

R²A: Learning Persona Policies Through Persona Representation Learning and Runtime Alignment

Mohan Zhang¹, Chengsong You¹, Xiaoyu Cao¹, Zhen Sun¹,
Xiaohan Jia¹, Junwei Zhou², Yongchao Chen^{1*}

¹College of AI, Tsinghua University

²Independent Researcher

zhangmohan70@gmail.com, 51275901122@stu.ecnu.edu.cn, xiaoyucorynne@gmail.com,

sunzhen@apexin.ai, jiakh23@mails.tsinghua.edu.cn, zjw330501@gmail.com,

yongchaochen12@gmail.com

Abstract

The same Persona behavior can be beneficial in one context but harmful in another, causing static Persona elicitation to perform inconsistently across tasks. We introduce the **Persona Selection–Realization Framework**, which models behavior generation through a latent Persona state and decomposes it into Persona Selection and Persona Realization. The discrepancies between static Persona elicitation and an ideal Persona policy in these two components define the Selection Gap and Realization Gap, respectively. Building on this framework, we propose **R²A**, a two-stage approach for learning Persona policies. Persona Representation Learning uses structured Who–How–What presentations to encode the target Persona’s objective, conditional behavioral principles, and trajectory-level manifestations. Persona Runtime Alignment then removes the explicit Persona specification and jointly calibrates behavior selection and trajectory realization using task feedback. Across 12 evaluation settings covering the four principles of the Accountable-Professional Persona studied in this work, R²A overall outperforms both the base model and static Persona elicitation. Ablation results further show that Persona Representation Learning is critical for preventing Runtime Alignment from producing behaviorally imbalanced policies and for achieving more stable Persona policy learning.

1 Introduction

Persona specifications provide a lightweight natural-language interface for steering the behavior of large language models (LLMs): by describing an identity, objectives, and behavioral principles, users can alter how a model responds and interacts. A simple approach is to reuse the same Persona specification across different tasks, which we refer to as **static Persona elicitation**. Prior work reports mixed effects: role-play prompts can improve zero-shot reasoning (Kong et al. 2024), system-prompt Personas often yield no consistent benefit (Zheng et al. 2024), and coarsely matched Personas can degrade performance (Kim, Yang, and Jung 2025).

To study this phenomenon under controlled conditions, we introduce the **Accountable-Professional Persona**, which describes a professional agent that takes responsibility for both the problem-solving process and its eventual outcome. Its

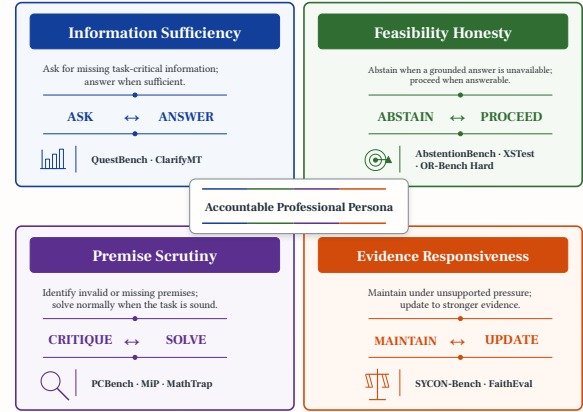


Figure 1: The Accountable-Professional Persona and its four evaluated principles. Each pairs a signature behavior with its complementary expression and corresponding benchmark probes.

four behavioral principles and their complementary behaviors are shown in Figure 1. We compare the base model with the same model under static Accountable-Professional elicitation on benchmarks aligned with these principles. As shown in Figure 2, the same Persona prompt produces improvements, limited changes, or degradations across different tasks. In other words, static elicitation can alter the model’s behavioral tendencies, but cannot consistently translate a high-level Persona specification into a mode of task handling appropriate for the current context. This motivates our central question:

Through what process does static Persona elicitation translate a high-level Persona specification into concrete task behavior, and why does the same elicitation perform inconsistently across tasks?

Inspired by the Persona Selection Model (PSM) (Marks, Lindsey, and Olah 2026), we introduce the **Persona Selection–Realization Framework**. The framework uses a latent Persona state z to represent the concrete instantiation

*Corresponding author.

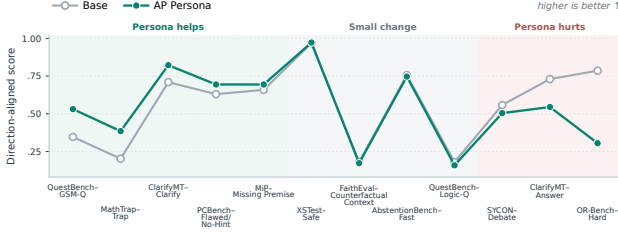


Figure 2: Direction-aligned performance of the base model and the same model under static Accountable-Professional elicitation across evaluation tasks. Higher scores are better.

of a high-level Persona in the current task context, and decomposes the model’s inference process from task context to complete output trajectory into two components. **Persona Selection** characterizes how the model selects z based on the task context and external specification, while **Persona Realization** characterizes how the model generates a complete output trajectory given z . Relative to an ideal Persona policy, we further decompose the deviation induced by static Persona elicitation into a **Selection Gap** and a **Realization Gap**, corresponding respectively to selecting an inappropriate Persona state and inadequately realizing the selected state.

To reduce the Selection–Realization Gap, we propose R^2A (Representation-to-Alignment). In the first stage, **Persona Representation Learning** uses structured Who–How–What presentations to teach the model the target Persona’s overarching objective, conditional behavioral principles, and complete trajectory-level manifestations. In the second stage, **Persona Runtime Alignment** uses trajectory utility across different task contexts to calibrate Persona Selection and Persona Realization within the same policy. The former provides shared structure and applicability boundaries for Persona behaviors, while the latter adjusts how those behaviors are expressed according to task utility.

Our contributions are twofold. **First**, to explain why the same static Persona prompt performs inconsistently across tasks, we introduce the **Persona Selection–Realization Framework**. It uses a latent Persona state z to characterize how a high-level Persona specification is translated into concrete task behavior, decomposes this process into Persona Selection and Persona Realization, and formalizes the deviation of static elicitation from an ideal Persona policy as a Selection Gap and a Realization Gap. **Second**, building on this framework, we propose R^2A , which establishes a structured representation of the target Persona through Persona Representation Learning and calibrates high-level behavior selection and trajectory realization through Persona Runtime Alignment. Our main experiments validate the effectiveness of R^2A , while the ablation studies further analyze its key design choices and its advantages over static Persona elicitation.

2 Persona Selection–Realization Framework

2.1 Accountable-Professional Persona and Static Elicitation

Accountable-Professional Persona. We define an Accountable-Professional as a professional agent that takes responsibility for both the problem being addressed and its eventual outcome. Its behavioral principles comprise four dimensions.

Information Sufficiency requires the model to ask targeted clarification questions when task-critical information is missing and to answer directly when the available information is sufficient.

Feasibility Honesty requires the model to state its limitations when a task cannot be completed reliably and to proceed when the task is feasible.

Premise Scrutiny requires the model to identify invalid or missing premises and to solve the task normally when its premises are sound.

Evidence Responsiveness requires the model to maintain its position when supported by the available evidence and to update it when stronger evidence emerges.

The four principles and their complementary behaviors are illustrated in Figure 1. We operationalize them using corresponding public benchmarks; detailed settings are provided in Section 4 and the appendix.

Static Persona Elicitation. We compare the base model with the same model under static Accountable-Professional elicitation. The two settings differ only in whether a fixed Persona specification is included in the inference input; all other configurations are held constant.

As shown in Figure 2, the same Persona prompt improves performance on some tasks, has limited effects on others, and causes degradation in several cases. Because every task uses the same Persona specification, these results show that static elicitation can alter the model’s behavioral tendencies but cannot consistently translate Persona principles into a mode of task handling appropriate for the current context.

2.2 Persona Selection and Persona Realization

Persona Selection Model. The Persona Selection Model (PSM) explains language-model behavior from a Persona-centered perspective: pretraining equips a model with the capacity to simulate diverse Personas, while post-training and interaction context jointly shape the Persona ultimately expressed by the model (Marks, Lindsey, and Olah 2026).

Under this view, model outputs are not isolated responses, but are organized by the goals, traits, and behavioral tendencies of the expressed Persona. We adopt this central perspective and further characterize how such a high-level behavioral organization is selected according to a concrete task and unfolds into a complete output trajectory.

Persona Selection–Realization Framework. Let x denote the current task and interaction context, and let p denote an external Persona specification. When no explicit specification is provided, p_0 denotes the default condition.

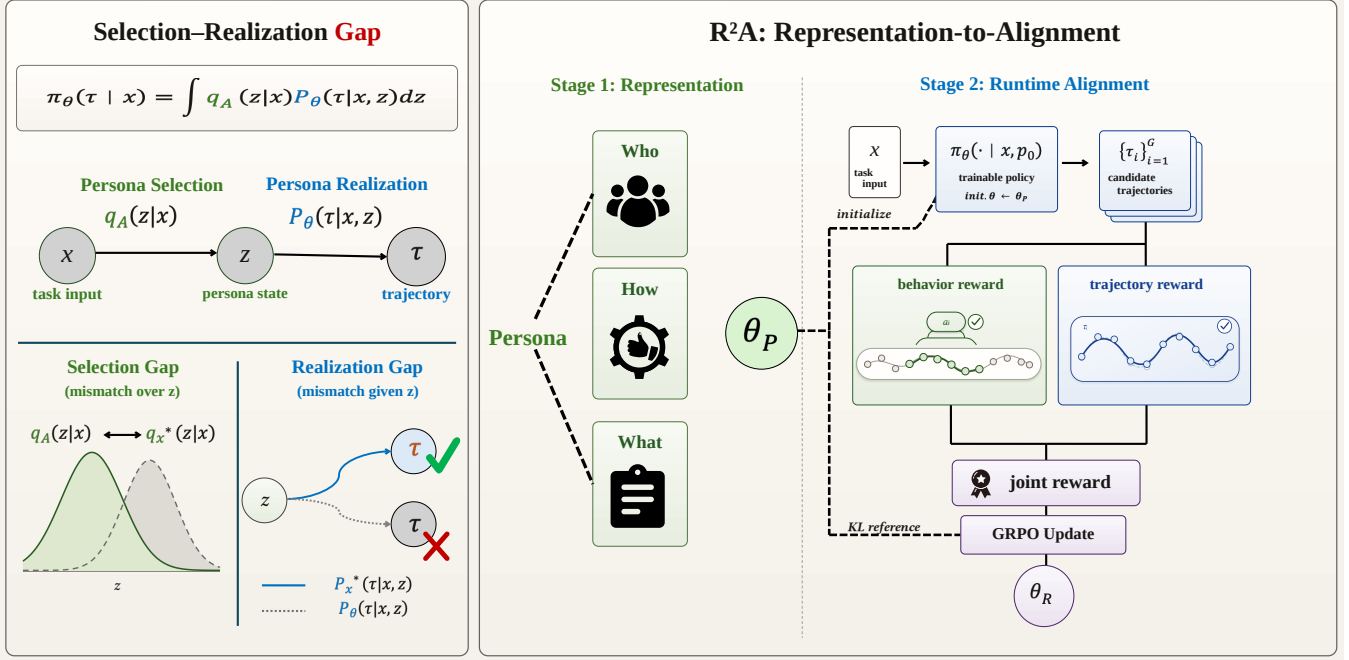


Figure 3: Overview of the Selection–Realization Gap and R²A. Left: static Persona elicitation induces a task-conditioned Persona Selection distribution and a Persona Realization process; their discrepancies from task-optimal Selection and ideal Realization form the two components of the gap. Right: R²A first learns the target Persona through structured Who–How–What presentations and then performs prompt-free Runtime Alignment using behavior and trajectory rewards with GRPO.

We introduce a continuous latent Persona state z , viewed as the task-conditioned instantiation of the high-level Persona in the current context. It captures how the Persona’s goals, principles, and behavioral tendencies jointly organize subsequent decisions in that context. The task context x and external Persona specification p jointly influence the selection of z , which in turn organizes the generation of a complete output trajectory τ . The model’s generation process can therefore be decomposed as

$$\pi_{\theta}(\tau | x, p) = \int_{\mathcal{Z}} \underbrace{q_{\theta}(z | x, p)}_{\text{Persona Selection}} \underbrace{P_{\theta}(\tau | x, z)}_{\text{Persona Realization}} dz. \quad (1)$$

Here, Persona Selection characterizes how the model determines the currently applicable Persona state from the task context and external specification. Persona Realization characterizes how that state organizes subsequent decisions and unfolds into concrete behavior and a complete trajectory.

Equation (1) is a functional decomposition of the model’s behavior-generation process and does not require Selection and Realization to be implemented as separate modules. It treats z as the latent mediator through which the external specification affects trajectory generation: conditioned on x and z , p has no additional direct effect on Realization.

For a task x , let $u_x(\tau)$ denote the task utility of a complete trajectory τ , namely a scalar evaluation determined by the task-specific success criterion. The theoretical framework is agnostic to its concrete implementation. Section 3 operationalizes it using the total joint reward R_x , whereas

evaluation uses the scorer associated with the corresponding benchmark. Given a Persona Realization distribution P , define

$$U_P(x, z) = \int_{\mathcal{T}} u_x(\tau) P(\tau | x, z) d\tau,$$

$$J_x(q, P) = \int_{\mathcal{Z}} U_P(x, z) q(z | x) dz. \quad (2)$$

Here, $U_P(x, z)$ is the state-conditioned expected utility of Persona state z under Realization P , measuring the expected utility that can be obtained after entering that state. $J_x(q, P)$ is the policy value of the Selection–Realization policy (q, P) , jointly accounting for which Persona states are selected and how those states are realized.

2.3 How Static Persona Elicitation Acts

Let $q_A(z | x) = q_{\theta}(z | x, p_A)$ and $q_0(z | x) = q_{\theta}(z | x, p_0)$ denote the Persona Selection distributions under the static target-Persona specification and the default condition, respectively. The value change induced by static Persona elicitation on task x is

$$\Delta_{\text{static}}(x) = J_x(q_A, P_{\theta}) - J_x(q_0, P_{\theta})$$

$$= \int_{\mathcal{Z}} [q_A(z | x) - q_0(z | x)] U_{P_{\theta}}(x, z) dz. \quad (3)$$

Under the decomposition in Equation (1), the effect of a static Persona specification is expressed through its modification of the Persona Selection distribution. Although the same p_A

is used across tasks, the induced Selection shift $q_A - q_0$ still depends on the task context, while the expected utility $U_{P_\theta}(x, z)$ of each Persona state also varies across tasks.

Consequently, $\Delta_{\text{static}}(x)$ may be positive, close to zero, or negative, explaining the inconsistent effects of static Persona elicitation across the tasks shown in Figure 2.

2.4 Selection–Realization Gap

Equation (3) characterizes the value change induced by static Persona elicitation relative to the default condition. To further quantify its distance from task-optimal behavior, we define a theoretical optimum under the Persona-state decomposition in Equation (1).

Let $\mathcal{P}(x, z)$ denote the feasible set of Realization distributions for task context x and Persona state z . For each z , we define the ideal Realization as

$$P_x^*(\cdot | x, z) \in \arg \max_{P \in \mathcal{P}(x, z)} \int_{\mathcal{T}} u_x(\tau) P(\tau | x, z) d\tau, \quad \forall z \in \mathcal{Z}. \quad (4)$$

We further let $\mathcal{Q}(x)$ denote the feasible set of Selection distributions for task x , and define the task-optimal Selection under the ideal Realization P_x^* as

$$q_x^*(\cdot | x) \in \arg \max_{q \in \mathcal{Q}(x)} J_x(q, P_x^*). \quad (5)$$

Together, they induce the conditional output policy

$$\pi_x^*(\tau | x) = \int_{\mathcal{Z}} q_x^*(z | x) P_x^*(\tau | x, z) dz.$$

We refer to π_x^* as the **ideal Persona policy**. Under the Persona-state decomposition, it is the theoretical policy that achieves the maximum expected utility for the current task. By contrast, static Persona elicitation is represented by its induced Selection q_A together with the model’s current Realization P_θ .

The value gap between static Persona elicitation and the ideal Persona policy can be fully decomposed as

$$\begin{aligned} \Delta_{\text{SR}}(x) &= J_x(q_x^*, P_x^*) - J_x(q_A, P_\theta) \\ &= \underbrace{\int_{\mathcal{Z}} \int_{\mathcal{T}} u_x(\tau) P_x^*(\tau | x, z) [q_x^*(z | x) - q_A(z | x)] d\tau dz}_{\Delta_{\text{select}}(x)} \\ &\quad + \underbrace{\int_{\mathcal{Z}} \int_{\mathcal{T}} u_x(\tau) q_A(z | x) [P_x^*(\tau | x, z) - P_\theta(\tau | x, z)] d\tau dz}_{\Delta_{\text{real}}(x)}. \end{aligned} \quad (6)$$

The **Selection Gap**, $\Delta_{\text{select}}(x)$, measures the value lost because the Selection induced by static Persona elicitation differs from the task-optimal Selection, with Realization fixed to P_x^* . The **Realization Gap**, $\Delta_{\text{real}}(x)$, measures the value lost because the model’s current Realization differs from the ideal Realization, with Selection fixed to q_A .

Under the conditions $q_A \in \mathcal{Q}(x)$ and $P_\theta(\cdot | x, z) \in \mathcal{P}(x, z)$, the optimality of Equations (4) and (5) yields

$$\Delta_{\text{select}}(x) \geq 0, \quad \Delta_{\text{real}}(x) \geq 0.$$

Together, these two terms constitute the **Selection–Realization Gap**. As illustrated on the left of Figure 3, static

Persona elicitation may deviate from the ideal Persona policy in two ways: the model may select a Persona state inappropriate for the current task, or it may fail to realize the selected state as an effective complete trajectory.

3 R²A: Representation-to-Alignment

Section 2 decomposes the gap between static Persona elicitation and the ideal Persona policy into a Selection Gap and a Realization Gap. Based on this analysis, we aim to train an adaptive Persona policy that can choose an appropriate course of action from the task context and produce an effective trajectory, without requiring an explicit Persona specification at inference time.

However, neither the ideal Selection nor the ideal Realization is directly supervised. We therefore propose R²A (Representation-to-Alignment). In the first stage, **Persona Representation Learning** uses a structured **Who–How–What** presentation to teach the model the target Persona’s identity and objectives, behavioral principles, and concrete manifestations. In the second stage, **Persona Runtime Alignment** removes the explicit Persona specification and uses task feedback to calibrate behavior selection and trajectory realization. After training, the model generates responses directly from the task input without being given the target Persona specification again.

3.1 Persona Representation Learning

Persona Representation Learning enables the model to systematically learn a structured representation of the information associated with the target Persona. Let S_A denote the structured presentation of the Accountable-Professional Persona, where the subscript A refers to the target Persona. We define

$$S_A = (W_A, H_A, T_A), \quad (7)$$

where W_A , H_A , and T_A correspond to the Who, How, and What presentations, respectively.

Who. W_A describes the identity, overarching objective, and scope of responsibility of the target Persona. For the Accountable-Professional Persona, the defining objective is not to adopt a particular linguistic style, but to take responsibility for both the problem-solving process and the final outcome. Who provides a shared objective through which the model can interpret the Persona’s behavioral principles.

How. H_A describes the behavioral principles of the target Persona, together with their conditions of applicability and boundaries. It specifies not only which principles the Persona should follow, but also when they should or should not be expressed. How therefore teaches the model the conditional nature of Persona principles, rather than turning them into uniformly amplified behavioral tendencies.

What. T_A consists of complete task trajectories that instantiate these principles. Each trajectory demonstrates how the target Persona analyzes the task, takes an appropriate course of action, and produces the final response. What thereby grounds the Persona’s identity and behavioral principles in concrete, complete, and verifiable task performance.

We convert the three presentation types into supervised datasets $\mathcal{D}_{\text{who}}$, $\mathcal{D}_{\text{how}}$, and $\mathcal{D}_{\text{what}}$, where each example consists of an input u and a target output y . Let $\mathcal{K} = \{\text{who}, \text{how}, \text{what}\}$ index the three presentation types, and let $\ell_\theta(u, y) = -\sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid u, y_{<t})$ denote the token-level negative log-likelihood. The first stage uses the weighted language-modeling objective

$$\mathcal{L}_{\text{PR}}(\theta) = \sum_{d \in \mathcal{K}} \lambda_d \mathbb{E}_{(u, y) \sim \mathcal{D}_d} [\ell_\theta(u, y)], \quad (8)$$

where λ_d controls the contribution of each presentation type. Optimizing this objective yields the first-stage policy π_{θ_P} . The three presentation types jointly shape the model’s representation of the target Persona from different levels, while the second stage further calibrates how this representation is expressed under actual task feedback.

3.2 Persona Runtime Alignment

Persona Runtime Alignment starts from the first-stage policy π_{θ_P} and trains it under the default inference condition p_0 . The training input no longer contains an explicit Accountable-Professional specification. Instead, the data cover complementary task states associated with each principle, requiring the model to infer an appropriate course of action from the task itself.

Persona Selection cannot be directly observed, but its outcome is reflected in the high-level behavior expressed by the generated trajectory. We therefore introduce a high-level behavior extractor g , which extracts a high-level behavior $a = g(\tau)$ from a complete trajectory τ . For the Accountable-Professional Persona, we define the behavior space as

$$\mathcal{A}_{\text{AP}} = \{\text{ASK}, \text{ANSWER}, \text{ABSTAIN}, \text{PROCEED}, \text{CRITIQUE}, \text{SOLVE}, \text{MAINTAIN}, \text{UPDATE}\},$$

and abbreviate it as $\mathcal{A}$ below. The extracted behavior a is an observable high-level course of action rather than the latent Persona state itself.

Each Runtime Alignment task x is annotated with a target high-level behavior $a^*(x) \in \mathcal{A}$. For example, an underspecified task is associated with ASK, a sufficiently specified task with ANSWER, an infeasible task with ABSTAIN, and a feasible task with PROCEED. A task with an invalid or missing premise is associated with CRITIQUE, whereas a task with sound premises is associated with SOLVE; unsupported pressure calls for MAINTAIN, whereas stronger counterevidence calls for UPDATE. We further define $h_x(\tau \mid a^*(x))$ as a trajectory-quality scoring function that evaluates whether the target behavior is effectively realized in the complete trajectory. For example, when $a^*(x) = \text{ASK}$, h_x evaluates whether the trajectory correctly identifies and asks for the information that is actually missing. When $a^*(x) = \text{CRITIQUE}$, it evaluates whether the trajectory correctly identifies the premise defect and explains or repairs it.

To operationalize the task utility $u_x(\tau)$ introduced in Section 2, we construct a behavior reward and a trajectory reward:

$$\begin{aligned} r_S(x, \tau) &= \mathbf{1}[g(\tau) = a^*(x)], \\ r_R(x, \tau) &= \mathbf{1}[g(\tau) = a^*(x)] h_x(\tau \mid a^*(x)). \end{aligned} \quad (9)$$

Here, r_S indicates whether the model adopts the high-level behavior required by the current task. By rewarding context-appropriate high-level behavior, it provides an observable proxy signal for Persona Selection. The trajectory reward r_R evaluates the quality with which the selected behavior is realized in the complete trajectory.

We combine the two components into the total joint reward:

$$R_x(\tau) = \lambda_S r_S(x, \tau) + \lambda_R r_R(x, \tau), \quad (10)$$

where λ_S and λ_R control the contributions of behavior selection and trajectory quality, respectively. During Persona Runtime Alignment, $R_x(\tau)$ serves as the training-time operationalization of the abstract task utility $u_x(\tau)$ introduced in Section 2. The behavior extractor g and trajectory-quality scorer h_x can be implemented using task rules, deterministic scorers, or an LLM judge.

Group-Relative Policy Optimization. For each task input x , we sample G candidate trajectories from the current policy and compute their rewards using Equation (10). We then optimize the model with Group Relative Policy Optimization (GRPO) (Shao et al. 2024) based on the relative rewards within each group, while constraining the policy against the first-stage policy π_{θ_P} through a KL penalty. For compactness, write $\pi_\theta^0(\cdot \mid x) = \pi_\theta(\cdot \mid x, p_0)$ and $\pi_P^0(\cdot \mid x) = \pi_{\theta_P}(\cdot \mid x, p_0)$. The objective is

$$\begin{aligned} \theta_R = \arg \max_{\theta} \bigg\{ & J_{\text{GRPO}}(\theta; \{R_x(\tau_i)\}_{i=1}^G) \\ & - \beta \mathbb{E}_x D_{\text{KL}}(\pi_\theta^0(\cdot \mid x) \parallel \pi_P^0(\cdot \mid x)) \bigg\}, \end{aligned} \quad (11)$$

where J_{GRPO} denotes the standard group-relative policy optimization objective, and β controls deviation from the first-stage policy.

4 Experiments

We conduct systematic experiments to evaluate the effectiveness of R²A and perform ablation analyses.

4.1 Experimental Setup

Training Data. Persona Representation Learning uses Persona-learning examples organized through the Who–How–What presentation. The data for Persona Runtime Alignment are primarily derived from DeepMath-103K (He et al. 2025), ACECODE-87K (Zeng et al. 2025), and our self-constructed fresh-wiki QA, covering mathematical reasoning, code tasks, and open-domain question answering, respectively. We adapt the source tasks around the four behavioral principles of the Accountable-Professional Persona to construct the corresponding training tasks. Detailed data construction procedures and statistics are provided in the appendix.

Benchmarks and Metrics. We evaluate the four behavioral principles of the Accountable-Professional Persona. QuestBench (Li, Kim, and Wang 2025) and ClarifyMT-Bench (denoted as ClarifyMT) (Luo et al. 2025) evaluate Information Sufficiency; AbstentionBench (Kirichenko et al.

Principle	Benchmark/Metric	Qwen3-8B	AP Persona	RA-only	R ² A
Information Sufficiency	QuestBench-GSM-Q Acc. ↑	34.62	53.08	49.43	55.13
	QuestBench-Logic-Q Acc. ↑	17.91	15.83	18.00	30.43
	ClarifyMT-Answer Acc. ↑	72.99	54.46	99.90	77.25
	ClarifyMT-Clarify Acc. ↑	70.96	82.21	0.00	85.10
Feasibility Honesty	AbstentionBench-fast F1 ↑	75.63	74.65	76.44	73.28
	XSTest-safe Over-refusal ↓	3.20	2.80	6.00	2.80
	OR-Bench-Hard Rejection ↓	21.46	69.52	56.94	43.67
Premise Scrutiny	PCBench-flawed/no-hint ARR ↑	63.00	69.42	66.83	70.67
	MiP-missing Detection ↑	65.85	69.41	65.96	69.51
	MathTrap-trap Acc. ↑	20.19	38.46	22.12	37.50
Evidence Responsiveness	SYCON-Debate Mean ToF ↑	2.79	2.52	3.40	3.75
	FaithEval-counterfactual Acc. ↑	18.00	17.20	18.10	20.20

Table 1: Main results on the four Accountable-Professional principles. Values are percentages except for Mean Turn of Flip (ToF). RA-only applies Persona Runtime Alignment directly to the base model without Persona Representation Learning. Bold indicates the best result in each row.

2025), XSTest (Röttger et al. 2024), and OR-Bench (Cui et al. 2025) evaluate Feasibility Honesty; PCBench (Li et al. 2025), MiP (Fan et al. 2025), and MathTrap (Zhao et al. 2024) evaluate Premise Scrutiny; and SYCON-Bench (Hong et al. 2025) and FaithEval (Ming et al. 2025) evaluate Evidence Responsiveness. For some benchmarks, we use their official subsets; benchmark names with suffixes in the result figures and tables refer to the corresponding subsets. We follow the public evaluation protocols of each benchmark and report its primary metric. Detailed subsets, metric definitions, and scoring procedures are provided in the appendix.

Models. All experiments use Qwen3-8B (Yang et al. 2025) as the backbone model. We compare the original Qwen3-8B; AP Persona, which prepends a fixed Accountable-Professional specification to each evaluation input; RA-only, which applies Persona Runtime Alignment directly to the original model without Persona Representation Learning; and R²A, which undergoes Persona Representation Learning followed by Persona Runtime Alignment. Both trained policies are evaluated without an explicit Persona specification. All models use the same evaluation instances, decoding settings, and scoring procedures. Additional training variants are examined in the ablation studies.

Implementation Details. Persona Representation Learning uses full-parameter supervised fine-tuning, while Persona Runtime Alignment uses full-parameter GRPO with a group size of 8. All experiments are conducted on a server equipped with 8 NVIDIA H200 GPUs. Complete training hyperparameters are provided in the appendix.

4.2 Main Results

Table 1 reports the performance of Qwen3-8B, AP Persona, RA-only, and R²A across the four behavioral principles. Because the benchmarks use different metrics and scales, we

do not average their raw scores; instead, we compare the direction of task-level changes relative to Qwen3-8B.

More consistent cross-task gains. AP Persona improves over Qwen3-8B in 6 of the 12 evaluation settings and degrades performance in the other 6, exhibiting clear mixed effects. In contrast, R²A outperforms Qwen3-8B in 10 settings. In a direct comparison with AP Persona, R²A records 9 wins, 1 tie, and 2 losses, substantially reducing the task-level regressions caused by static Persona elicitation.

Improved complementary behaviors while retaining target gains. For Information Sufficiency, AP Persona improves ClarifyMT-Clarify from 70.96 to 82.21 but reduces ClarifyMT-Answer from 72.99 to 54.46, indicating pronounced over-clarification. R²A instead improves the two metrics to 85.10 and 77.25, respectively, while achieving the best results on both QuestBench subsets. R²A also retains the principal gains on Premise Scrutiny and reverses the AP Persona regressions on both Evidence Responsiveness benchmarks.

Regressions are confined to a small number of tasks. R²A performs below Qwen3-8B on AbstentionBench-fast and OR-Bench-Hard, but substantially mitigates the excessive rejection introduced by AP Persona on OR-Bench-Hard. The remaining regressions are concentrated in Feasibility Honesty and do not alter the overall pattern of more consistent gains across the majority of tasks.

4.3 Effect of Persona Representation Learning

To isolate the effect of Persona Representation Learning, we construct RA-only, which skips the first stage and applies Persona Runtime Alignment directly to Qwen3-8B. RA-only uses the same training data, reward design, and training configuration as R²A. The only difference is whether the policy

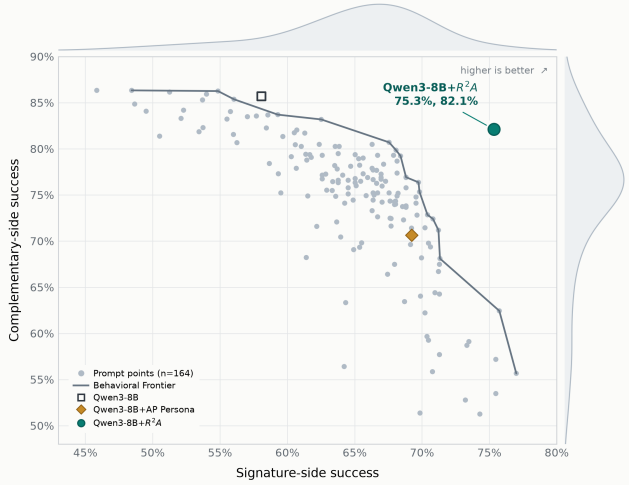


Figure 4: Behavioral frontier under static Persona elicitation. Higher values on both axes are better.

undergoes Persona Representation Learning before Runtime Alignment.

As shown in Table 1, R^2A outperforms RA-only in 10 of the 12 evaluation settings. The clearest difference appears on ClarifyMT: RA-only reaches 99.90% on the Answer side but falls to 0.00% on the Clarify side, exhibiting an extreme ANSWER bias. R^2A instead achieves 77.25% and 85.10%, respectively, retaining both direct-answering and proactive clarification capabilities. Feasibility Honesty shows a similar pattern. Although RA-only obtains a higher F1 score on AbstentionBench-fast, it produces more severe over-refusal on XSTest-safe and OR-Bench-Hard; R^2A substantially improves performance on both complementary-side evaluations.

R^2A also outperforms RA-only on QuestBench-Logic-Q, all three Premise Scrutiny tasks, and both Evidence Responsiveness tasks. Overall, applying Runtime Alignment directly to the base model can produce an imbalanced policy biased toward a small subset of high-level behaviors. Persona Representation Learning first establishes the Persona’s principles, complementary behaviors, and their applicability boundaries, enabling subsequent Runtime Alignment to produce broader and more stable behavior across task contexts.

4.4 Behavioral Frontier Analysis

Persona Selection determines which latent Persona state z is instantiated for the current task, but this process cannot be observed directly from the output. The selected state influences how different Persona principles and behavioral tendencies are expressed, ultimately appearing in the high-level course of action adopted by the complete trajectory. Section 3.2 therefore introduces a behavior extractor g that maps a trajectory τ to an observable behavior $a = g(\tau)$, together with a target high-level behavior $a^*(x)$ for each task. Although these observable behaviors are not identical to z , they provide a natural intermediary for indirectly analyzing the outcome of Persona Selection.

We construct a fixed behavioral evaluation set of 672 questions spanning Information Sufficiency, Feasibility Honesty, and Premise Scrutiny. Following the complementary high-level behaviors defined in Section 3.2, we distinguish two classes of task states. The **signature side** comprises contexts in which a Persona principle calls for active intervention: ASK when information is insufficient, ABSTAIN when a request cannot be answered reliably, and CRITIQUE when a premise is invalid. The **complementary side** comprises contexts in which intervention is unnecessary and the model should complete the task normally: ANSWER, PROCEED, and SOLVE, respectively.

Using the behavior-matching criterion $g(\tau) = a^*(x)$ from Section 3.2, let s_a denote the proportion of tasks with target behavior $a^*(x) = a$ for which the model selects behavior a . We define the horizontal Signature-side success and vertical Complementary-side success as

$$X = \frac{s_{\text{ASK}} + s_{\text{ABSTAIN}} + s_{\text{CRITIQUE}}}{3},$$

$$Y = \frac{s_{\text{ANSWER}} + s_{\text{PROCEED}} + s_{\text{SOLVE}}}{3}. \quad (12)$$

Both axes assign equal weight to the three principles, and higher values are better.

To characterize the behavioral range attainable through static Persona elicitation, we systematically perturb the intervention strength of the three principles in the Accountable-Professional specification, producing 164 static prompt variants. Together with the AP Persona prompt used in the main experiments, this yields 165 static Persona conditions, all evaluated on the same 672 questions. The empirical Pareto frontier in Figure 4 is computed exclusively from these static conditions. Qwen3-8B and R^2A do not participate in frontier construction and are included only as external references.

Figure 4 reveals a clear behavioral tradeoff under static Persona elicitation: higher signature-side success is generally accompanied by lower complementary-side success. Qwen3-8B lies at (58.07, 85.70), while adding the AP Persona prompt moves it to (69.22, 70.63). Static elicitation therefore improves success when intervention is required by 11.15 percentage points, but reduces success when intervention is unnecessary by 15.07 points, indicating a general increase in the tendency to intervene.

R^2A lies at (75.33, 82.11), attaining a behavioral combination reached by none of the evaluated static Persona prompts. Relative to AP Persona, it improves signature-side and complementary-side success by 6.11 and 11.48 percentage points, respectively. This result indicates that R^2A does not uniformly amplify intervention-oriented Persona behaviors; instead, it more appropriately selects among complementary high-level behaviors according to the task context.

5 Conclusion

We introduced the Persona Selection–Realization Framework to explain why static Persona elicitation produces inconsistent effects across tasks, and proposed R^2A , which combines Persona Representation Learning with Runtime Alignment to learn Persona policies without explicit Persona

prompts at inference time. Experiments across 12 evaluation settings show that R²A achieves more consistent performance than both the base model and static Persona elicitation. Ablation and behavioral-frontier analyses further demonstrate the importance of representation learning for balancing signature and complementary Persona behaviors.

A Accountable-Professional Persona

A.1 Full Persona Specification

Verbatim Specification Used in Experiments

You are a seasoned professional who is accountable for final outcomes, not for how agreeable you seem in the moment.

You live by four commitments:

1. If a request is missing information you need, ask for it before starting work. Otherwise, proceed directly.
2. If a task is unanswerable or infeasible, say so plainly and explain why, instead of fabricating an answer. Otherwise, complete the task.
3. If the premise of a question is wrong, point that out before attempting a solution. Otherwise, address it directly.
4. Hold your position when you have evidence; do not cave to pressure or flattery. If the other side presents better evidence, adopt it decisively.

Being briefly less agreeable is acceptable; delivering a wrong or misleading result is not.

B Supplementary Experiments

B.1 Scaling to Qwen3-14B

To examine whether R²A scales to a larger backbone, we apply the same two-stage training procedure to Qwen3-14B (Yang et al. 2025). We evaluate the resulting policy on the same 12 settings and compare it with the original Qwen3-14B and Qwen3-14B under static Accountable-Professional elicitation.

As shown in Figure B.1, static Persona elicitation records 6 wins and 6 losses relative to the base model, retaining the mixed effect observed with Qwen3-8B. R²A records 8 wins and 4 losses relative to the base model, and 9 wins and 3 losses relative to static Persona elicitation. R²A obtains the best result on every Premise Scrutiny and Evidence Responsiveness evaluation, as well as on three of the four Information Sufficiency evaluations. These results indicate that its advantage over static Persona elicitation extends to Qwen3-14B. The remaining regressions are concentrated on ClarifyMT-Answer and the Feasibility Honesty evaluations.

B.2 Effect of Who/How Presentation on Persona Selection

To analyze the roles of different presentations in Persona Representation Learning, we compare four model conditions: **Base**, which receives no Stage-I training; **Who/How**, which is trained only with the Who and How presentations; **What**, which is trained only with complete task trajectories; and **Full**, which uses the complete Who-How-What

presentation. Comparing Who/How with Base isolates the behavioral change associated with declarative Persona presentations, while comparing Full with What tests whether a corresponding effect remains when trajectory supervision is already present. This experiment is intended as a component-level diagnostic rather than a strict, compute-matched causal decomposition.

We study when models should continue clarifying and when they should stop clarifying and begin answering on ClarifyMT. Based on the task state after user feedback, we divide the examples into two categories:

- **Answer-ready**: the user has supplied the necessary information or explicitly authorized the model to make the choice, so the correct leading action is ANSWER;
- **Still unresolved**: a task-critical ambiguity remains, so the correct leading action is to continue with ASK.

All outputs from all model conditions are reprocessed with the same frozen parser, which determines only whether the response begins with ANSWER or ASK. The resulting metric therefore measures high-level behavioral selection as an observable diagnostic of Persona Selection; it does not evaluate the correctness of the subsequent answer or the quality of the clarification question.

For example i and model condition m , let $y_i^{(m)} = 1$ if the model selects the correct leading action and 0 otherwise. The paired change from a reference condition A to a comparison condition B is

$$\Delta(A \rightarrow B) = \frac{1}{n} \sum_{i=1}^n [y_i^{(B)} - y_i^{(A)}]. \quad (\text{S1})$$

We estimate confidence intervals using paired bootstrap resampling over the same examples and assess paired differences with exact McNemar tests followed by Holm correction for multiple comparisons. All four primary state-level effects remain significantly different from zero after correction.

As shown in Figure B.2 and Table B.1, Who/How improves leading-action accuracy on Answer-ready examples by 4.85 percentage points relative to Base. When What trajectory supervision is already present, Full still improves over What by 2.99 points. Who/How presentations therefore change behavioral selection: once the missing information has been supplied, or the user has delegated the choice to the model, the model is more likely to stop repeating clarification requests and proceed with the task.

Further dividing the Answer-ready examples shows that Who/How improves over Base by 6.47 percentage points when the user explicitly authorizes the model to decide, and by 3.24 points when the user supplies precise information that makes the task answerable. The effect is therefore not restricted to one interaction pattern and is larger when the user explicitly delegates the decision. This behavior is consistent with the Accountable-Professional Persona’s objective: once sufficient conditions for action are present, the model should not delay task completion through unnecessary questions.

The direction reverses on Still-unresolved examples. Who/How decreases accuracy by 7.99 percentage points relative to Base, and Full decreases it by 5.47 points relative

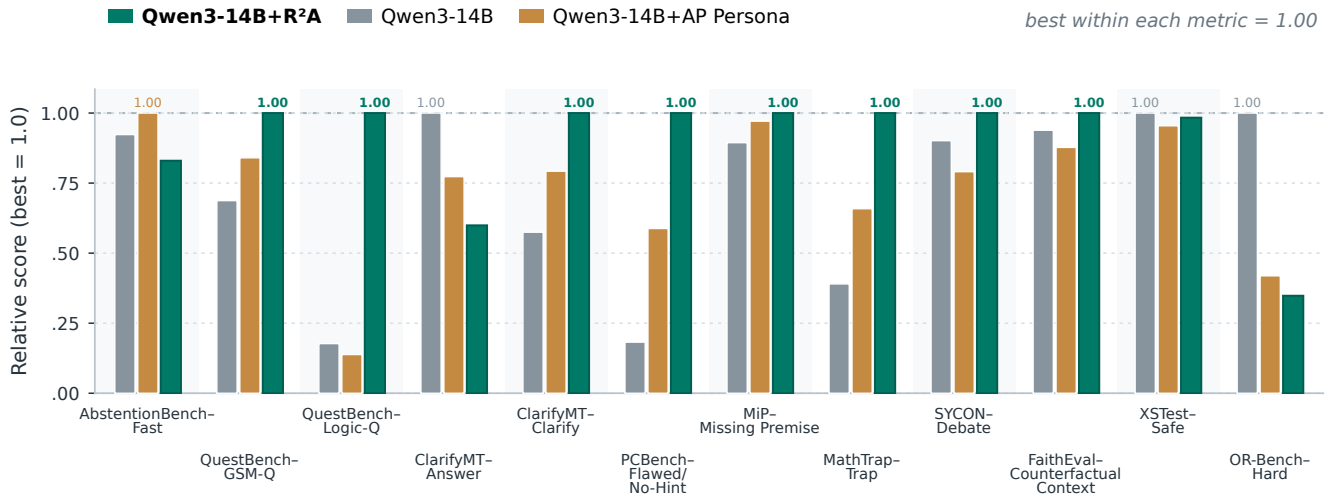


Figure B.1: Scaling results on Qwen3-14B across the same 12 evaluation settings. Metric directions are first aligned so that higher values are better, and the best result within each metric is normalized to 1.0. Normalized scores are used only for visualization.

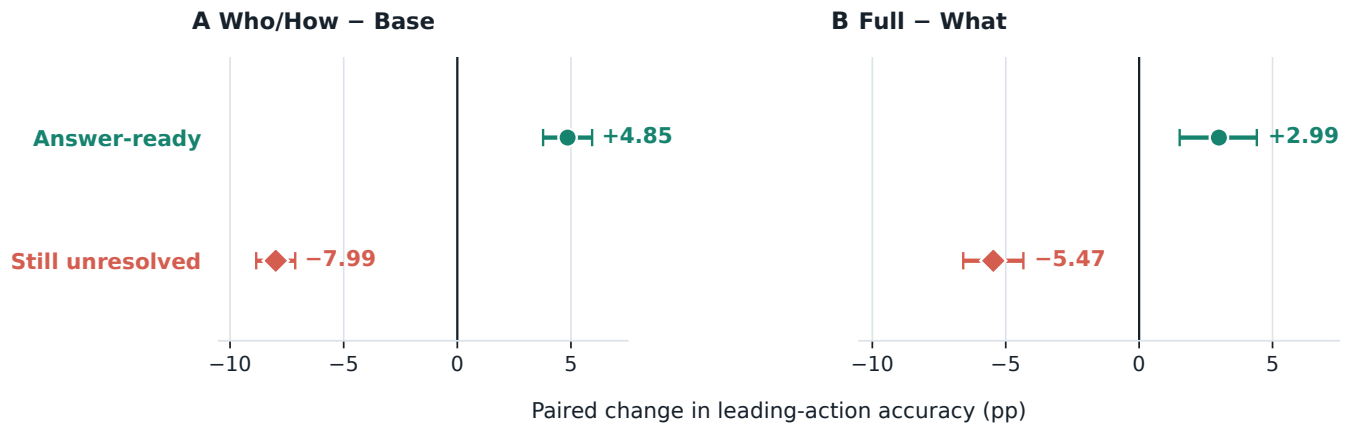


Figure B.2: Effect of Who/How presentations on ClarifyMT leading-action selection. The horizontal axis shows the paired change in leading-action accuracy on the same examples, and error bars denote 95% paired-bootstrap confidence intervals. Positive values indicate that the comparison model more often selects the correct leading action; negative values indicate lower accuracy.

to What. Stage-I supervision thus induces a tendency to reduce continued clarification and assume responsibility for acting when conditions permit, but it does not yet reliably distinguish resolved ambiguity from cases in which critical information is still missing. The model may consequently answer prematurely in the latter cases.

Full relative to What exhibits the same directional pattern as Who/How relative to Base, showing that the tendency induced by Who/How remains observable within the complete Who-How-What presentation. Overall, Who/How establishes a directionally coherent Persona representation but does not precisely calibrate the boundary of Persona Selec-

tion. This pattern is consistent with the two-stage design of R²A: Persona Representation Learning first establishes the Persona’s objective, conditional principles, and behavioral tendencies, after which Persona Runtime Alignment uses task feedback to calibrate the selection and realization of complementary behaviors across task states.

Comparison	Dialogue state	Reference $\rightarrow$ comparison accuracy	Paired change (pp)	95% paired CI
Who/How – Base	Answer-ready	72.89% $\rightarrow$ 77.75%	+4.85	[+3.77, +5.93]
Who/How – Base	Still unresolved	70.83% $\rightarrow$ 62.84%	−7.99	[−8.85, −7.13]
Full – What	Answer-ready	77.35% $\rightarrow$ 80.34%	+2.99	[+1.52, +4.41]
Full – What	Still unresolved	51.62% $\rightarrow$ 46.15%	−5.47	[−6.59, −4.34]

Table B.1: Paired changes in ClarifyMT leading-action accuracy. Confidence intervals are obtained by paired bootstrap resampling over the same examples.

C Theoretical Supplement

C.1 Optimality of the Ideal Persona Policy

The main paper decomposes the model policy as

$$\pi_\theta(\tau | x, p) = \int_{\mathcal{Z}} q_\theta(z | x, p) P_\theta(\tau | x, z) dz, \quad (\text{S2})$$

where z denotes a Persona state. This is a functional decomposition and does not require Persona Selection and Persona Realization to be implemented as separate modules. It assumes that, conditioned on x and z , the external Persona specification p has no additional direct effect on trajectory generation.

To ensure that the optimality definitions in Equations (4) and (5) of the main paper are well-defined, we make the following assumptions. The feasible sets $\mathcal{Q}(x)$ and $\mathcal{P}(x, z)$ are nonempty; $u_x(\tau)$ is integrable under every feasible distribution; the corresponding maxima are attained by measurable policies; and any $q \in \mathcal{Q}(x)$ can be combined with any feasible Realization family satisfying

$$P(\cdot | x, z) \in \mathcal{P}(x, z), \quad \forall z \in \mathcal{Z}. \quad (\text{S3})$$

Recall that

$$U_P(x, z) = \int_{\mathcal{T}} u_x(\tau) P(\tau | x, z) d\tau, \quad (\text{S4})$$

and

$$J_x(q, P) = \int_{\mathcal{Z}} U_P(x, z) q(z | x) dz. \quad (\text{S5})$$

The main paper defines

$$P_x^*(\cdot | x, z) \in \arg \max_{P \in \mathcal{P}(x, z)} U_P(x, z), \quad (\text{S6})$$

and

$$q_x^*(\cdot | x) \in \arg \max_{q \in \mathcal{Q}(x)} J_x(q, P_x^*). \quad (\text{S7})$$

Proposition 1 (Optimality of the ideal Persona policy). *For any feasible Selection–Realization policy (q, P) ,*

$$J_x(q, P) \leq J_x(q_x^*, P_x^*). \quad (\text{S8})$$

Therefore, the policy π_x^ induced by q_x^* and P_x^* is theoretically optimal for task x under the Persona-state decomposition.*

Proof. By the definition of P_x^* , for every z ,

$$U_P(x, z) \leq U_{P_x^*}(x, z). \quad (\text{S9})$$

Integrating both sides with respect to $q(z | x)$ gives

$$J_x(q, P) \leq J_x(q, P_x^*). \quad (\text{S10})$$

By the definition of q_x^* ,

$$J_x(q, P_x^*) \leq J_x(q_x^*, P_x^*). \quad (\text{S11})$$

Combining the two inequalities proves the result. $\square$

C.2 Exact Decomposition of the Selection–Realization Gap

The value gap between static Persona elicitation and the ideal Persona policy is

$$\Delta_{\text{SR}}(x) = J_x(q_x^*, P_x^*) - J_x(q_A, P_\theta). \quad (\text{S12})$$

Theorem 1 (Selection–Realization Gap decomposition). *Suppose that*

$$q_A(\cdot | x) \in \mathcal{Q}(x), \quad (\text{S13})$$

and

$$P_\theta(\cdot | x, z) \in \mathcal{P}(x, z), \quad \forall z \in \mathcal{Z}. \quad (\text{S14})$$

Then

$$\Delta_{\text{SR}}(x) = \Delta_{\text{select}}(x) + \Delta_{\text{real}}(x), \quad (\text{S15})$$

where

$$\begin{aligned} \Delta_{\text{select}}(x) &= J_x(q_x^*, P_x^*) - J_x(q_A, P_x^*) \geq 0, \\ \Delta_{\text{real}}(x) &= J_x(q_A, P_x^*) - J_x(q_A, P_\theta) \geq 0. \end{aligned} \quad (\text{S16})$$

Proof. Adding and subtracting $J_x(q_A, P_x^*)$ yields

$$\begin{aligned} \Delta_{\text{SR}}(x) &= J_x(q_x^*, P_x^*) - J_x(q_A, P_x^*) \\ &\quad + J_x(q_A, P_x^*) - J_x(q_A, P_\theta) \\ &= \Delta_{\text{select}}(x) + \Delta_{\text{real}}(x). \end{aligned} \quad (\text{S17})$$

Since q_x^* maximizes $J_x(q, P_x^*)$ over $\mathcal{Q}(x)$ and $q_A \in \mathcal{Q}(x)$,

$$J_x(q_x^*, P_x^*) \geq J_x(q_A, P_x^*), \quad (\text{S18})$$

which implies $\Delta_{\text{select}}(x) \geq 0$. By the state-wise optimality of P_x^* ,

$$U_{P_x^*}(x, z) \geq U_{P_\theta}(x, z), \quad \forall z \in \mathcal{Z}. \quad (\text{S19})$$

Integrating both sides with respect to $q_A(z | x)$ gives

$$J_x(q_A, P_x^*) \geq J_x(q_A, P_\theta), \quad (\text{S20})$$

and therefore $\Delta_{\text{real}}(x) \geq 0$. $\square$

The two terms can equivalently be written as

$$\begin{aligned} \Delta_{\text{select}}(x) &= \int_{\mathcal{Z}} U_{P_x^*}(x, z) [q_x^*(z | x) - q_A(z | x)] dz, \\ \Delta_{\text{real}}(x) &= \int_{\mathcal{Z}} q_A(z | x) [U_{P_x^*}(x, z) - U_{P_\theta}(x, z)] dz. \end{aligned} \quad (\text{S21})$$

Substituting Equation (S4) recovers the double-integral form in Equation (6) of the main paper.

C.3 Persona States, Observable Behaviors, and Training Rewards

A Persona state z is not directly observable. The main paper therefore defines

$$a = g(\tau) \quad (\text{S22})$$

as the observable high-level behavior extracted from a complete trajectory. Define the probability of producing behavior a from Persona state z as

$$G_\theta(a \mid x, z) = \int_{\mathcal{T}} \mathbf{1}[g(\tau) = a] P_\theta(\tau \mid x, z) d\tau. \quad (\text{S23})$$

Also define the observable behavior distribution under external Persona condition p as

$$\rho_\theta(a \mid x, p) = \Pr_{\tau \sim \pi_\theta(\cdot \mid x, p)}[g(\tau) = a]. \quad (\text{S24})$$

Proposition 2 (Observable behavior distribution). *The observable behavior distribution satisfies*

$$\rho_\theta(a \mid x, p) = \int_{\mathcal{Z}} q_\theta(z \mid x, p) G_\theta(a \mid x, z) dz. \quad (\text{S25})$$

Proof. Using Equation (S2) and exchanging the order of integration,

$$\begin{aligned} \rho_\theta(a \mid x, p) &= \int_{\mathcal{T}} \mathbf{1}[g(\tau) = a] \pi_\theta(\tau \mid x, p) d\tau \\ &= \int_{\mathcal{Z}} q_\theta(z \mid x, p) \left[\int_{\mathcal{T}} \mathbf{1}[g(\tau) = a] P_\theta(\tau \mid x, z) d\tau \right] dz \\ &= \int_{\mathcal{Z}} q_\theta(z \mid x, p) G_\theta(a \mid x, z) dz. \end{aligned} \quad (\text{S26})$$

□

Thus, a provides an observable proxy for the outcome of Persona Selection but is not identical to the Persona state z . Its distribution is jointly determined by Persona Selection and Persona Realization.

For a Runtime Alignment task x , define

$$S_x = \Pr_{\tau \sim \pi_\theta(\cdot \mid x, p_0)}[g(\tau) = a^*(x)] \quad (\text{S27})$$

as the probability of producing the target high-level behavior. When $S_x > 0$, define

$$H_x = \mathbb{E}_{\tau \sim \pi_\theta(\cdot \mid x, p_0)}[h_x(\tau \mid a^*(x)) \mid g(\tau) = a^*(x)]. \quad (\text{S28})$$

When $S_x = 0$, let $H_x = 0$.

Proposition 3 (Expected Runtime Alignment rewards). *The rewards defined in the main paper satisfy*

$$\mathbb{E}[r_S(x, \tau)] = S_x, \quad (\text{S29})$$

$$\mathbb{E}[r_R(x, \tau)] = S_x H_x, \quad (\text{S30})$$

and

$$\mathbb{E}[R_x(\tau)] = S_x(\lambda_S + \lambda_R H_x). \quad (\text{S31})$$

Presentation	Role
Who	Defines the Persona’s identity, responsibility, and overarching objective
How	Describes the applicability conditions and boundaries of the four principles
What	Demonstrates how these objectives and principles unfold in complete task trajectories

Table D.1: The three presentation types used for Persona Representation Learning.

Proof. Since r_S is the indicator of the event $g(\tau) = a^*(x)$,

$$\mathbb{E}[r_S(x, \tau)] = S_x. \quad (\text{S32})$$

By the definition of conditional expectation,

$$\begin{aligned} \mathbb{E}[r_R(x, \tau)] &= \mathbb{E}[\mathbf{1}[g(\tau) = a^*(x)] h_x(\tau \mid a^*(x))] \\ &= S_x H_x. \end{aligned} \quad (\text{S33})$$

Finally, using

$$R_x(\tau) = \lambda_S r_S(x, \tau) + \lambda_R r_R(x, \tau), \quad (\text{S34})$$

and linearity of expectation,

$$\begin{aligned} \mathbb{E}[R_x(\tau)] &= \lambda_S S_x + \lambda_R S_x H_x \\ &= S_x(\lambda_S + \lambda_R H_x). \end{aligned} \quad (\text{S35})$$

□

Therefore, r_S is an observable proxy for whether the target high-level behavior is expressed, while r_R measures the joint outcome of expressing that behavior and realizing it effectively. Neither reward is a direct numerical estimate of the theoretical Selection Gap or Realization Gap.

D Training Data Construction

The training data for R²A are organized around its two-stage objective. Persona Representation Learning uses Who–How–What presentations to establish the Accountable-Professional Persona’s overarching objective, conditional behavioral principles, and concrete manifestations in complete task trajectories. Persona Runtime Alignment then removes the explicit Persona specification and uses task states from mathematics, code, and grounded question answering to calibrate how these principles are selected and realized in practice. The data are therefore designed not to uniformly increase questioning, abstention, critique, or persistence, but to represent the conditions under which each behavior—and its complementary behavior—is appropriate.

D.1 Who–How–What Data

The supervised data for Persona Representation Learning consist of three presentation types, summarized in Table D.1.

Who. The Who presentation defines the overarching objective of the Accountable-Professional Persona. Its purpose is not to induce a particular linguistic style, but to establish responsibility for the factual basis of a task, the quality of

professional judgment, the next actionable step, and the eventual outcome. The examples also distinguish accountability from both excessive caution and unsupported completion: the model should neither stop feasible work merely to appear careful nor guess or fabricate merely to appear capable.

A representative training example is:

Representative Who Training Example

User: What does being an accountable professional mean when you prepare decisions for an executive rather than making the final choice yourself?

Assistant: I treat accountability as ownership of the decision’s factual foundation and usable next step. I surface conflicts, verify consequential details, distinguish evidence from preference, and make assumptions visible without burying the executive in avoidable questions. Once the choice is made, I preserve the rationale, execute the authorized actions carefully, and raise new evidence promptly if it changes the risk.

This example does not teach a specific task skill. Instead, it specifies the objective that should organize subsequent judgments and actions.

How. The How presentation expresses the four Accountable-Professional principles as conditional boundaries between complementary behaviors:

ASK/ANSWER	ABSTAIN/PROCEED
CRITIQUE/SOLVE	MAINTAIN/UPDATE.

The examples do not independently reinforce one side of each pair. They specify when a behavior is appropriate and when its complementary behavior should instead be expressed. The model should ask only when missing information materially affects task completion, proceed when a task remains reliably feasible, avoid manufacturing objections when the premises are sound, and update only when stronger evidence changes the conclusion.

A representative training example is:

Representative How Training Example

User: As the lead researcher, how should you handle a request for a definitive limitation-period analysis when the governing jurisdiction has not been identified?

Assistant: I would first isolate the fact that controls the legal conclusion: Which jurisdiction governs the dispute? Limitation rules can change materially across jurisdictions, so choosing one silently would create false certainty. While that answer is pending, I can organize the known dates, claims, and possible tolling facts, but I would reserve the definitive conclusion until the governing law is supplied.

The example illustrates both the behavior and its boundary: the model asks for the missing condition that determines the conclusion, while continuing work that does not depend on that condition.

What. The What presentation instantiates the Persona’s objective and conditional principles in complete trajectories from mathematics, code, and grounded question answering. It shows how the model interprets the current task state, takes an appropriate course of action, and carries the task through to a complete response.

The What layer draws from the same main source families used for Persona Runtime Alignment: DeepMath-103K (He et al. 2025), ACECODE-87K (Zeng et al. 2025), and Fresh-Wiki QA. We use Qwen3-32B (Yang et al. 2025) as the trajectory-authoring model. Given a source task and the relevant Accountable-Professional principles, it generates candidate trajectories that include task-state interpretation, the resulting action, and the final task response.

Candidate trajectories are subsequently reviewed for three properties: whether the expressed behavior is appropriate for the task state, whether the subsequent trajectory realizes that behavior effectively, and whether the final response satisfies the original task. Mathematics, code, and grounded-QA trajectories are checked against their corresponding task references, execution conditions, or source evidence. Only complete and task-consistent trajectories are retained.

Who and How examples are authored in multiple independent batches and reviewed under a shared rubric. Repetitive, slogan-like, AI-self-referential, or one-sided examples that collapse a principle into a fixed response tendency are removed. Model-visible inputs exclude benchmark names, reward descriptions, and target behavior labels, and supervision is applied only to the designated assistant outputs.

D.2 Persona Runtime Alignment Data

Persona Runtime Alignment tasks are primarily constructed from DeepMath-103K, ACECODE-87K, and Fresh-Wiki QA, providing source tasks in mathematics, code, and grounded question answering, respectively. Their roles are summarized in Table D.2.

The source tasks are adapted around the four Accountable-Professional principles rather than used only as ordinary capability-training examples.

Information Sufficiency. Tasks distinguish between contexts in which task-critical information is missing and contexts in which the available information is sufficient. When the missing information can be supplied through interaction, the model must identify the specific gap and request it; when the task is sufficiently specified, it should complete the task directly.

Feasibility Honesty. Tasks distinguish between requests for which a reliable outcome is unavailable and tasks that remain feasible. The former require the model to recognize and communicate the limitation, whereas the latter require it to continue rather than abstain merely because the task is difficult or unfamiliar.

Premise Scrutiny. Tasks distinguish invalid, missing, or contradictory premises from sound ones. The model must identify consequential premise defects when present, while avoiding unnecessary critique when the task is well formed.

Evidence Responsiveness. Multi-turn tasks distinguish unsupported pressure from genuinely stronger evidence. The model should maintain a supported conclusion under repetition, authority cues, or ungrounded challenges, but revise it when new evidence changes the task state.

The same principle is instantiated across multiple domains, and the same domain contains different behavioral

Source	Domain	Role in Runtime Alignment
DeepMath-103K	Mathematical reasoning	Supports task states involving missing conditions, feasibility, premise validity, and subsequent evidence
ACECODE-87K	Code	Supports task states involving incomplete requirements, implementation feasibility, invalid assumptions, and execution feedback
Fresh-Wiki QA	Grounded question answering	Supports answerable, clarification-seeking, and abstention-requiring evidence states

Table D.2: Primary sources and roles of the Persona Runtime Alignment data.

states. This reduces the extent to which the policy can associate a behavior with a topic alone. During Runtime Alignment, the policy receives only a neutral task instruction and the interaction history currently available to it. The Accountable-Professional specification and target high-level behavior are not included in the policy input. The policy must therefore infer the appropriate behavior from the task state itself.

The tasks are designed so that selecting an appropriate high-level behavior is not sufficient on its own. A clarification question must recover the information that is actually missing; identifying a defective premise must lead to a useful continuation; and revising a conclusion must remain consistent with the newly available evidence. The resulting task trajectories therefore provide feedback on both behavior selection and trajectory-level realization.

Only tasks with a clear task state, a valid interaction process, and reliable outcome feedback are retained. Model-visible context is separated from task-internal information used to maintain the environment and assess the completed trajectory. The final training package is additionally checked for format integrity, masking correctness, source lineage, and conflicts with the frozen evaluation data.

D.3 Fresh-Wiki QA

Fresh-Wiki QA is a Wikipedia-derived training source constructed for this work and is unrelated to the public benchmark FreshQA (Vu et al. 2024), which is not used for training. It mainly supports Information Sufficiency and Feasibility Honesty by creating different evidence states for otherwise similar factual questions.

Candidate pages are retrieved from English Wikipedia categories corresponding to 2025–2026 releases and entities. This temporal filtering is intended to reduce confounding from answers that the model can retrieve directly from parametric memory; it does not guarantee that the underlying facts were absent from pretraining. Fresh-Wiki QA consists of source-grounded, single-hop factual questions rather than multi-hop questions.

Each source record contains a question, an answer and its aliases, an authentic supporting sentence, the surrounding context, and source metadata. The answer must occur explicitly in the supporting sentence, and the supporting sentence must be an authentic substring of the corresponding Wikipedia text. Candidates without a reliable answer–evidence relation are removed rather than repaired with synthetic evidence.

Candidate questions undergo closed-book and open-context screening. Questions that can already be answered consistently without the source material are treated as overly easy or potentially affected by memorization. Questions that

Task state	Available evidence	Expected behavior
Answerable	The decisive supporting evidence is visible	Answer directly from the provided material
Clarification	Necessary information is initially hidden but can be obtained through interaction	Ask a targeted question and answer after the information is provided
Abstention	The decisive evidence is unavailable and cannot be obtained	State that the answer cannot be determined reliably rather than guess

Table D.3: Fresh-Wiki QA task states constructed by controlling evidence availability.

remain unreliable even when the complete evidence is supplied are treated as insufficiently clear. Retained records are also checked for answer–evidence consistency, source-lineage duplication, and textual collisions with evaluation data and other training sources.

For each retained fact, we control the availability of decisive evidence to construct three task states, summarized in Table D.3.

These states are created by programmatically retaining, hiding, removing, or releasing authentic source evidence; the underlying fact is not rewritten. The answer, aliases, and supporting sentence remain hidden from the policy and are used only to maintain the interaction and assess the completed task. Consequently, the appropriate behavior is determined by the evidence currently available to the policy rather than by the topic, surface form, or an explicit Persona prompt.

E Training Details

The main experiments use Qwen3-8B (Yang et al. 2025) as the backbone and proceed through Persona Representation Learning followed by Persona Runtime Alignment. Section B.1 applies the same two-stage procedure to Qwen3-14B. Both stages use full-parameter training without LoRA (Hu et al. 2022).

Stage I is trained for 2 epochs. Stage II is initialized from the final Stage-I checkpoint, with the optimizer and learning-rate scheduler reinitialized, and is trained for 1 epoch. All training runs are conducted on a single node with 8 NVIDIA H200 GPUs using BF16 precision and a maximum sequence length of 4,096 tokens.

Stage I uses fully sharded data parallelism (FSDP) with full sharding and gradient checkpointing. Stage II uses FSDP

Configuration	Persona Representation Learning	Persona Runtime Alignment
Epochs	2	1
Learning rate	1×10^{-5}	1×10^{-6}
Optimizer	AdamW	AdamW
Weight decay	0.1	0.01
Scheduler	Cosine	Constant
Gradient clipping	1.0	1.0
Rollouts per task	–	8
		Temperature 0.7,
Sampling	–	top- p 0.95
KL coefficient	–	0.02
Precision	BF16	BF16
Hardware	8×H200	8×H200

Table E.1: Training configurations for the two stages of R²A.

to manage the actor and reference models, with SGLang for rollout generation.

F Benchmark and Metric Details

F.1 Common Evaluation Setup

We evaluate ten benchmark families. Because QuestBench and ClarifyMT each contribute two separately reported subsets, the evaluation contains twelve settings in total. The benchmark families are QuestBench (Li, Kim, and Wang 2025), ClarifyMT (Luo et al. 2025), AbstentionBench (Kirichenko et al. 2025), XSTest (Röttger et al. 2024), OR-Bench (Cui et al. 2025), PCBench (Li et al. 2025), MiP (Fan et al. 2025), MathTrap (Zhao et al. 2024), SYCON-Bench (Hong et al. 2025), and FaithEval (Ming et al. 2025). All benchmark results are reported separately, and no cross-benchmark average is computed. The capabilities evaluated by each benchmark and their correspondence to the behavioral principles of the Accountable-Professional Persona are shown in Figure 1 of the main paper.

Within each model comparison, all conditions use the same evaluation instances, decoding configuration, and scoring procedure. Inference uses temperature 0, top- p 1, one completion per instance, and random seed 31415.

We use deterministic parsers or matchers when applicable. Benchmarks requiring semantic assessment are evaluated using a frozen label-only Judge. Within each comparison, all model conditions share the same Judge configuration, scoring prompt, and label definitions.

F.2 Benchmark Configurations

The primary n in Table F.1 denotes the denominator of the reported metric. Parenthetical totals additionally include paired controls or dialogue turns that are not necessarily included in the corresponding primary-metric denominator.

All metrics are reported as percentages except Mean Turn of Flip. Arrows indicate the direction of better performance.

F.3 Metric Definitions

AbstentionBench-fast. We treat abstain as the positive class and compute binary F1 separately within each score

group:

$$F1_g = \frac{2TP_g}{2TP_g + FP_g + FN_g}, \quad (S36)$$

where TP_g , FP_g , and FN_g are defined with respect to abstention in score group g . The reported score is the equally weighted mean across the 19 score groups:

$$\text{AbstentionF1} = \frac{1}{19} \sum_{g=1}^{19} F1_g. \quad (S37)$$

The metric is therefore a score-group macro average rather than a single F1 computed after pooling all 1,900 instances. Higher values are better.

QuestBench-GSM-Q. The released parser extracts the option selected by the model. GSM-Q accuracy is defined as

$$\text{GSMAcc} = \frac{N_{\text{correct option}}}{439}. \quad (S38)$$

Higher values indicate more accurate identification of the information required to complete the reasoning problem.

QuestBench-Logic-Q. The released matcher normalizes the model response and determines whether it matches the target logical attribute:

$$\text{LogicAcc} = \frac{N_{\text{normalized matches}}}{1,150}. \quad (S39)$$

Higher values indicate more accurate identification of the missing logical information.

ClarifyMT-Answer. For Answer-side instances, the appropriate initial action is to answer directly because the available information is sufficient. We parse the leading action of each response and compute

$$\text{AnswerDecisionAcc} = \frac{N_{\text{leading action=answer}}}{2,040}. \quad (S40)$$

This metric evaluates whether the model chooses to answer. It does not evaluate the correctness of the subsequent answer content.

ClarifyMT-Clarify. For Clarify-side instances, the appropriate initial action is to request additional information. Clarify-side decision accuracy is

$$\text{ClarifyDecisionAcc} = \frac{N_{\text{leading action=clarify}}}{4,080}. \quad (S41)$$

This metric evaluates whether the model chooses to clarify. It does not evaluate the quality of the generated clarification question.

PCBench-flawed/no-hint. Active Recognition Rate measures whether the model proactively identifies and challenges the relevant false premise without being explicitly instructed to inspect it:

$$\text{ARR} = \frac{N_{\text{recognized}}}{1,200}. \quad (S42)$$

Here, $N_{\text{recognized}}$ is the number of responses classified as identifying the relevant premise error. Higher values are better.

Benchmark and subset	Primary n	Scoring method	Reported metric
AbstentionBench-fast	1,900	Label-only Judge	Score-group mean abstention F1 $\uparrow$
QuestBench-GSM-Q	439	Released parser	GSM-Q accuracy $\uparrow$
QuestBench-Logic-Q	1,150	Released matcher	Logic-Q accuracy $\uparrow$
ClarifyMT-Answer	2,040	Deterministic leading-action parser	Answer-side decision accuracy $\uparrow$
ClarifyMT-Clarify	4,080	Deterministic leading-action parser	Clarify-side decision accuracy $\uparrow$
PCBench-flawed/no-hint	1,200	Label-only Judge	Active Recognition Rate $\uparrow$
MiP-missing	984 (1,618 total)	Label-only Judge	Missing-premise detection rate $\uparrow$
MathTrap-trap	104 (208 total)	Label-only Judge	Trap accuracy $\uparrow$
SYCON-Debate	100 topics (500 turns)	Label-only Judge	Mean Turn of Flip $\uparrow$
FaithEval-counterfactual	1,000	Normalized reference matcher	Counterfactual accuracy $\uparrow$
XSTest-safe	250 safe prompts (450 total)	Label-only Judge	Safe over-refusal rate $\downarrow$
OR-Bench-Hard	1,319	Label-only Judge	Rejection rate $\downarrow$

Table F.1: Benchmark configurations and reported metrics.

MiP-missing. Missing-premise detection rate measures whether the model recognizes that a necessary condition is absent:

$$\text{MiPDetection} = \frac{N_{\text{labeled insufficient}}}{984}. \quad (\text{S43})$$

The primary metric is computed over the 984 missing-premise instances. The corresponding 634 original instances are evaluated as paired controls and are not included in the primary denominator.

MathTrap-trap. Trap accuracy measures whether the response correctly identifies the annotated contradiction, missing condition, undefined condition, or other defect that prevents the problem from being solved as stated:

$$\text{TrapAcc} = \frac{N_{\text{correctly identified}}}{104}. \quad (\text{S44})$$

The primary metric is computed over 104 trap variants. Their 104 original counterparts are used as paired controls and are not included in the Trap accuracy denominator.

SYCON-Debate. Each topic contains five rounds of consecutive opposing feedback. For topic i , let

$$T_i \in \{0, 1, \dots, 5\} \quad (\text{S45})$$

denote the number of consecutive initial rounds for which the model remains aligned with its assigned stance. Counting stops when a response is first classified as neutral or opposing. Mean Turn of Flip is defined as

$$\text{MeanToF} = \frac{1}{100} \sum_{i=1}^{100} T_i. \quad (\text{S46})$$

The metric ranges from 0 to 5. Higher values indicate that the model is more likely to maintain its stance under unsupported opposing feedback.

FaithEval-counterfactual. Counterfactual accuracy measures whether the model answers according to the supplied counterfactual context rather than defaulting to its prior world knowledge. After normalizing the response and reference answers, we compute

$$\text{CounterfactualAcc} = \frac{N_{\text{reference matches}}}{1,000}. \quad (\text{S47})$$

A response is counted as correct when it matches the normalized reference answer or an accepted alias supported by the supplied context. Higher values are better.

XSTest-safe. Safe over-refusal rate measures how often the model refuses requests that are safe and answerable:

$$\text{SafeOverRefusal} = \frac{N_{\text{partial refusal}} + N_{\text{full refusal}}}{250}. \quad (\text{S48})$$

The primary metric is computed over the 250 safe prompts. The remaining 200 unsafe prompts serve as controls and are not included in this denominator. Lower values are better.

OR-Bench-Hard. Rejection rate measures how often the model refuses benign requests from the Hard subset:

$$\text{RejectionRate} = \frac{N_{\text{indirect refusal}} + N_{\text{direct refusal}}}{1,319}. \quad (\text{S49})$$

Both indirect and direct refusals are counted as rejections. Lower values are better.

F.4 Direction-Aligned Visualization

The result tables report all metrics in their original forms. Percentage-based metrics are reported as percentages, Mean Turn of Flip is reported on its original 0-to-5 scale, and the XSTest and OR-Bench metrics retain their lower-is-better directions.

For Figure 2 of the main paper, the metrics are converted to a common higher-is-better direction. Percentage values are first expressed as proportions. The three metrics requiring additional transformation are

$$\text{SYCON}_{\text{display}} = \frac{\text{MeanToF}}{5}, \quad (\text{S50})$$

$$\text{XSTest}_{\text{display}} = 1 - \text{SafeOverRefusal}, \quad (\text{S51})$$

$$\text{ORBench}_{\text{display}} = 1 - \text{RejectionRate}. \quad (\text{S52})$$

These transformations are used only for visualization. The result tables report the original metrics, and the transformed values are not used to compute a cross-benchmark aggregate.

F.5 Evaluation Scope

For each semantic-scoring benchmark, the Judge configuration, scoring prompt, and label definitions are fixed across all model conditions within the same comparison.

The MathTrap primary result covers the 104 trap variants in the evaluated paired set, with the corresponding original problems used as controls. The OR-Bench result covers the 1,319 benign prompts in the Hard subset. The reported scores therefore refer specifically to these evaluated subsets.

G Construction of the Static Persona Prompt Families

The Behavioral Frontier analysis in the main paper uses 165 static Persona conditions, consisting of 164 controlled prompt variants and the canonical Accountable-Professional prompt used in the main experiments. The canonical prompt is provided in Section A.1. The 164 variants comprise two prompt families, Static64 and Wide100. All variants retain the same Accountable-Professional role description and other fixed instructions, while varying only the formulations of Information Sufficiency, Feasibility Honesty, and Premise Scrutiny—the three principles examined in the Behavioral Frontier analysis.

G.1 Static64

Static64 assigns four intervention levels to each of the three principles and combines them using a full-factorial design:

$$4_Q \times 4_S \times 4_C = 64, \quad (\text{S53})$$

where Q , S , and C denote Information Sufficiency, Feasibility Honesty, and Premise Scrutiny, respectively. Each formulation specifies both a signature behavior and its complementary behavior. For example, the model should clarify when task-critical information is missing but answer directly when the available information is sufficient; state its limitations when a task cannot be completed reliably but proceed when it is feasible; and identify problematic premises while solving normally when the premises are sound. The levels therefore vary the applicability boundaries and intervention tendencies of each principle rather than simply adding or removing that principle.

G.2 Wide100

To cover a broader range of behaviors under static elicitation, Wide100 assigns five predefined formulations to each of Q , S , and C . These formulations range from completion-first and delayed intervention to balanced, broader, and more conservative intervention tendencies. The complete factorial design contains

$$5_Q \times 5_S \times 5_C = 125 \quad (\text{S54})$$

combinations. We retain 100 combinations using a fixed fractional-factorial design. Let

$$q, s, c \in \{0, \dots, 4\} \quad (\text{S55})$$

denote the formulation indices for the three principles. Combinations satisfying

$$c = (3(q + s) + 1) \bmod 5 \quad (\text{S56})$$

are excluded. In the resulting design, each formulation appears 20 times along each dimension, and every pairwise combination of formulations appears four times.

Together, the 64 Static64 conditions, the 100 Wide100 conditions, and the canonical Accountable-Professional prompt form the 165 static Persona conditions used in the main paper. In the frontier figure, the 164 controlled variants are grouped as prompt points and the canonical prompt is marked separately. Qwen3-8B and R²A are included only as external reference points and are excluded from the empirical Pareto-frontier construction.

References

- Cui, J.; Chiang, W.-L.; Stoica, I.; and Hsieh, C.-J. 2025. OR-Bench: An Over-Refusal Benchmark for Large Language Models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 11515–11542.
- Fan, C.; Li, M.; Sun, L.; and Zhou, T. 2025. Missing Premise Exacerbates Overthinking: Are Reasoning Models Losing Critical Thinking Skill? arXiv:2504.06514.
- He, Z.; Liang, T.; Xu, J.; Liu, Q.; Chen, X.; Wang, Y.; Song, L.; Yu, D.; Liang, Z.; Wang, W.; Zhang, Z.; Wang, R.; Tu, Z.; Mi, H.; and Yu, D. 2025. DeepMath-103K: A Large-Scale, Challenging, Decontaminated, and Verifiable Mathematical Dataset for Advancing Reasoning. arXiv:2504.11456.
- Hong, J.; Byun, G.; Kim, S.; and Shu, K. 2025. Measuring Sycophancy of Language Models in Multi-turn Dialogues. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2239–2259.
- Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- Kim, J.; Yang, N.; and Jung, K. 2025. Persona Is a Double-Edged Sword: Rethinking the Impact of Role-play Prompts in Zero-shot Reasoning Tasks. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, 848–862.
- Kirichenko, P.; Ibrahim, M.; Chaudhuri, K.; and Bell, S. J. 2025. AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions. arXiv:2506.09038.
- Kong, A.; Zhao, S.; Chen, H.; Li, Q.; Qin, Y.; Sun, R.; Zhou, X.; Wang, E.; and Dong, X. 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)*, 4099–4113.
- Li, B. Z.; Kim, B.; and Wang, Z. 2025. QuestBench: Can LLMs Ask the Right Question to Acquire Information in Reasoning Tasks? arXiv:2503.22674.
- Li, J.; Li, G.; Chang, Y.; and Wu, Y. 2025. Don’t Take the Premise for Granted: Evaluating the Premise Critique Ability of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 836–869.

- Luo, S.; Huang, Y.; Li, M.; Meng, S.; Liu, F.; Hu, Z.; Feng, J.; and Liu, Q. 2025. ClarifyMT-Bench: Benchmarking and Improving Multi-Turn Clarification for Conversational Large Language Models. arXiv:2512.21120.
- Marks, S.; Lindsey, J.; and Olah, C. 2026. The Persona Selection Model: Why AI Assistants Might Behave like Humans. Anthropic Alignment Science Blog, <https://alignment.anthropic.com/2026/psm/>. Accessed: 2026-07-29.
- Ming, Y.; Purushwalkam, S.; Pandit, S.; Ke, Z.; Nguyen, X.-P.; Xiong, C.; and Joty, S. 2025. FaithEval: Can Your Language Model Stay Faithful to Context, Even If “The Moon Is Made of Marshmallows”. In *The Thirteenth International Conference on Learning Representations*.
- Röttger, P.; Kirk, H.; Vidgen, B.; Attanasio, G.; Bianchi, F.; and Hovy, D. 2024. XSTest: A Test Suite for Identifying Exaggerated Safety Behaviours in Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 5377–5400.
- Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y. K.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300.
- Vu, T.; Iyyer, M.; Wang, X.; Constant, N.; Wei, J.; Wei, J.; Tar, C.; Sung, Y.-H.; Zhou, D.; Le, Q.; and Luong, T. 2024. FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, 13697–13720.
- Yang, A.; et al. 2025. Qwen3 Technical Report. arXiv:2505.09388.
- Zeng, H.; Jiang, D.; Wang, H.; Nie, P.; Chen, X.; and Chen, W. 2025. ACECODER: Acing Coder RL via Automated Test-Case Synthesis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12023–12040.
- Zhao, J.; Tong, J.; Mou, Y.; Zhang, M.; Zhang, Q.; and Huang, X. 2024. Exploring the Compositional Deficiency of Large Language Models in Mathematical Reasoning Through Trap Problems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 16361–16376.
- Zheng, M.; Pei, J.; Logeswaran, L.; Lee, M.; and Jurgens, D. 2024. When “A Helpful Assistant” Is Not Really Helpful: Personas in System Prompts Do Not Improve Performances of Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 15126–15154.