluated with a Neutral solver (Base), LLM with dataset-aligned persona (Fixed Persona), Persona solver (Persona), and Jekyll & Hyde. Bold values mean the best performance among the four methods. We run each model three times and average their performance. Fixed personas are provided in Appendix E.

one open source model, Llama 3 (Llama3-8b-Instruct) (Grattafiori et al., 2024). These models are used as the backbone model of our framework.

Implementation Details. To evaluate Jekyll & Hyde, we test four cases for each dataset: **Base**, **Fixed Persona**, **Persona**, and **Jekyll & Hyde**. **(1) Base** utilizes only the Neutral solver where a persona is not assigned to LLMs. **(2) Fixed Persona** represents an LLM that uses a coarsely aligned persona for each dataset, which is known as the common practice of role-play prompting. **(3) Persona** uses only the Persona solver, an LLM assigned with an instance-specific persona generated from the Persona Generator. **(4) Jekyll & Hyde** is our proposed framework. We evaluate the model's performance by computing the accuracy using the provided labels for each dataset. When using the LLM evaluator in Jekyll & Hyde, the hyperparameters are set as follows: the max attempt k to 5 and temperature τ to 0.7. For using Persona Generator for Persona and Jekyll & Hyde, we set the temperature of LLM to 0.7. Details for determining the hyperparameters are shown in Appendix J. Moreover, the coarsely aligned personas used for the Fixed Persona method are in Appendix E.

4.2 Results and Analysis

Main Result. Table 2 shows the performance of different methods on arithmetic datasets, while Table 3 reports results of the remaining datasets, all evaluated in terms of accuracy. Across these evaluations, Jekyll & Hyde consistently enhances

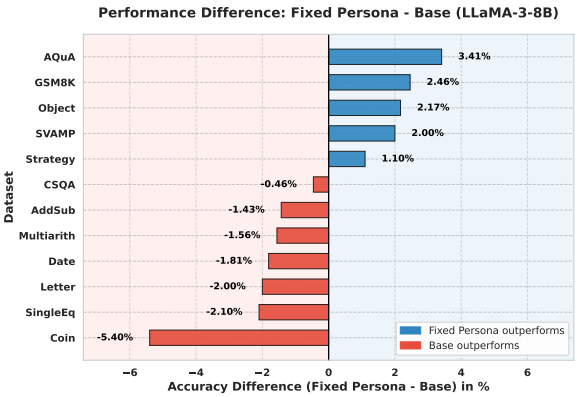

Figure 3: **Performance Gap across datasets.** The performance differences vary across tasks, with neither approach consistently outperforming the other, indicating no clear dominance of either Fixed Persona or Base.

model performance, outperforming the baselines. Notably, Jekyll & Hyde achieves superior results across the majority of datasets, regardless of the model type, demonstrating the robustness of the approach. Furthermore, the results highlight that employing instance-aligned personas (**Persona**) yields a higher average accuracy than using coarsely aligned personas (**Fixed Persona**), confirming that instance-aligned personas can better guide reasoning than dataset-level role assignments. These findings confirm the effectiveness of the ensemble approach and highlight the benefits of instance-specific personas.

Fixed Persona vs. Base: No Clear Winner To investigate whether fixed persona prompting provides a consistent advantage over the base model,

Models	Method	Common Sense		Symbolic Reasoning		Other Tasks		
		CSQA	Strategy	Letter	Coin	Date	Object	**Average**
GPT-4	Base	79.91	76.42	19.80	66.93	79.22	45.96	61.37
	Fixed Persona	**81.82**	74.45	92.60	**85.40**	78.32	45.47	76.34
	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	Base	77.31	68.75	18.67	47.53	67.84	34.67	52.46
	Fixed Persona	**79.77**	69.52	45.2	51.6	**79.95**	33.87	59.98
	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**
Llama 3-8B	Base	74.50	69.21	86.40	95.80	77.42	44.76	74.68
	Fixed Persona	74.04	70.31	84.40	90.40	75.61	46.93	73.62
	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.

Model	Datasets	Methods	Accuracy (↑)	Average LLM runs (↓)
GPT-4	AQuA	Base + voting	70.87	4
		Persona + voting	73.23	6
		Jekyll & Hyde	**76.90**	3.81
	Object	Base + voting	46.00	5
		Persona + voting	59.20	6
		Jekyll & Hyde	**61.69**	4.14
GPT-3.5-turbo	AQuA	Base + voting	66.14	5
		Persona + voting	66.53	6
		Jekyll & Hyde	**67.98**	**4.35**
	Object	Base + voting	34	5
		Persona + voting	33.73	6
		Jekyll & Hyde	**42.22**	**4.30**

Table 4: **Comparison of performance between Jekyll & Hyde, Base with self-consistency, and Persona with self-consistency** Jekyll & Hyde outperforms other methods when running the same amount of LLM executions, showing that running the LLM multiple times does not necessarily improve its reasoning ability.

we conducted an empirical comparison of their performance across a diverse set of benchmark datasets. Figure 3 reports the accuracy differences between the **Fixed Persona** and **Base**. The results reveal substantial variability across tasks, with no single approach demonstrating clear or consistent superiority. While fixed personas occasionally yield performance improvements, they also frequently lead to degradation, highlighting that the effectiveness of coarsely aligned personas is highly task-dependent. These findings suggest that adopting a coarsely aligned persona does not guarantee uniform gains and, in certain scenarios, may even hinder performance.

Comparison with Self-Consistency Unlike single-perspective LLMs, Jekyll & Hyde varies the number of execution trials per instance, which might give the false impression of better performance due to more trials. To clarify, we conducted an experiment with equal or larger execution trials for single-perspective LLMs, verifying that simply increasing the LLM execution does not help improve reasoning capability. Specifically, we run two cases, namely Base and Persona, in a setting of self-consistency (Wang et al., 2022), which executes the LLM multiple times and selects the most frequent answer. Hence, **Base** and **Persona** can be executed in the same amount as the number of Jekyll & Hyde runs. For the experimental setting, we utilize GPT-3.5-turbo and GPT-4 as our models, along with two reasoning datasets: the AQuA and Object Tracking datasets. For single-perspective LLMs that utilize self-consistency, we refer to the methods as *Base + voting* and *Persona + voting*, respectively. The specific settings for *Base + voting* and *Persona + voting* can be found in Appendix H. As shown in table 4, the result reveals that Jekyll & Hyde outperforms single LLM with self-consistency by gaining better performance and lower LLM execution trials. In addition, it shows that running the LLM multiple times does not necessarily improve its reasoning ability, highlighting the effectiveness of Jekyll & Hyde.

Effectiveness of LLM-Generated Personas In Section 3.1, we evaluate the effectiveness of automatically generated personas in Jekyll & Hyde

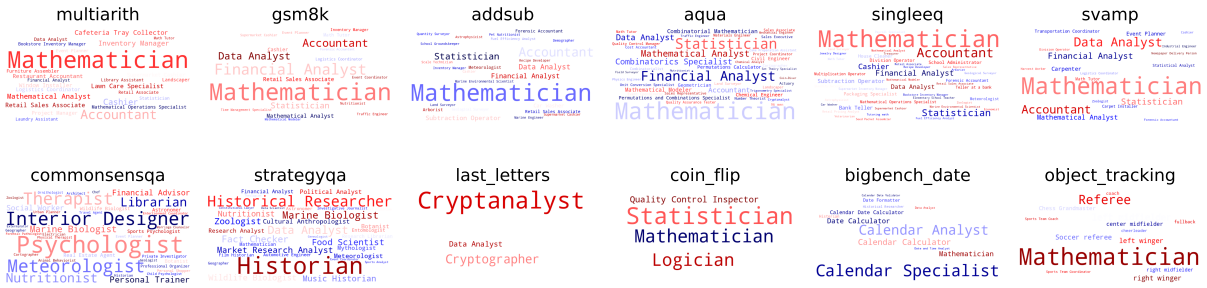

Figure 4: **Word clouds of LLM-generated personas (Llama-3-8B) across twelve reasoning datasets.** showing diverse and task-relevant roles that highlight the model's adaptive reasoning capability.

Model	Datasets	Methods	Average Accuracy (↑)	Standard Deviation (↓)
Llama 3-8B	AQuA	Fixed Persona	51.71	6.11
		LLM generated persona	**52.23**	**2.08**
	Object	Fixed Persona	44.31	8.02
		LLM generated persona	**47.60**	**3.06**

Table 5: **Standard deviation of Fixed Persona LLM and LLM-generated persona LLM** We ran each dataset three times and found that LLM-generated personas yield lower standard deviations in two datasets, indicating more stable performance.

compared to coarsely aligned **Fixed Personas**. Using Llama-3-8B-Instruct as the backbone, we conduct experiments on the AQuA and Object Tracking datasets, where personas are generated by sampling from the LLM's output probability distribution. For comparison, we use manually crafted Fixed Personas (*Math Teacher*, *Mathematician*, *Math Tutor* for AQuA; *Observer*, *Recorder*, *Logical Reasoner* for Object Tracking). As shown in Table 5, LLM-generated personas outperform Fixed Personas in both accuracy and stability, exhibiting smaller variance across runs. Furthermore, we visualize the LLM-generated personas for each dataset in Llama-3-8B-Instruct as word clouds in Figure 4. The diverse and instance-aligned personas (e.g. *Financial Analyst* in AQuA) highlight the Persona Generator's ability to adaptively choose roles suitable for each problem, explaining its performance gains and enhanced robustness.

Position Bias Mitigation in Jekyll & Hyde The Evaluator should not suffer from position bias when choosing the correct solution between the two. For further analysis of the framework's mitigation process, we conduct an experiment that estimates the performance of Jekyll & Hyde in six different datasets, comparing with the two existing position bias mitigation methods: (1) Portia (Li et al., 2023) and (2) MEC+BPC (Wang et al., 2023a). For a more thorough investigation, we also consider the ideal case, in which the Evaluator always selects the gold answer from the Neutral and Persona Solvers' answers, referred to as the Oracle Evaluator, as an upper bound. Experimental results for SingleEq and Coin Flip datasets can be found in table 6. The results show that Jekyll & Hyde's evaluator outperforms other methods across most datasets and achieves performance close to the Oracle evaluator, confirming its reliability in selecting the correct response between perspectives. The details for the implementation of Portia and MEC+BPC are shown in Appendix G. Full results can be exhibited in Appendix I.

Models	Method	SingleEq	Coin
GPT-4	Oracle Evaluator	99.41	88.80
	Portia	98.82	74.40
	MEC+BPC	98.43	74.00
	Jekyll & Hyde†	98.43	78.20
	Jekyll & Hyde	**98.95**	**80.27**
GPT-3.5-turbo	Oracle Evaluator	99.21	66.73
	Portia	**98.23**	57.80
	MEC+BPC	97.64	57.60
	Jekyll & Hyde†	97.83	56.60
	Jekyll & Hyde	98.03	**64.00**
Llama 3-8B	Oracle Evaluator	96.06	99.00
	Portia	93.31	96.40
	MEC+BPC	91.73	95.40
	Jekyll & Hyde†	94.29	97.00
	Jekyll & Hyde	**94.62**	**98.67**

Table 6: **Mitigating positional bias.** Jekyll & Hyde's LLM Evaluator outperforms prior methods, narrowing the gap to Oracle-level performance. Jekyll & Hyde† indicates the Evaluator without positional bias mitigation.

4.3 Qualitative Analysis

Case Study. We qualitatively analyze examples where Jekyll & Hyde ultimately provides the correct answers on three datasets: AQuA, StrategyQA,

Dataset	Question (q)	Neutral Solver(r_1)	Persona Solver(r_2)	Final prediction
AQuA	A starts a business with Rs.40,000. After 2 months, B joined him with Rs.60,000. C joined them after some more time with Rs.120,000. At the end of the year, out of a total profit of Rs.375,000, C gets Rs.150,000 as his share. How many months after B joined the business, did C join? **Answer Choices:** (A) 2 months (B) 4 months (C) 23 months (D) 24 months (E) 84 months	Let x be the number of months after B joined the business that C joined. A's investment for the entire year = 40,000 x 12 = 480,000 $\cdots$ So, the correct answer is: (A) 2 months. Therefore, among A through E, the answer is **A** ✗ **persona:** None	Let's break down the problem step by step: 1. A starts the business with Rs.40,000. $\cdots$. The answer is (B) 4 months. Therefore, among A through E, the answer is **B** ✓ **persona:** Accountant	B
Strategy	Would JPEG be a good format for saving an image of Da Vinci's Vitruvian Man? **Answer Choices:** yes, no	Yes, JPEG would be a good format for saving an image of Da Vinci's Vitruvian Man. JPEG is a commonly $\cdots$ Therefore, the answer (Yes or No) is **yes** ✗ **persona:** None	JPEG is a commonly used format for saving images due to its compression capabilities, which can reduce file size while maintaining decent image quality. However, $\cdots$ Therefore, the answer (Yes or No) is **no** ✓ **persona:** Digital Archivist	no
Object	Alice, Bob, and Claire are holding a white elephant gift exchange. At the start of the event, $\cdots$ At the end of the event, Alice has the Which choice is true? **Answer Choices:** (A) orange ball. (B) purple present. (C) blue present.	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. $\cdots$. 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	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.$\cdots$. So, the correct answer is: (A) orange ball. Therefore, among A through C, the answer is **A** ✗ **persona:** Gift Exchange Analyst	B

Table 7: **Qualitative analysis.** We conduct a qualitative analysis of Jekyll & Hyde on three datasets. Jekyll & Hyde's Persona Solver attains an instance-specific persona based on the Persona Generator (e.g., Accountant, Digital Archivist, Gift Exchange Analyst). Additionally, examples demonstrate that the Evaluator is functioning properly by selecting the correct answer among the various solutions provided by each Solver.

and Object Tracking. Table 7 shows the outputs for each process. In the AQuA and StrategyQA examples, the Persona Generator creates an instance-specific persona aligned with the given question (e.g., *Accountant*, *Digital Archivist*), leading the Persona Solver to produce the correct answer. In contrast, the Object Tracking example demonstrates a case where the Persona Solver initially provides an incorrect answer, but Jekyll & Hyde's evaluator correctly selects the Neutral Solver's output. This process enables the framework to choose better results between Neutral and Persona solvers, thereby improving overall performance.

Estimating *Can't Answer* Instances. We further measure the frequency of *Can't Answer* predictions in Jekyll & Hyde, as shown in Table 8. The results reveal that such cases are extremely rare across all models, demonstrating that our methodology can reliably select the better answer between the Neutral and Persona solvers in nearly all scenarios.

Dataset	# of *Can't Answer* instances (% of Instances)		
	GPT-4	GPT-3.5-turbo	Llama 3-8B
Multiarith	0 (0.000 %)	1 (0.167 %)	0 (0.000 %)
GSM8K	4 (0.303 %)	10 (0.758 %)	5 (0.379 %)
AddSub	0 (0.000 %)	1 (0.253 %)	1 (0.253 %)
AQuA	4 (1.575 %)	8 (3.150 %)	6 (2.362 %)
SingleEq	0 (0.000 %)	2 (0.394 %)	1 (0.197 %)
SVAMP	2 (0.200 %)	4 (0.400 %)	7 (0.700 %)
CSQA	23 (1.884 %)	3 (0.246 %)	9 (0.737 %)
Strategy	22 (2.949 %)	18 (2.413 %)	7 (0.938 %)
Letter	0 (0.000 %)	7 (1.400 %)	6 (1.200 %)
Coin	4 (0.800 %)	0 (0.000 %)	1 (0.200 %)
Date	0 (0.000 %)	1 (0.271 %)	3 (0.813 %)
Object	7 (0.933 %)	8 (1.067 %)	22 (2.933 %)

Table 8: **Number and proportion of Can't Answer predictions across datasets.** The results show that Can't Answer cases occur only rarely, indicating that Jekyll & Hyde almost always selects between the Base and Persona solvers rather than abstaining

5 Conclusion

In this paper, we propose Jekyll & Hyde, a novel framework that solves the reasoning task by ensembling instance-aligned personated and neutral perspectives. Evaluations across twelve renowned reasoning benchmark datasets show that our framework surpasses both cases when the persona is assigned and when it is not on most datasets. Our findings revealed that a coarsely aligned persona does not consistently improve the model performance; instead, effective performance improvement requires personas to be aligned with individual instances. Additionally, we observed the potential benefits of combining different viewpoints of LLMs, contributing to the enhancement of the model's performance. Overall, this work sets the initial stage for further investigation in combining solutions from various perspectives within the LLM community, a promising research direction for improving reasoning abilities.

Limitations

While Jekyll & Hyde introduces additional computation by executing both persona-assigned and neutral solvers, it remains comparable in efficiency to existing ensemble-based methods. Moreover, users can balance performance and efficiency by setting a small maximum number of evaluator attempts (e.g., two), which still provides consistent gains over single-perspective LLMs. As with any dual-solver framework, Jekyll & Hyde cannot recover cases where both solvers fail; however, our results demonstrate that such instances are rare in the reasoning tasks we study. Since this work focuses primarily on conventional reasoning benchmarks, further exploration on broader task types and efficiency–accuracy trade-offs represents promising future work.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1236–1270.

Xuan Long Do, Kenji Kawaguchi, Min-Yen Kan, and Nancy Chen. 2025. Aligning large language models with human opinions through persona selection and value–belief–norm reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2526–2547.

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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shashank Gupta, Vaishnavi Shrivastava, Ameet Deshpande, Ashwin Kalyan, Peter Clark, Ashish Sabharwal, and Tushar Khot. 2023. Bias runs deep: Implicit reasoning biases in persona-assigned llms. *arXiv preprint arXiv:2311.04892*.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533, Doha, Qatar. Association for Computational Linguistics.

Ke Ji, Yixin Lian, Linxu Li, Jingsheng Gao, Weiyuan Li, and Bin Dai. 2025. Enhancing persona consistency for llms' role-playing using persona-aware contrastive learning. *arXiv preprint arXiv:2503.17662*.

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.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113.

Xinyue Li, Zhenpeng Chen, Jie M Zhang, Yiling Lou, Tianlin Li, Weisong Sun, Yang Liu, and Xuanzhe Liu. 2024. Benchmarking bias in large language models during role-playing. *arXiv preprint arXiv:2411.00585*.

Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2023. Split and merge: Aligning position biases in large language model based evaluators. *arXiv preprint arXiv:2310.01432*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

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

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Daniele Potertì, Andrea Seveso, and Fabio Mercorio. 2025. Designing role vectors to improve llm inference behaviour. *arXiv preprint arXiv:2502.12055*.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.

Leonard Salewski, Stephan Alaniz, Isabel Rio-Torto, Eric Schulz, and Zeynep Akata. 2024. In-context impersonation reveals large language models' strengths and biases. *Advances in Neural Information Processing Systems*, 36.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, 623(7987):493–498.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

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

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, and 1 others. 2023b. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. *arXiv preprint arXiv:2310.00746*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Gerard Yeo, Fiona Tan An Ting, Kokil Jaidka, Shaz Furniturewala, Wu Fanyou, Weijie Xu, Vinija Jain, Aman Chadha, Yang Liu, and See Kiong Ng. 2025. Phantom: Persona-based prompting has an effect on theory-of-mind reasoning in large language models. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 2124–2142.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

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 9, 10 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 9: 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 specific 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 progress is shown in figure 5. The answer trigger sentences for various datasets are depicted in Table 11.

C Dataset Details

In this section, we briefly introduce twelve datasets spanning four categories below. Specific details are shown in Table 12

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

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.

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

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 11: Answer trigger sentences for various answer formats.

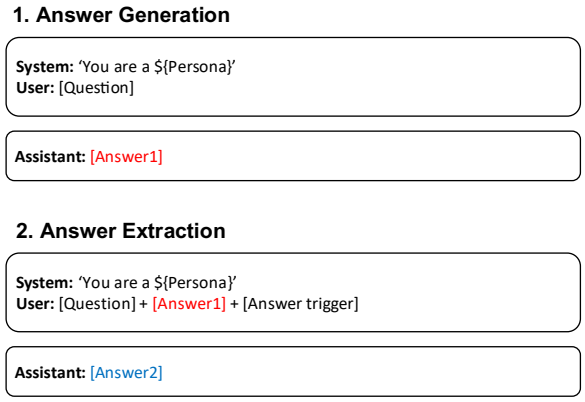

Figure 5: an entire process of how Solver works

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 asks for a certain object's final location.

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 13 exhibits the confusion matrix for each dataset respectively.

E Handcrafted persona for each dataset

To investigate the impact of utilizing handcrafted personas aligned to each dataset, we chose 6 different persona occupations for our experiment. Table 14 shows the handcrafted persona aligned to the given dataset.

F Impact of prompt design

This section introduces the default prompt design for persona LLM. While there are a lot of variations in prompts, we are the first to compare the impact of prompt designs for LLM-generated role-playing prompts according to the best of our knowledge. Hence, we conducted three different prompts and computed the performance of each prompt with GPT-3.5-turbo using the Aqua dataset. Table 15 shows different forms of prompts and their performance. The result reveals that using a single persona acquires the optimal performance in persona LLM, thereby outperforming other settings in Jekyll & Hyde.

G Implementation details for Portia and MEC+BPC

In section 4.2, 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., 2023a) 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, the final scores of each response are aggregated and computed by averaging scores for the two sequences of solutions, respectively. We

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 12: 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 13: Confusion matrix between Neutral Solver (w/o Persona) and its Persona Solver (w/ Persona) on StrategyQA dataset.

implemented MEC+BPC by preparing two evaluation prompts for the two sequences. Then, we ran the model and computed the score for each response. The model is run three times for robust answer generation, and the average of the scores is computed.

H Settings for the number of self-consistency of the base, persona LLMs

In table 4, we executed Jekyll & Hyde for each model and calculated the average number of LLM executions per instance. In order to compare the performance of the Base and Persona LLMs under the condition of executing the LLM the same number as Jekyll & Hyde, we ensured self-consistency for both LLMs. Specifically, we executed Jekyll & Hyde for both datasets and computed the average number of LLM executions for a single instance of each dataset. Afterward, the number of self-consistency for Base + voting is determined as the ceiling of the average executions for Jekyll & Hyde. In the case of Persona + voting, given an average number of LLM executions n, we determined the number of self-consistency k following the formula below:

$$k = \begin{cases} 2 \cdot \lfloor \frac{n}{2} \rfloor & \text{if } \lceil n \rceil \div 2 = 0 \\ 2 \cdot (\lfloor \frac{n}{2} \rfloor + 1) & \text{if } \lceil n \rceil \div 2 = 1 \text{ or } n = 4 \end{cases} \quad (3)$$

since Persona requires two times inference (Persona Generator + Persona Solver), we incremented the self-consistency iterations if the number is odd. When n is 4, it means that Persona yields two outputs, leading it impossible to find the most frequent answer if two outputs are different.

I Full experiment for comparing methods of positional bias mitigation

Table 16 shows the performance of three different position bias mitigation methods for six datasets. Portia is implemented by dividing the given solution into three chunks, each with the same number of tokens. MEC+BPC is implemented by generating scores ranging from 1 to 10 three times for each solution. The final solution is determined by comparing the average score of each solution. The result exhibits that utilizing the Jekyll & Hyde evaluator achieves optimal performance across most datasets.

Tasks	Handcrafted persona
Arithmetic, GSM8K, AddSub, SingleEq, SVAMP	*"Math teacher"*
CSQA, Strategy	*"Commonsense quiz contest contestant"*
Letter	*"Software engineer"*
Coin	*"Coin flip analyst"*
Date	*"Date calculator"*
Object	*"Recorder"*

Table 14: Handcrafted personas aligned to each dataset

form	prompt	AQUA	Accuracy (↑)
persona	You are a [$persona$]	Persona	**65.75**
		Jekyll & Hyde	**69.68**
persona + task description	You are a [$persona$]. Your task is to solve the given math question and come up with a correct answer.	Persona	62.99
		Jekyll & Hyde	68.11
task description	Your task is to solve the given math question and come up with a correct answer.	Persona	65.35
		Jekyll & Hyde	68.90

Table 15: **Performance of different prompt designs** Among different types of prompt design, using only persona for the prompt obtains the highest performance in Persona LLM and Jekyll & Hyde, thereby setting it as a default prompt for our Persona LLM.

J 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. We utilize four datasets: MultiArith, SingleEq, AQuA, and Date Understanding. As shown in figure 6-(a), we compare the experimental results executed from Llama 3-8B as a backbone model and reveal the framework's performance increases as the number of attempts increases. Experimental results for other models can be found in Appendix K. Furthermore, we could identify that Jekyll & Hyde could outperform the single perspective LLM even when the max attempt k is 2. Since the enhancement of the framework is getting smaller as the number of the max attempts increases, we decided to use $k = 5$ as our default setting, which can balance the framework's performance and prevent costing the model excessively.

The Temperature of the Evaluator (τ). We further investigate the impact of the Evaluator's generation temperature by comparing the framework's performance. Specifically, we utilize the Llama 3-8b model and leverage 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 6-(b) shows that temperature $\tau = 0.7$ exhibits the optimal performance among others. Experimental results for other models can be found in Appendix K.

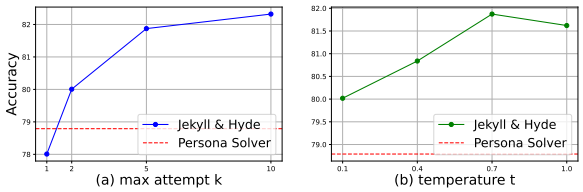

Figure 6: **Hyper-parameters Experiments.** Variation of averaged accuracy with a (a) various number of max attempt k and (b) temperature of the LLM τ used in LLM evaluator. X and Y axes correspond to each hyper-parameter setting and accuracy, respectively.

K Hyperparameter settings for GPT-4 and GPT-3.5-turbo

Figures 7, and 8 show the experimental result for the hyperparameter setting. As it shows, GPT-3.5-turbo shows that obtaining 0.7 as a temperature achieves the highest performance among other settings, and GPT-4 reveals that using 0.1 or 1.0

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	97.47	74.41	98.82	91.80	74.40	80.76
	MEC+BPC	97.22	74.41	98.43	91.20	74.00	79.95
	Jekyll & Hyde†	97.72	**78.35**	98.43	92.20	78.20	80.22
	Jekyll & Hyde	**97.72**	76.90	**98.95**	**92.03**	**80.27**	**82.38**
GPT-3.5-turbo	Oracle Evaluator	95.19	74.41	99.21	87.10	66.73	80.22
	Portia	91.14	62.60	**98.23**	81.80	57.80	72.63
	MEC+BPC	89.37	62.60	97.64	80.20	57.60	75.61
	Jekyll & Hyde†	92.15	62.60	97.83	82.50	56.60	72.63
	Jekyll & Hyde	**92.91**	**67.98**	98.03	**84.77**	**64.00**	**76.78**
Llama 3-8B	Oracle Evaluator	92.41	63.39	96.06	90.20	99.00	84.55
	Portia	88.35	51.97	93.31	86.10	96.40	78.86
	MEC+BPC	88.10	**55.91**	91.73	84.50	95.40	**81.03**
	Jekyll & Hyde†	90.38	51.18	94.29	86.10	97.00	79.95
	Jekyll & Hyde	**89.03**	54.07	**94.62**	**86.50**	**98.67**	79.04

Table 16: **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.

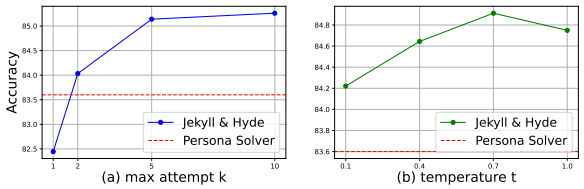

Figure 7: **Hyper-parameters Experiments for gpt-3.5-turbo** Variation of averaged accuracy with a (a) various number of max attempt k and (b) temperature of the LLM τ used in LLM evaluator. X and Y axes correspond to each hyper-parameter setting and accuracy, respectively.

as a temperature yields the highest performance. Since using 0.7 as a temperature does not lead the model to a significant performance decline, we determined 0.7 as our default temperature. Meanwhile, both GPT-3.5-turbo and GPT-4 present that the slope of the graph gradually flattens as the maximum number of attempts increases, leading to performance saturation at a certain performance. Hence we concluded to use 5 as our default max attempt setting considering the trade-off between the performance and the computational cost.

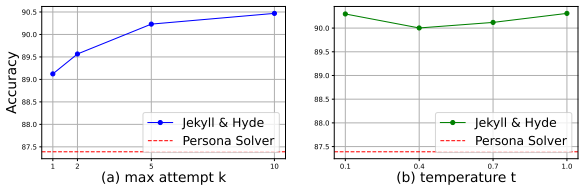

Figure 8: **Hyper-parameters Experiments for GPT-4** Variation of averaged accuracy with a (a) various number of max attempt k and (b) temperature of the LLM τ used in LLM evaluator. X and Y axes correspond to each hyper-parameter setting and accuracy, respectively.