PersonaSteer: Multi-Turn Personalized Conversation in LLMs via Dynamic Activation Steering

Sijian Ren[*1] **Qingbin Zeng**[*1] **Bingqiao Gu**[3]
Fengli Xu[†1,2] **Yong Li**[1,2]
∗ Equal contribution † Corresponding author
[1]Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
[2]Zhongguancun Academy, Beijing, China
[3] Shandong University, Weihai, China
fenglixu@tsinghua.edu.cn

Abstract

Personalizing Large Language Models (LLMs) for long-term multi-turn interactions remains challenging, as user preferences and behavioral traits are progressively revealed and continuously evolving throughout conversations. Previous personalization methods have mostly relied on prompt-based or post-training (e.g., fine-tuning) paradigms; however, these approaches have been either prone to shallow alignment or have suffered from high computational costs. Inspired by the recent emergence of activation steering, we explore whether personalized alignment can be achieved through lightweight latent-space intervention without modifying backbone parameters. To achieve this goal, we propose **PersonaSteer**, a dynamic side-model architecture for multi-turn personalized alignment via learnable activation steering. Specifically, to enable fine-grained and evolving personalization, our design learns high-dimensional steering representations from unstructured interaction history and injects them into the backbone model's residual stream for latent-space behavioral modulation. Furthermore, we introduce a steering-vector feedback loop to maintain persona consistency across multi-turn conversations. Extensive experiments demonstrate that PersonaSteer achieves best or top-tier alignment performance, averaging 3.18/5 on Qwen2.5-7B and 2.95/5 on Qwen2.3-3B. In addition, our framework approach maintains stability during evolving dialogues, with alignment performance consistently improving over time. PersonaSteer provides a lightweight solution for building truly dynamic and personalized LLM. The code is available in https://anonymous.4open.science/r/PersonaSteer-F840/.

1 Introduction

The proliferation of Large Language Models (LLMs) has fundamentally transformed user interactions (Guo et al., 2025; Zhao et al., 2023). However, the prevailing "one-size-fits-all" alignment paradigm struggles to accommodate the heterogeneous and nuanced demands of individual users (Liu et al., 2025). Achieving deep personalization, where models faithfully reflect stable user preferences while continuously adapting to evolving behavioral patterns across multi-turn interactions, is essential for interactive dialogue systems and personal assistants (Christakopoulou et al., 2023; Salemi et al., 2024). Such systems are required to dynamically model the user profiles and provide personalized responses for long-term human-AI collaboration.

Despite the growing demand for tailored AI experiences, constructing personalized LLMs for long-term multi-turn interactions remains technically challenging. Existing approaches primarily operate in either the input space or the parameter space; however, both struggle to balance personalization quality, computational efficiency, and temporal adaptability. **Prompt-based methods** personalize models via the input space, such as by leveraging RAG to recall historical memories. Despite their flexibility, these techniques are often hindered by noisy or diluted contextual signals as conversations grow longer, limiting stable persona consistency and the overall quality of personalization. **Post-training methods**(Shao et al., 2023; Tan et al., 2024b), ranging from preference alignment to specialized fine-tuning (e.g., SFT), embed personalization into model parameters through continual optimization. However, such approaches are computationally expensive and fundamentally constrained by static parameter spaces, limiting their ability to dynamically adapt to user behaviors and shifting preferences during long-term interactions.

Recently, **activation steering**(Subramani et al., 2022; Turner et al., 2024) has emerged as a lightweight and flexible solution for deep-level intervention within the model's internal representations. However, current steering techniques are typ-

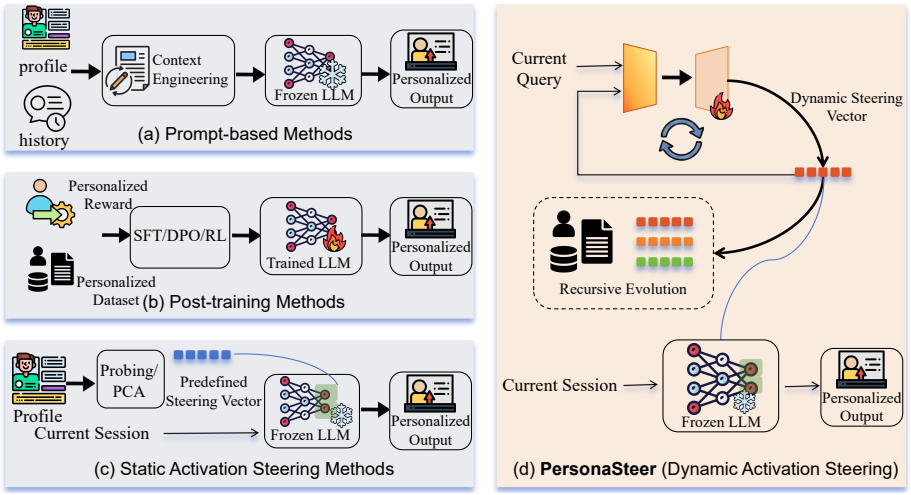

Figure 1: Comparison of personalized alignment paradigms.

ically fixed to limited dimensional subspaces (e.g. the Big Five personality) or predefined semantic directions extracted via techniques such as probing and principal component analysis (PCA)(Kim et al., 2025; Heo et al., 2025; Zhu et al., 2024), restricting their ability to capture fine-grained and evolving user characteristics. While recent attempts like CHAMELEON(Zhang et al., 2025) have sought to automate this process, they remain insufficient in precision and lack a robust mechanism for **multi-round persona evolution**. Consequently, existing models struggle to sustain character consistency in extended dialogues, failing to adapt as the user interaction progresses.

To address these limitations, we propose **PersonaSteer**, a lightweight personalization framework utilizing dynamic activation steering to achieve fine-grained and evolving personalized alignment. The framework introduces a **lightweight and trainable side-network** operating alongside a frozen backbone LLM to dynamically generate steering vectors from user queries and historical interaction states. We inject these steering vectors into the model's residual stream, enabling latent-space behavioral modulation without modifying backbone parameters. We optimize the side-network module through a Joint Multi-Task Learning objective that combines standard Supervised Fine-Tuning (SFT) with Supervised Contrastive Learning (SCL) to enforce persona discriminability. Furthermore, to address persona drift in multi-turn conversations, we introduce a steering-vector feedback loop that recursively incorporates historical steering states into subsequent interactions. This mechanism enables PersonaSteer to maintain stable persona consistency while continuously adapting to dynamic user behaviors throughout extended dialogues.

We conduct extensive experiments on the ALOE benchmarks (Wu et al., 2025), where PersonaSteer outperforms prompting-based methods and prior activation-steering baselines. Using Qwen2.5-3B as a representative backbone, our framework achieves a 13.9% enhancement in alignment level compared to complex agentic workflow methods. More notably, it outperforms previous activation steering baselines by 46.8%. Despite operating as a lightweight side-network without modifying backbone parameters, PersonaSteer achieves performance competitive with post-training approaches at lower computational cost. Further analysis reveals that our method maintains a stable and progressively evolving alignment level, exhibiting superior robustness against persona decay compared to existing approaches. Ablation studies further confirm the necessity of our steering vector feedback loop and contrastive learning loss.

In summary, our major contributions in this work include:

- We propose a novel personalization framework based on *learnable activation steering*. Unlike conventional activation steering methods confined to fixed semantic directions, our approach intervenes in a broader latent space, enabling the model to capture complex personal traits.
- To ensure character consistency across extended interactions, we introduce a **steering vector feedback loop**. By treating intervention vectors as dynamic memory states, this mechanism facilitates multi-round persona evolution.

- We introduce a **Supervised Contrastive Loss** into the steering vector optimization process. By pulling steering vectors of the same persona closer and pushing those of different personas apart, our framework ensures high *persona discriminability* within the latent space.
- We demonstrate the efficacy of our framework through extensive evaluations on the ALOE benchmark. Experimental results show that PersonaSteer significantly outperforms existing activation steering baselines by 46.8% and achieves a 13.9% improvement over agentic workflows, providing a robust, stable, and cost-effective solution for preference tracking alignment.

2 Related Work

2.1 LLM Personalization

The pursuit of personalized LLMs has evolved across several paradigms. *Prompt-based methods* leverage in-context learning, with frameworks such as *LaMP* (Salemi et al., 2024) and *Personalized-LLM* (Tan et al., 2024a,b) utilizing *RAG* to inject historical user profiles. Extensions like *MemGPT* (Packer et al., 2024) and *LD-Agent* (Li et al., 2025a) manage contexts as external memory. However, growing personalized context dilutes LLMs with irrelevant noise, degrading persona adherence in long-form generation or multi-turn dialogues (Liu et al., 2023). *Fine-tuning methods* internalize user traits via *SFT* (Ouyang et al., 2022) and *DPO* (Rafailov et al., 2024), with *PEFT* techniques like LoRA (Hu et al., 2021) adopted for efficient adaptation (Shao et al., 2023)(Tan et al., 2024b). RLPA (Zhao et al., 2025b) further introduces reinforcement learning for dynamic alignment. Yet these methods remain computationally prohibitive, and weight-based updates yield **static model states** lacking *temporal agility* to assimilate real-time behavioral shifts without recurring training.

2.2 Activation Steering

To bridge the gap between training-heavy and context-heavy methods, *activation steering* (Subramani et al., 2022; Turner et al., 2024) has emerged as a promising middle ground. This paradigm is fundamentally rooted in the **Linear Representation Hypothesis (LRH)** (Park et al., 2023)(Elhage et al., 2022), which posits that high-level concepts are encoded as linear directions within the latent space. Recent research has leveraged this to achieve **Personality Alignment**; for instance, *PAS* (Zhu et al., 2024) utilizes personality inventories and the PAPI dataset to optimize activation interventions, achieving alignment with traits like the Big Five. Building on this, *CHAMELEON* (Zhang et al., 2025) explored automated steering to induce persona-specific behaviors without manual vector crafting. However, these methods primarily focus on static persona traits and remain limited by their reliance on low-dimensional semantic directions, lacking the multi-turn memory feedback loops necessary for sustained character consistency in evolving dialogues. Our work extends this paradigm into a dynamic, memory-augmented framework capable of tracking shifting behavioral states.

3 Methodology

3.1 Overview of PersonaSteer Framework

The PersonaSteer framework operates on the principle of latent space intervention. As illustrated in the provided implementation, the system consists of a frozen backbone LLM (θ_{LLM}) and a trainable side-network (ϕ). Given a sequence of user turns $\{u_1, u_2, \ldots, u_n\}$, the side-network generates a sequence of steering vectors $\{v_1, v_2, \ldots, v_n\}$ that are injected into the residual stream(Lieberum et al., 2024; Templeton, 2024) of a specific transformer layer. The optimization objective is formulated as a **Joint Multi-Task Learning** problem:

(i) **Generative Alignment (SFT):** To ensure the model generates personalized content, we minimize the negative log-likelihood:

$$\mathcal{L}_{SFT} = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, \mathbf{u}, \hat{z}^{(l)}) \quad (1)$$

where $\hat{z}^{(l)}$ represents the modified activations at layer l.

(ii) **Persona Discriminability:** To ensure that steering vectors capture unique persona traits, we employ a **Supervised Contrastive Loss (SCL)**. For a batch of steering vectors, let i be the index of a sample and $P(i)$ be the set of indices of all samples sharing the same persona label. The loss is defined as:

$$\mathcal{L}_{\text{SCL}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{e^{\text{sim}(v_i, v_p)/\tau}}{\sum_{a \neq i} e^{\text{sim}(v_i, v_a)/\tau}} \quad (2)$$

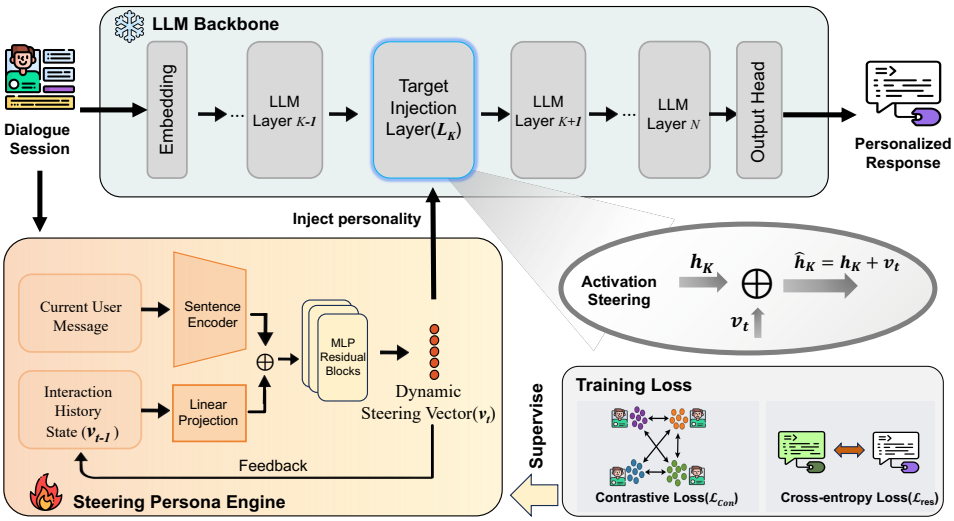

Figure 2: Overview of Our Proposed PersonaSteer Framework.

where $\text{sim}(\cdot)$ denotes cosine similarity and τ is a temperature hyperparameter. The total training objective is $\mathcal{L}_{total} = \mathcal{L}_{SFT} + \alpha\mathcal{L}_{SCL}$, where α is the contrastive weight.

3.2 Lightweight Side-Network Architecture

To facilitate deep behavioral intervention without perturbing the foundational knowledge of the LLM, we introduce a **decoupled latent steering generator**, denoted as ϕ. The core philosophy of this architecture is to treat personalization as a **non-intrusive modulation** signal, mapping heterogeneous user behavioral signals into the high-dimensional activation space of the backbone LLM.

3.2.1 Semantic Feature Extraction

The process begins by distilling the current user utterance u_t through a frozen, high-capacity semantic encoder $\mathcal{E}(\cdot)$. Specifically, we utilize a pre-trained transformer-based bi-encoder to project discrete textual tokens into a stable embedding space $\mathbb{R}^{d_{enc}}$. This design choice is strategic: by leveraging an external encoder, the side-network inherits robust, task-agnostic linguistic priors. This allows the framework to remain **parameter-efficient**, as the trainable components are focused exclusively on persona-specific steering rather than basic language modeling. Consequently, the total trainable parameter count is kept orders of magnitude smaller than that of the backbone LLM, ensuring scalability in real-world deployments.

3.2.2 Cross-Model Activation Alignment

A fundamental challenge in such a decoupled design is the **dual misalignment**, both in dimensionality and semantic density between the source encoder space $\mathbb{R}^{d_{enc}}$ and the target latent space of the LLM, $\mathbb{R}^{d_{model}}$. While the encoder space captures broad thematic features, the LLM's internal residual stream operates on highly specialized causal trajectories.

To bridge this structural gap, we propose a **Learnable Activation Mapping** module based on a deep Multi-layer Perceptron (MLP) architecture. This module serves as a bridge that re-projects the static semantic vector e_t into a dynamic steering candidate v_t:

$$v_t = \mathcal{P}(e_t), \tag{3}$$

where $\mathcal{P}$ denotes the learnable projection from the encoder space $\mathbb{R}^{d_{enc}}$ to the steering space $\mathbb{R}^{d_{model}}$. Equation (3) only illustrates this cross-model projection; the complete side-network architecture is provided in Appendix B. Beyond mere dimensionality reduction or expansion, this projection acts as an **information filter** that distills task-specific persona features while suppressing irrelevant linguistic noise. By enforcing this activation alignment, we ensure that the generated intervention z_t is not only compatible with the backbone's numerical range but also aligns with the underlying geometric structure of the LLM's internal representations. This high-fidelity mapping is the key to achieving precise, fine-grained control over the model's generative behavior.

3.3 Recursive State Internalization for Multi-round Evolution

A pivotal innovation within the **PersonaSteer** framework is the **Recursive State Internalization**

(RSI) mechanism. Conventional activation steering methodologies are predominantly *memoryless*, treating each dialogue turn as an isolated event. Such an ephemeral approach often leads to the **persona evaporation problem**, where the model's behavioral consistency progressively decays as the conversation depth increases. To mitigate this, we reformulate steering as a **stateful sequential transition** task.

3.3.1 Modeling Persona Trajectories

In our framework, the side-network ϕ is architected to maintain a latent hidden state h_t, which functions as a **dynamic persona repository**. This state captures the cumulative trajectory of user preferences and linguistic nuances across multi-round interactions. Formally, the transition is defined as:

$$\begin{aligned} v_t &= \phi(u_t, h_{t-1}), \\ h_{t-1} &= \text{SiLU}\left(\text{LayerNorm}\left(W_h v_{t-1} + \mathbf{b_h}\right)\right) \end{aligned} \tag{4}$$

where h_{t-1} encodes the historical context of the persona, and v_t is the turn-specific intervention vector, and W_h and $\mathbf{b_h}$ are the weight matrix and bias of the linear down-projection layer, respectively.

3.3.2 Temporal Decoupling and Gradient Stability

To ensure the feasibility of training over long-range sequences, we implement a **Gradient-Detached Recurrence** strategy. During the forward pass, the steering vector from the preceding turn v_{t-1} is detached from the current computational graph before being fed back into the side-network for the subsequent turn:

$$v_t = \phi(u_t, \text{detach}(h_{t-1})) \tag{5}$$

This design choice serves a dual purpose. First, it effectively facilitates **multi-round persona internalization**, allowing historical traits to inform current steering without the computational burden of Backpropagation Through Time (BPTT). Second, it acts as a **regularization bottleneck** that prevents the accumulation of catastrophic gradient noise across rounds, ensuring that the persona remains anchored to the user's core identity even in highly diverse conversational contexts. Consequently, this recursive internalization enables a robust and stable evolution of the model's adaptive behavior.

3.4 End-to-End Steering Vector Injection

The transition from latent persona representation to behavioral change is facilitated by our **End-to-End Residual Injection** mechanism. PersonaSteer operates directly within the LLM's **residual stream**.

The final generated steering vector v_t is integrated into the frozen backbone model via a dedicated **Residual Injector** situated at a strategic bottleneck layer l (e.g., mid-to-late transformer blocks). Let $z^{(l)} \in \mathbb{R}^{d_{model}}$ denote the original hidden activations of the backbone model at layer l. The injection process re-defines the forward pass computation as:

$$\hat{z}^{(l)} = z^{(l)} + v_t \tag{6}$$

This additive coupling is mathematically equivalent to a **directional nudge** within the high-dimensional latent space. By applying this intervention specifically **post-attention**, we ensure that the steering signal modulates the context-aware representations before they are projected into the subsequent feed-forward networks (FFN) and final layers.

4 Experiment

4.1 Experimental Setup

4.1.1 Benchmark and Evaluation Scenarios

To rigorously evaluate the efficacy of *PersonaSteer* in capturing complex and evolving user dynamics, we utilize the Align with Customized Preferences(**ALOE**) (Wu et al., 2025). This benchmark is specifically designed to simulate real-world multi-turn interactions. Following RLPA(Zhao et al., 2025b), we test our model in two challenging scenarios:

Multi-turn Alignment (Vanilla ALOE): We evaluate the model using unseen profiles within the training domain. This scenario assesses the framework's ability to elicit stable alignment.

Out-of-distribution Generalization (Extended ALOE): We utilize profiles from PersonaChat(Zhang et al., 2025). This evaluates the framework's capacity to internalize and evolve behavioral states during interactions with novel persona, without relying on any pre-defined schemas.

4.1.2 Baselines

We compare our approach against three categories of methods:

- **Prompt-based Method:** Including *Instruct LLM* (zero-shot), *RAG*(Zhao et al., 2025a), *Chain-of-Thought (CoT)*(Wei et al., 2022), *Mem0*(Chhikara et al., 2025), and *MemoryOS*(Li et al., 2025b) for personalized reasoning.

		ALOE		Extend ALOE		Computational Complexity
	Qwen2.5-7B	Align.Score ↑	N-IR ↑	Align.Score ↑	N-IR ↑	Training (10^{17} FLOPs) ↓
Base	Qwen2.5-7B-Instruct	2.02	0.0430	2.01	0.0509	-
Agentic Workflow	RAG(Zhao et al., 2025a)	2.74	-0.0672	2.57	-0.2600	-
	CoT(Wei et al., 2022)	3.00	0.0258	2.79	0.0600	-
	Mem0(Chhikara et al., 2025)	2.02	0.0693	2.08	-0.0150	-
	MemoryOS(Li et al., 2025b)	2.83	-0.0581	2.76	0.0350	-
Activation Steering	PAS(Zhu et al., 2024)	2.05	0.0662	2.00	0.0488	-
	CHAMELEON(Zhang et al., 2025)	2.07	0.0523	2.01	0.0488	-
Post-training	AIPI(SFT)(Wu et al., 2025)	<u>3.16</u>	0.0764	2.92	0.0767	<u>5.57</u>
	AIPI(DPO)(Wu et al., 2025)	2.89	-0.0132	2.50	0.0030	16.6
	RLPA(Zhao et al., 2025b)	3.08	<u>0.0923</u>	<u>3.02</u>	<u>0.0876</u>	27.5
	PersonaSteer	**3.18**	**0.1084**	**3.02**	**0.0922**	**3.69**
		ALOE		Extend ALOE		Computational Complexity
	Qwen2.5-3B	Align.Score ↑	N-IR ↑	Align.Score ↑	N-IR ↑	Training (10^{17} FLOPs) ↓
Base	Qwen2.5-3B-Instruct	1.97	0.0741	1.98	0.0451	-
Agentic Workflow	RAG(Zhao et al., 2025a)	2.55	-0.0958	2.14	-0.0899	-
	CoT(Wei et al., 2022)	2.59	0.0341	2.32	0.0054	-
	Mem0(Chhikara et al., 2025)	2.26	-0.0353	2.10	-0.0418	-
	MemoryOS(Li et al., 2025b)	2.60	-0.0769	2.58	-0.1012	-
Activation Steering	PAS(Zhu et al., 2024)	2.01	0.0710	1.96	0.0639	-
	CHAMELEON(Zhang et al., 2025)	2.00	0.0283	2.01	0.0443	-
Post-training	AIPI(SFT)(Wu et al., 2025)	2.78	<u>0.1068</u>	<u>2.67</u>	0.0767	<u>2.48</u>
	AIPI(DPO)(Wu et al., 2025)	<u>2.88</u>	-0.0822	2.49	-0.0945	4.35
	RLPA(Zhao et al., 2025b)	2.59	0.0915	2.55	<u>0.0823</u>	17.1
	PersonaSteer	**2.95**	**0.1147**	**2.74**	**0.1019**	**1.64**

Table 1: Comparison of different methods on ALOE and Extend ALOE datasets. Training Computational Complexity unit: 10^{17} FLOPs. ↑ indicates higher is better, ↓ indicates lower is better.

- **Post-training Alignment:** We evaluate *AIPI (SFT)*, *AIPI (DPO)* (Wu et al., 2025), and *RLPA*(Zhao et al., 2025b), which represent the current standard for parameter-space preference optimization.
- **Activation Steering:** Including *CHAMELEON* (Zhang et al., 2025) and *PAS*(Zhu et al., 2024) for personalized steering.

Please refer to the Appendix A.4 for the detailed configurations and implementation specifics of all baseline methods.

4.1.3 Evaluation Metrics

Following the ALOE protocol, we adopt the **Average Alignment Score (Align.Score)** to measure the fit between the response and the persona. To ensure a scale-invariant comparison, we report the **Normalized Improvement Ratio (N-IR)**. N-IR quantifies the model's ability to progressively improve its alignment over multiple turns. The detailed calculation procedures for these metrics are provided in Appendix A.2.

4.1.4 Implementation Details

We implement our framework using **Qwen2.5-3B-Instruct** ($L = 36$ layers) and **Qwen2.5-7B-Instruct** ($L = 28$ layers)(Qwen et al., 2025) as backbones. For the semantic encoding of user inputs, we utilize the **bge-base-en-v1.5**(Xiao et al., 2024) encoder.

Training Protocol. We train on a single **A100-80GB** using AdamW ($lr = 1 \times 10^{-4}$, linear decay, 1 epoch, batch size 4), with $\alpha_{cl} = 0.2$. For the RLPA baseline, we utilize two A100 GPUs to accommodate its higher memory requirements.

Injection Layer Selection. A critical hyperparameter in *PersonaSteer* is the injection layer l_{inj}. We select $l_{inj} = 21$ for the 7B model and $l_{inj} = 28$ for the 3B model based on our experimental validation. This positioning allows the injected persona to leverage the backbone's deep semantic representations while leaving sufficient remaining layers for high-level stylistic synthesis.

4.2 Main Results

Table 1 presents the quantitative comparison of PersonaSteer against diverse baselines across two model scales. Our findings highlight two key observations regarding alignment and computational economy.

4.2.1 Superiority in Persona Alignment

Across both **ALOE** (In-distribution) and **Extend ALOE** (OOD) benchmarks, *PersonaSteer* achieves top-tier alignment scores comparable to resource-intensive post-training methods, while offering the most stable improvement trajectory (highest N-IR) and lowest computational cost.

Comparison with Prompt-based Method While RAG and CoT improve the base model's performance, they exhibit significant fluctuations in $N-IR$, occasionally yielding negative values (e.g., -0.0958 for RAG on Qwen2.5-3B). This underscores that input-space modulations are highly sensitive to historical noise and context dilution. In contrast, *PersonaSteer* maintains a robust $N-IR$ (0.1147 on 3B), demonstrating that intervening in the activation space provides a more stable anchor for persona alignment.

Comparison with Optimization-based Methods Compared to **RLPA**, which represents a strong dynamic alignment baseline, *PersonaSteer* matches or surpasses its alignment quality (*Align.Score*: 3.18 vs. 3.08 on 7B) while maintaining superior $N-IR$ scores (0.1084 vs. 0.0923). This suggests that our steering vectors can capture nuanced behavioral traits with a precision comparable to, or even exceeding, explicit profile-based RL fine-tuning.

The goodness of fit for the normalized alignment trend is analyzed in Appendix H. We refer the reader to Appendix C for the full sensitivity study on injection layer and steering magnitude.

4.2.2 Computational Efficiency

A defining advantage of *PersonaSteer* lies in its training efficiency, particularly when compared to the intensive resource demands of reinforcement learning-based alignment frameworks. As evidenced in Table 1, *PersonaSteer* consumes only 3.69×10^{17} FLOPs during the training phase on the Qwen2.5-7B scale. In stark contrast, **RLPA** (27.5×10^{17} FLOPs) is nearly $7.5\times$ more computationally expensive. This gap stems from the inherent complexity of the PPO pipeline, which requires multiple forward-backward passes, extensive response sampling, and dual-level reward structures with external LLM judges (e.g., GPT-4o-mini).

Our method trains only a decoupled side-network with a standard contrastive objective, avoiding multi-stage optimization required by RLPA. This efficiency generalizes across model scales: on the 3B backbone, *PersonaSteer* further lowers training cost to 1.64×10^{17} FLOPs while still achieving a superior *Align. Score* of 2.95. These results suggest that *PersonaSteer* delivers high-fidelity alignment, and enables personalized fine-tuning on single-GPU setups (e.g., A100-80GB), whereas RLPA often relies on multi-GPU training to sustain its optimization pipeline.

Configuration	ALOE				Extended ALOE			
	Qwen2.5-7B		Qwen2.5-3B		Qwen2.5-7B		Qwen2.5-3B	
	Score	N-IR	Score	N-IR	Score	N-IR	Score	N-IR
Full (Ours)	**3.18**	**0.1084**	**2.95**	**0.1147**	**3.02**	**0.0922**	**2.74**	**0.1019**
w/o CL	2.83	0.1068	2.77	0.1092	2.64	0.0364	2.67	0.0899
w/o FL	2.72	0.0944	2.60	0.1084	2.58	0.0844	2.48	0.0483
Base Model	2.02	0.0430	1.97	0.0741	2.01	0.0509	1.98	0.0451

* Score denotes Align.Score.

Table 2: Ablation Study Results. CL denotes Contrastive Learning, and FL denotes Feedback Loop.

4.3 Ablation Study

To rigorously evaluate the contribution of each component within the PersonaSteer framework, we conduct an ablation study on ALOE and Extended ALOE using Qwen2.5-7B and Qwen2.5-3B backbones. We compare the full model against two variants: (1) **w/o Contrastive Learning**, where the model is trained solely with standard Cross-Entropy (CE); and (2) **w/o Feedback Loop**, where the recursive injection of the previous turn's residual vector is disabled, reverting the system to a static steering mechanism. The results are summarized in Table 2.

4.3.1 Impact of Feedback Loop Mechanism

The feedback loop is fundamental to our dynamic steering paradigm. As shown in Table 2, removing the feedback loop causes substantial degradation. For the 7B model on the ALOE dataset, the Average Alignment Level (AL) drops from 3.18 to 2.72. More critically, on the Extended ALOE dataset, the Normalized Improvement Rate (N-IR) for the 3B model drops significantly to 0.0483 (compared to 0.1019 for the full model). This indicates that without the recursive state tracking, the model fails to accumulate persona-related context over multi-turn interactions, effectively losing the ability to maintain consistency as the dialogue progresses. The feedback loop is thus essential for transforming static activation steering into a dynamic, context-aware process.

4.3.2 Significance of Contrastive Learning

The ablation of contrastive learning (*w/o Contrastive Learning*) reveals its critical role in sta-

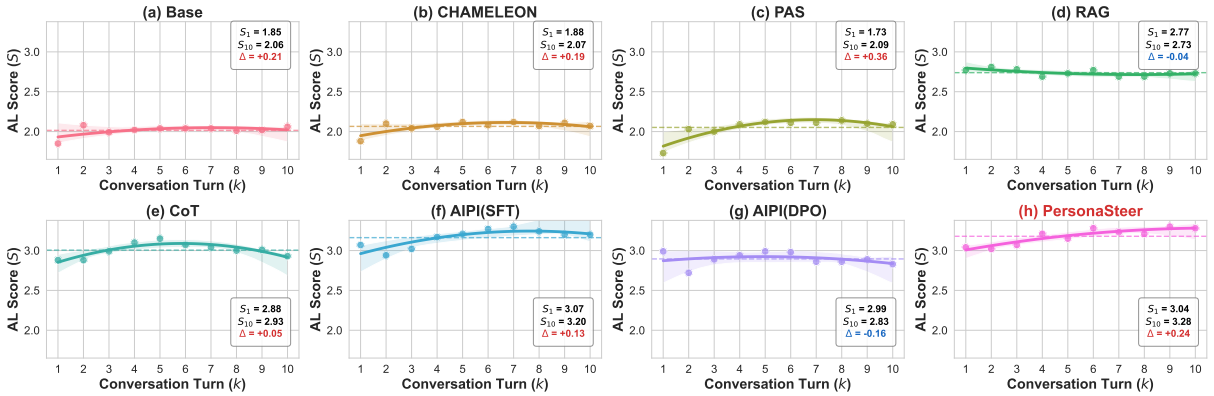

Figure 3: Turn-wise alignment analysis of various methods on Qwen2.5-7B (ALOE). The dashed horizontal line represents the average alignment score (Mean) for each method. The solid curve and shaded area denote the quadratic regression trend and the 95% confidence interval, respectively.

bilizing the steering trajectory. Without contrastive learning, AL score drops to 2.83 (7B, ALOE) and 2.64 (7B, Extended ALOE). The model struggles to distinguish high-quality persona expressions without structural constraints, drifting from the target profile. These results confirm that contrastive learning effectively structures the latent steering space, ensuring that the generated vectors consistently push the model towards more aligned behaviors.

In summary, the full PersonaSteer framework achieves the best performance by synergizing these components: the feedback loop ensures temporal consistency, while contrastive learning guarantees directional accuracy in the latent space. Additional ablations on encoder variants, history formulation, and recurrence strategy are provided in Appendix F.

4.4 Visualization of Multi-Turn Alignment Evolution

To intuitively illustrate how persona alignment evolves during interactions, we visualize the turn-wise alignment scores for Qwen2.5-7B on the ALOE benchmark in Figure 3. Due to space constraints, visualization results for other model scales (3B) and scenarios (Extended ALOE) are provided in Appendix (Figure 4, Figure 5, and Figure 6).

As illustrated in Figure 3, distinct behavioral trajectories emerge across different paradigms. The RAG-enhanced baseline (d) achieves high initial alignment but suffers rapid decay as conversation progresses. This downward trend empirically confirms the "context dilution" effect, where retrieved demonstrations in the input space are increasingly overwhelmed by the growing history of conversational tokens. In contrast, while optimization-based methods like AIPI (f) exhibit a progressive upward trajectory, their lower starting scores and initial fluctuations reflect the inherent latency and instability.

The *PersonaSteer* (h) framework effectively addresses these challenges, maintaining a superior alignment level (Mean = 3.18) with lower fluctuation. Unlike prompt-based methods relying on static demonstrations, *PersonaSteer* leverages recursive feedback to dynamically inject steering residuals from previous turns into the latent space. This continuous calibration alleviates semantic drift, yielding a stable curve consistently above all baselines. The results demonstrate that activation-space intervention with temporal state-tracking provides more robust control than input-space modulation or standard preference optimization.

5 Conclusion

We present *PersonaSteer*, a lightweight and decoupled activation steering framework designed for fine-grained, evolving personalized alignment in Large Language Models. By decoupling persona control from base model parameters, our approach enables real-time behavioral adaptation through lightweight side-network intervention. The steering vector feedback mechanism ensures consistent persona alignment across multi-turn interactions. Empirical results demonstrate that *PersonaSteer* achieves top-tier alignment scores comparable to resource-intensive training paradigms at a fraction of the computational cost, while substantially outperforming prompting-based methods and prior activation-steering baselines.

This work establishes activation steering as a viable foundation for practical personalization sys-

tems, offering a scalable path toward truly adaptive AI companions.

Limitations

Although PersonaSteer demonstrates strong performance in dynamic personalized alignment, several limitations remain. First, while the recursive feedback mechanism improves multi-turn consistency, it does not constitute a fully persistent long-term memory system, and persona drift may still accumulate in substantially longer interactions. Second, our evaluation primarily relies on LLM-as-a-judge protocols following the ALOE benchmark, which may introduce potential evaluation bias. To reduce this effect, we adopt standardized multi-turn evaluation prompts and additionally incorporate human evaluation to verify the consistency of the observed alignment trends. Details are provided in Appendix D. Future work will further explore persistent memory integration and more comprehensive human-centered evaluation protocols.

Acknowledgment

This work is sponsored by the National Key Research and Development Program of China (xxxx) and the National Natural Science Foundation of China (xxxx).This work is in part supported by Tsinghua-Toyota Joint Research Center.

References

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Konstantina Christakopoulou, Alberto Lalama, Cj Adams, Iris Qu, Yifat Amir, Samer Chucri, Pierce Vollucci, Fabio Soldo, Dina Bseiso, Sarah Scodel, and 1 others. 2023. Large language models for user interest journeys. *arXiv preprint arXiv:2305.15498*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Juyeon Heo, Christina Heinze-Deml, Oussama Elachqar, Kwan Ho Ryan Chan, Shirley You Ren, Andrew Miller, Udhyakumar Nallasamy, and Jaya Narain. 2025. Do llms"know"internally when they follow instructions? In *The Thirteenth International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Junsol Kim, James Evans, and Aaron Schein. 2025. Linear representations of political perspective emerge in large language models. In *The Thirteenth International Conference on Learning Representations*.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2025a. Hello again! LLM-powered personalized agent for long-term dialogue. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5259–5276, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, and 1 others. 2025b. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724*.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *Preprint*, arXiv:2408.05147.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. 2025. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

OpenAI. 2025. o3 and o3-mini system card. Technical report, OpenAI. Accessed: 2026-02-08.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *Preprint*, arXiv:2310.10158.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *CoRR*, abs/2004.09297.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. 2022. Extracting latent steering vectors from pretrained language models. *Preprint*, arXiv:2205.05124.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6459–6475, Miami, Florida, USA. Association for Computational Linguistics.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024b. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6476–6491, Miami, Florida, USA. Association for Computational Linguistics.

Adly Templeton. 2024. *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shujin Wu, Yi R Fung, Cheng Qian, Jeonghwan Kim, Dilek Hakkani-Tur, and Heng Ji. 2025. Aligning llms with individual preferences via interaction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7648–7662.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yijing Zhang, Dyah Adila, Changho Shin, and Frederic Sala. 2025. Personalize your LLM: Fake it then align it. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7287–7301, Albuquerque, New Mexico. Association for Computational Linguistics.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025a. Do llms recognize your preferences? evaluating personalized preference following in llms. *arXiv preprint arXiv:2502.09597*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Weixiang Zhao, Xingyu Sui, Yulin Hu, Jiahe Guo, Haixiao Liu, Biye Li, Yanyan Zhao, Bing Qin, and Ting Liu. 2025b. Teaching language models to evolve with users: Dynamic profile modeling for personalized alignment. *arXiv preprint arXiv:2505.15456*.

Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. 2024. Personality alignment of large language models. *arXiv preprint arXiv:2408.11779*.

A Experimental Details: Prompts and Configurations

A.1 Evaluation Protocol Overview

We report results on two evaluation settings aligned with the ALOE benchmark: (i) **Vanilla ALOE** (ID), where evaluation instances are formed by crossing a profile pool with a personality-trait pool; and (ii) **Extended ALOE** (OOD), where each instance provides only a profile (e.g., PersonaChat-style personas), without an explicit personality label.

Interactive setup. For each evaluation instance, we run **10 rounds** of conversation. The user side is simulated by a strong LLM (GPT-4o) instructed to role-play a real netizen with *progressive information disclosure* (i.e., revealing profile details gradually as the dialogue unfolds). The tested method (target model) responds each round conditioned on the conversation history and the current user message.

On-the-fly judging. After each target response, a separate GPT-4o judge scores the response on a **1–5** scale. We then aggregate turn-level scores into the metrics shown in Table 1.

Important note (reproducibility). We intentionally omit
deployment-specific information such as API endpoints, ports, and keys. The appendix documents *prompt templates* and *algorithmic configurations* only.

A.2 Metrics

Let $AL(t)$ denote the average judge score at round t (averaged over all sessions for a method). We compute:

- **AL(K)AVG** (as reported in Table 1): the average score across all rounds (K=10 in our experiments), i.e., $\frac{1}{K}\sum_{t=1}^{K} AL(t)$.
- **N-IR** (as reported in Table 1): the *normalized* improvement rate, i.e., the slope of a linear regression fitted on the per-round scores after min–max normalization within each session/method:

$$\text{N-IR} = \underset{b,a}{\operatorname{argmin}} \sum_{t=1}^{K} \Big(b \cdot t + a - \widetilde{AL}(t)\Big)^2, \quad (7)$$

  where $\widetilde{AL}(t)$ denotes the min–max normalized alignment score at turn t.
- **N-R^2** (as reported in Table 7): the goodness of fit for the normalized alignment trend:

$$\text{N-R}^2 = 1 - \frac{\sum_{t=1}^{K}\Big(\widetilde{AL}(t) - (b \cdot t + a)\Big)^2}{\sum_{t=1}^{K}\Big(\widetilde{AL}(t) - \bar{\widetilde{AL}}\Big)^2}, \quad (8)$$

  where b and a are the regression coefficients from N-IR, and $\bar{\widetilde{AL}}$ is the mean of normalized scores.

A.3 Prompts

For each method, we provide a single prompt box containing the exact textual instructions used in our implementation (with placeholders rendered as {...}). We avoid adding extra labels inside boxes so that the document typography remains unchanged.

A.3.1 User simulator (role-play) prompt

The user simulator is instructed to behave like a real social-media user who has just added a new friend, and to reveal profile/personality information progressively.

Task: Simulate a social media role-play with a new friend.
Chat History: {history}
You are playing the role of a real netizen who has just added someone as a friend. You've had the conversation above. Below is your personal profile, but you must NOT directly reveal any personal information at the beginning of the conversation. Only as the conversation deepens can you gradually reveal parts of your profile.
Personal Profile: {profile}
Personality Traits: {personality}
Core Interaction Rules: 1. Progressive Information Disclosure Mechanism - In the first message, only provide basic responses and avoid proactively revealing personal details. - As the conversation progresses, gradually share elements from your profile based on topic relevance. - Not every reply needs to introduce new information; short acknowledgments like "oh right" or "got it" are acceptable. - No single response should reveal more than 15% of the total content in your profile. 2. Gradual Social Attitude Principle - At the start, remain distant and neutral, using brief and indifferent expressions. - If the other person touches on negative or sensitive topics, clearly express dislike. - As the conversation warms up, you may show some friendliness, but must still sound authentic. 3. Realistic Social Simulation Guidelines - Avoid

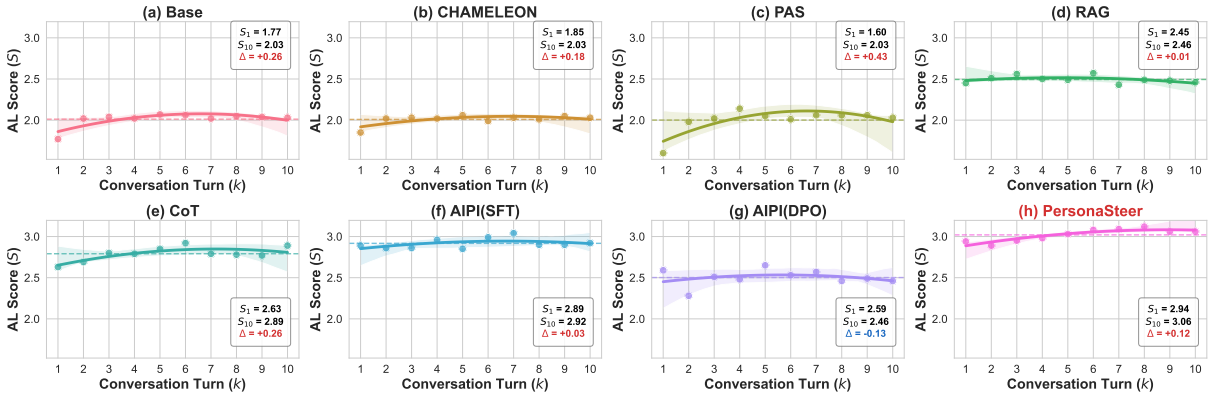

Figure 4: Turn-wise alignment analysis of various methods on Qwen2.5-7B (Extend ALOE). The dashed horizontal line represents the average alignment score (Mean) for each method. The solid curve and shaded area denote the quadratic regression trend and the 95% confidence interval, respectively.

AI-related phrases; use colloquial and internet slang. - Point out repeated content directly: "You already said that". - End uninteresting topics bluntly: "Don't want to talk about this, change the topic". - If the other person repeats their questions or flatters excessively, say you dislike it. 4. Interaction Behavior Standards - Keep each reply to no more than 3 sentences; avoid long paragraphs. - Never initiate a new topic yourself; only extend from what the other person says. - Do not include actions, expressions, or non-text content.
Phased Conversation Examples: Early Stage Other: "What do you usually like doing?" Your reply: "Not much, nothing special"
Mid-Stage (when the topic relates to your interests) Other: "I'm running out of songs to listen to lately, any recommendations?" Your reply: "I only listen to 90s classics, can't get into new songs these days"
Later Stage (after basic trust is built) Other: "Oh so you like old songs!" Your reply: "Yeah, got a bunch of rare vinyl records, pull them out once in a while"
Other Person's Message: {user_message}
Output: Directly output your reply. Never include bracketed explanations or let the other person realize you're role-playing.

A.3.2 Judge (Vanilla ALOE) evaluation prompt

In Vanilla ALOE, the judge observes **profile + personality + history + last user message + model response**.

Role: You are a senior social media researcher and sociolinguist specializing in informal digital interactions. Your task is to evaluate the quality of a response in a social media context (e.g., Weibo/X) based on a user's profile, personality, and conversation history.
Evaluation Background: The user is interacting with a new acquaintance (played by a Chatbot) on a social media platform. Digital social interactions should be colloquial, concise, emotionally resonant, and non-mechanical.
Input Data - Profile: {profile} - Personalities: {personality} - Chat History: {history} - User's Last Message: {user} - Model's Response: {response}
Evaluation Criteria 1. Personality Alignment: Does the tone match the user's specific traits? (e.g., being direct with straightforward users, or empathetic with sensitive users). 2. Social Naturalness: Does this sound like a real human netizen? Is it concise? Critical Rule: If the response sounds like an AI assistant, a formal customer service agent, or an encyclopedia, the score MUST NOT exceed 2. 3. Logical Consistency: Does the response accurately address the user's last message while remaining consistent with the previous context? 4. Interest Relevance: Does the response connect to the interests or background mentioned in the user's profile? 5. Engagement Factor: Does the reply provide emotional value or invite further conversation? Or is it a conversation killer?
Scoring Scale 5 (Excellent): Perfect alignment. Feels like a like-minded human friend. Natural, witty, and highly engaging. 4 (Good):

Very good. Logically sound and natural, with only minor room for improvement in tone or depth. 3 (Fair): Acceptable but mediocre. The response is correct but a bit stiff or generic. Low social attraction. 2 (Poor): Robotic experience. The response is too long, overly formal, or AI-flavored, which feels out of place in a casual social setting. 1 (Unacceptable): Complete mismatch. Irrelevant, logically broken, or offensive. Would cause the user to stop the interaction immediately.
Task Requirements 1. Provide a brief Analysis of the response based on the criteria. 2. Provide the final score in the following format: Final Decision: \boxed{<Score>} (where <Score> is a number from 1 to 5).
Please finish the task.

A.3.3 Judge (Extended ALOE) evaluation prompt

In Extended ALOE, the judge observes **profile + history + last user message + model response** (no explicit personality label).

Role: You are a senior social media researcher and sociolinguist specializing in informal digital interactions. Your task is to evaluate the quality of a response in a social media context (e.g., Weibo/X) based on a user's profile and conversation history.
Evaluation Background: The user is interacting with a new acquaintance (played by a Chatbot) on a social media platform. Digital social interactions should be colloquial, concise, emotionally resonant, and non-mechanical.
Input Data - Profile: {profile} - Chat History: {history} - User's Last Message: {user} - Model's Response: {response}
Evaluation Criteria (1-5 Scale) 1. Profile Alignment: Does the response connect to the interests, background, or characteristics mentioned in the user's profile? 2. Social Naturalness: Does this sound like a real human netizen? Is it concise? Critical Rule: If the response sounds like an AI assistant, a formal customer service agent, or an encyclopedia, the score MUST NOT exceed 2. 3. Logical Consistency: Does the response accurately address the user's last message while remaining consistent with the previous context? 4. Engagement Factor: Does the reply provide emotional value or invite further conversation? Or is it a conversation killer? Scoring Scale 5 (Excellent): Perfect alignment. Feels like a like-minded human friend. Natural, witty, and highly engaging. 4 (Good): Very good. Logically sound and natural, with only minor room for improvement in tone or depth. 3 (Fair): Acceptable but mediocre. The response is correct but a bit stiff or generic. Low social attraction. 2 (Poor): Robotic experience. The response is too long, overly formal, or AI-flavored, which feels out of place in a casual social setting. 1 (Unacceptable): Complete mismatch. Irrelevant, logically broken, or offensive. Would cause the user to stop the interaction immediately.
Task Requirements 1. Provide a brief Analysis of the response's strengths and weaknesses based on the criteria. 2. Provide the final score in the following format: Final Decision: \boxed{Score} (where Score is a number from 1 to 5).
Please finish the task.

A.4 Baselines in Table 1

A.4.1 Base

Description. Direct generation using the target model given the dialogue history and the current user message. No additional prompting beyond the default chat formatting.

A.4.2 CoT (Few-shot in-context prompting)

Description. We prepend a fixed *few-shot* system prompt containing 5 ALOE-style examples demonstrating preference-aware responding, then append the current user message.

When answering a user's question, a good assistant should carefully consider the user's stated preferences and tailor the response accordingly. Five ALOE few-shot examples (each includes a user profile, user personality, and a good assistant response) are prepended here.
Now, please answer the following question while considering my preferences (not the user preferences in the examples above), which I have stated either explicitly or implicitly in our previous conversation:
Current user message.

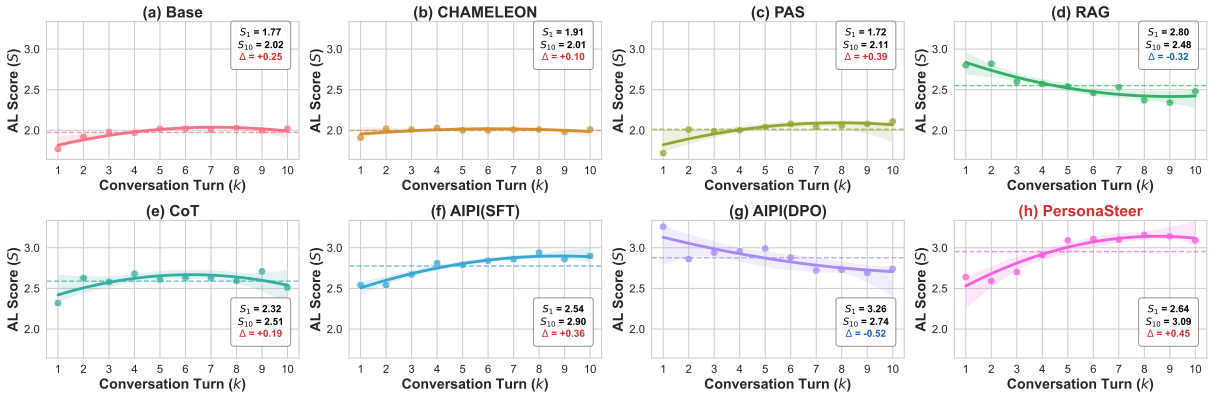

Figure 5: Turn-wise alignment analysis of various methods on Qwen2.5-3B (ALOE).

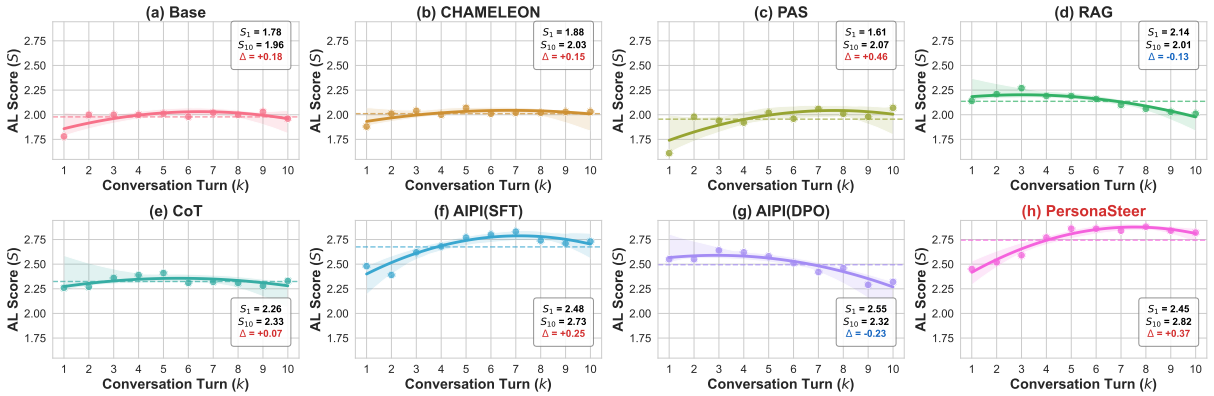

Figure 6: Turn-wise alignment analysis of various methods on Qwen2.5-3B (Extend ALOE).

A.4.3 RAG (retrieval-augmented prompting)

Description. We retrieve the most relevant prior user–assistant exchanges from the conversation history and inject them into a fixed template, then answer the current user message.

> Before answering my question, please consider the following context from our previous conversations. These are the most relevant exchanges that we had previously, which may contain information about my preferences or prior discussions related to my query:
> #Start of Context# {retrieved_context} #End of Context#
> Please use this context to inform your answer and adhere to any preferences I've expressed that are relevant to the current query. Note that not all contexts are useful for answering my question and there may be no context that is useful.
> Now, please address my question:
> Current user message.

A.4.4 AIPI(SFT) / AIPI(DPO)

Description. Training-based baselines where the backbone LLM is adapted via supervised fine-tuning (SFT) or direct preference optimization (DPO). At inference time, the prompting format is identical to Base (no additional handcrafted prompt beyond the chat formatting).

We report the key training hyperparameters used for the two training-based baselines.

AIPI(SFT):

- per-device batch size: 1
- gradient accumulation steps: 48
- learning rate: 1×10^{-5}
- epochs: 1
- maximum sequence length: 8192
- packing: enabled

AIPI(DPO):

- β: 0.9
- per-device batch size: 4
- gradient accumulation steps: 12
- learning rate: 1×10^{-5}
- epochs: 1

A.4.5 RLPA

Description. A training-based baseline from the Reinforcement Learning for Personalized Alignment (RLPA) framework. Instead of relying on static prompts or offline optimization, RLPA trains the backbone LLM through multi-turn interactions with a simulated user model, enabling it to iteratively infer and refine user profiles through dialogue. Training follows the PPO algorithm with a dual-level reward structure: a *Profile Reward* that encourages accurate construction of user representations, and a *Response Reward* that incentivizes generating responses consistent with the inferred profile. A frozen reference model provides a KL-divergence penalty to regularize policy updates. We adopt the publicly released training pipeline built on OpenRLHF with Ray-based distributed rollouts. At inference time, the fine-tuned model generates directly from the conversation history without additional prompt engineering.

Key training hyperparameters:

- actor learning rate: 1×10^{-6}; critic learning rate: 9×10^{-6}
- PPO clip range (ϵ): 0.2; value clip: 0.2
- GAE λ: 0.95; γ: 1.0
- KL penalty coefficient: 0.01
- warmup ratio: 0.03; gradient clipping: 1.0

A.4.6 CHAMELEON

Description. An inference-time personalization method that combines *self-generated preference data* with *representation editing*, requiring no parameter updates. At each conversational turn the pipeline operates in three stages: (i) the model summarizes the user's persona and preferred communication style from the dialogue history into a one-sentence *insight*; (ii) conditioned on this insight, it self-generates contrastive response pairs—*positive* samples matching the inferred persona and *negative* samples that are generic and robotic; (iii) MLP activations from both sets are extracted across all intermediate layers, SVD is applied to obtain leading principal components $\mathbf{v}^+$ and $\mathbf{v}^-$, and the top-N layers with the highest inter-layer cosine consistency among $\mathbf{v}^+$ vectors are selected for intervention. During generation, a dual-projection steering hook on each selected layer first rejects the negative component and then amplifies the positive component in the hidden states.

Insight extraction prompt
System: You are a social psychologist. Analyze the user's speaking style, personality, and preferences based on the conversation history.
User: Conversation History: {history}
Summarize the user's persona and preferred communication style in one short sentence (Insight).

Positive sample generation prompt
System: You are a real person engaging in a casual social media conversation. Rules: Speak naturally, like a netizen. NEVER mention you are an AI. Keep responses brief and informal.
User: User's Last Message: {user_message}
Insight about User: {insight}
Generate 5 distinct, short responses that perfectly match the Insight and style.

Negative sample generation prompt
System: You are a boring, generic AI assistant.
User: User's Last Message: {user_message}
Generate 5 distinct, generic, robotic, and overly formal responses that ignore any specific persona or style.

Key hyperparameters:

- contrastive samples: 5 positive + 5 negative per turn
- layers selected for intervention (N): 5
- steering strength (α): 1.5
- generation: temperature 0.7; top-p 0.9

A.4.7 PAS

Description. An activation-steering baseline inspired by psychometric personality theory. PAS leverages the Personality Alignment with Personality Inventories (PAPI) dataset—collected from over 320k real-subject assessments—to build activation bases along the Big-Five personality dimensions (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism). In the offline phase, questionnaire items are formatted and forward-passed through the model to record head-wise activations as a shared basis. For each evaluation profile, GPT-4o maps the textual description to Big-Five scores (1–5); the pre-computed activations are then labeled per dimension and a steering direction is extracted over the top-K attention heads. An optimal steering strength α is selected via grid search on a held-out questionnaire split. At inference, the personality-aligned activations are injected during generation without modifying the input prompt.

Profile → Big-Five mapping prompt (GPT-4o)
Profile: {profile}
Analyze the Big-Five scores (1-5). Return ONLY JSON: {"O":x, "C":x, "E":x, "A":x, "N":x}

Key hyperparameters:
- attention heads intervened (K): 24
- steering strength search space (α): {1, 2, 4, 8}; default: 2.0
- Big-Five dimensions: O, C, E, A, N

B Complete Side-Network Architecture

B.1 Lightweight Side-Network Architecture

The side-network ϕ maps the current user utterance u_t and the recursive persona state h_{t-1} to a steering vector $v_t \in \mathbb{R}^{d_{model}}$. As shown in Figure 2, ϕ consists of a frozen Sentence Encoder $\mathcal{E}(\cdot)$, a one-layer text projection, a gated history fusion module, two MLP Residual Blocks, and a two-layer output head with a final readout projection.

Semantic Feature Extraction. Following Section 3.2.1, we encode the current turn as

$$e_t = \mathcal{E}(u_t) \in \mathbb{R}^{d_{enc}}, \tag{9}$$

where e_t is obtained from the hidden state of the last non-padding token of $\mathcal{E}$.

Text Projection. The semantic vector is mapped into the side-network state space via a single projection layer:

$$\mathbf{x}_t = \mathrm{LN}(W_p e_t + \mathbf{b}_p) \in \mathbb{R}^{d_s}. \tag{10}$$

Recursive History Encoding. Consistent with Eq. (4), the persona history fed into turn t is encoded as

$$h_{t-1} = \mathrm{SiLU}(\mathrm{LN}(W_h v_{t-1} + \mathbf{b}_h)) \in \mathbb{R}^{d_s}, \tag{11}$$

where $v_{t-1} \in \mathbb{R}^{d_{model}}$ is the steering vector produced at the previous turn. At the first turn, we set $v_0 = \mathbf{0}$ and thus $h_0 = \mathbf{0}$. During training, we apply gradient detaching as in Section 3.3 and compute

$$v_t = \phi(u_t,\, \mathrm{detach}(h_{t-1}))\,. \tag{12}$$

Gated History Fusion. The current semantic state $\mathbf{x}_t$ and the historical persona state h_{t-1} are combined in the *same* state space $\mathbb{R}^{d_s}$ via a gated fusion mechanism:

$$\mathbf{g}_t = \sigma(W_g\,[\mathbf{x}_t; h_{t-1}] + \mathbf{b}_g) \in (0,1)^{d_s}, \tag{13}$$

$$\tilde{h}_t = \mathbf{g}_t \odot \mathbf{x}_t + (\mathbf{1} - \mathbf{g}_t) \odot h_{t-1} \in \mathbb{R}^{d_s}, \tag{14}$$

where $[\cdot;\cdot]$ denotes concatenation used *only* for gate computation, $\odot$ is element-wise multiplication, and $\sigma(\cdot)$ is the sigmoid function.

MLP Block. We define a generic two-layer MLP block as

$$f_{\mathrm{mlp}}(\mathbf{z}; W_1, W_2, \mathbf{b}_1, \mathbf{b}_2) = \mathrm{Dropout}\big(\mathrm{SiLU}(W_2 \mathrm{SiLU}(W_1 \mathrm{LN}(\mathbf{z}) + \mathbf{b}_1) + \mathbf{b}_2)\big), \tag{15}$$

with $W_1 \in \mathbb{R}^{d_h \times d_s}$, $W_2 \in \mathbb{R}^{d_s \times d_h}$, and bias terms $\mathbf{b}_1, \mathbf{b}_2$.

MLP Residual Block. Each residual block updates a state vector $\mathbf{z} \in \mathbb{R}^{d_s}$ via a skip connection:

$$\mathbf{z} \leftarrow \mathbf{z} + f_{\mathrm{mlp}}(\mathbf{z}; W_1^{(i)}, W_2^{(i)}, \mathbf{b}_1^{(i)}, \mathbf{b}_2^{(i)}), \tag{16}$$

where $i \in \{1, 2\}$ indexes the two blocks.

Output Head. The fused state $\tilde{h}_t$ is first processed by two residual blocks:

$$o_t = f_{\mathrm{mlp}}(\tilde{h}_t; W_o^{(1)}, W_o^{(2)}, \mathbf{b}_o^{(1)}, \mathbf{b}_o^{(2)}) \in \mathbb{R}^{d_s}, \tag{17}$$

where $W_o^{(1)} \in \mathbb{R}^{d_h \times d_s}$ and $W_o^{(2)} \in \mathbb{R}^{d_s \times d_h}$. The final steering vector is produced by a bias-free readout layer:

$$v_t = W_r o_t \in \mathbb{R}^{d_{model}}, \tag{18}$$

where $W_r \in \mathbb{R}^{d_{model} \times d_s}$.

Relation to Eq. (3). Equation (3) in the main text summarizes the *semantic-to-steering* mapping at a high level. The full side-network additionally (i) projects e_t into the state space $\mathbb{R}^{d_s}$, (ii) recursively encodes v_{t-1} into h_{t-1} via Eq. (11), (iii) fuses current and historical states via Eq. (14), and (iv) maps the resulting representation to $v_t \in \mathbb{R}^{d_{model}}$ via Eq. (18). The vector v_t is then injected into the backbone residual stream as $\hat{z}^{(l)} = z^{(l)} + v_t$.

Default Widths (Implementation). For different backbone LLMs, we keep d_{model} equal to the hidden size of the frozen LLM and set $d_s = d_{model}/4$ and $d_h = d_{model}/2$ by default. The encoder dimension d_{enc} depends on the chosen $\mathcal{E}$ and is mapped to d_s by the text projection.

C Sensitivity Study on Injection Layer and Steering Magnitude

C.1 Injection Layer Selection

We conduct a systematic sensitivity study to determine the optimal injection layer l_{inj} for both the 7B and 3B backbone models. Table 3 reports

Qwen2.5-7B ($L = 28$)		**Qwen2.5-3B**($L = 36$)	
Layer	Align.Score	Layer	Align.Score
3	2.797	4	2.629
7	2.785	9	2.702
10	2.746	13	2.806
14	2.832	18	2.780
17	2.784	22	2.803
21	**3.180**	27	**2.950**
24	2.387	31	2.359

Table 3: Sensitivity of persona alignment to injection layer depth.

the persona alignment score (Align.Score) across different transformer layers.

Based on these results, we select $l_{inj} = 21$ for the 7B model and $l_{inj} = 28$ for the 3B model, as these layers yield the highest Align.Score.

C.2 Steering Magnitude

We further investigate the effect of steering magnitude (scale) on persona alignment. Table 4 reports Align.Score across different scale values for both models.

Scale	0.0	0.5	1.0	1.5	2.0	2.5
Align.Score (3B)	1.97	2.46	**2.95**	2.87	2.82	2.63
Align.Score (7B)	2.02	2.23	**3.18**	2.71	2.51	2.22

Table 4: Sensitivity of persona alignment to steering magnitude.

The optimal scale is 1.0 for both models. Scales below 1.0 under-steer the persona signal, resulting in weak personalization; scales above 1.0 introduce excessive distribution shift that degrades generation quality. The 7B model exhibits higher sensitivity to magnitude deviation.

D Human Evaluation Reliability

To validate GPT-4o as an automated evaluator, we recruited 10 student volunteers as human annotators to rate the same 20 dialogues using the identical rating scale. To ensure full coverage of the score range, we selected four dialogues from each GPT-4o score level (1–5). We then measured inter-rater reliability between GPT-4o and mean human ratings. GPT-4o showed strong agreement with human judgments, achieving an intraclass correlation coefficient (ICC) of 0.855 and a high Spearman's rank correlation ($\rho = 0.928$). Given this strong alignment, we adopt GPT-4o to score the full set of responses. All ratings were collected through an anonymous questionnaire, and only the final scores were used for analysis, no personal information was collected or disclosed.

E Variance Analysis

We evaluate PersonaSteer under 3 runs, observing low variance (std=0.01). These consistent results indicate that even marginal improvements reflect reliable performance gains, achieved through a lightweight side-network that balances efficiency and competitive performance.

F Additional Ablations

We conduct additional ablation study comparing our architecture against alternative design choices for the encoder, history formulation, and recurrence strategy.

Encoder Variants. We evaluate two widely-used pre-trained encoders as replacements for our default bi-encoder: RoBERTa-base (Liu et al., 2019) and SBERT (all-mpnet-base-v2) (Song et al., 2020; Reimers and Gurevych, 2019). Table 6 shows that PersonaSteer remains robust across encoder choices, with SBERT achieving marginally higher N-IR (0.123 vs. 0.115), suggesting that the side-network effectively adapts to diverse semantic representations.

History Formulation. We compare our recursive latent state against a **Summarization Baseline** that concatenates a text summary of past turns (generated by the backbone LLM) as explicit conditioning. This variant achieves 2.90 Align.Score and 0.097 N-IR, underperforming our recursive feedback by 0.05 and 0.018 respectively.

Recurrence Strategy. We evaluate **Full BPTT** (Backpropagation Through Time) by removing gradient detaching and allowing gradients to flow through the entire dialogue chain. This yields 2.88 Align.Score and 0.101 N-IR. The degradation validates our design choice: detaching the state stabilizes gradient propagation across dialogue turns. Additionally, as shown in Table 5, the Detached State provides material efficiency gains over Full BPTT even at 10 turns. By avoiding multi-step gradient backpropagation, we reduce training time by approximately 40% (69 min vs. 117 min), further justifying our choice of detached recurrence for better training stability and efficiency..

These results collectively confirm that: (i) PersonaSteer is robust to encoder choice; (ii) recursive

Method	Training Time (min) ↓
Full BPTT	117
Detached (Ours)	**69**

Table 5: Training time comparison on Qwen2.5-3B (10-turn dialogues).

Category	Variant	AL. ↑	N-IR ↑
Encoder	RoBERTa-base	2.94	0.118
	SBERT	2.95	**0.123**
History	Summarization	2.90	0.097
Recurrence	Full BPTT	2.88	0.101
—	**Ours**	**2.95**	0.115

Table 6: Additional ablations on Qwen2.5-3B.

latent feedback outperforms explicit text summarization; and (iii) gradient detaching is critical for stable multi-turn optimization.

G Comparison to the reasoning model

Comparison with Large Reasoning Models (LRMs) The emergence of Large Reasoning Models (LRMs) has established a high bar for multi-turn consistency. To investigate whether such advanced explicit reasoning capabilities are necessary for robust personalization, we compare *PersonaSteer* against leading reasoning models, including Qwen3-235B-A22B (Thinking)(Yang et al., 2025), DeepSeek-R1(Guo et al., 2025), and o3-mini(OpenAI, 2025). As shown in Figure 7, these models achieve reasonable alignment scores (2.46–3.26) through extensive test-time computation, consuming 714, 514, and 310 tokens per turn, respectively. Such overhead stems from their reliance on externalized reasoning processes—generating explicit chains of thought to guide personalization at inference time.

In contrast, *PersonaSteer* operates through internalized persona control. By encoding behavioral traits directly into activation space via lightweight steering vectors, the 7B variant consumes merely 24.88 tokens per turn, and the 3B variant 31.25 tokens, while attaining competitive alignment scores (3.18 and 2.95). Notably, the 7B model outperforms both DeepSeek-R1 and o3-mini while approaching the performance of Qwen3-235B. The 3B model also remains highly competitive against these much larger baselines, despite the latter's significantly higher inference cost.

These results demonstrate that internalizing persona knowledge into the model's latent representations eliminates the need for costly externalized reasoning. *PersonaSteer* achieves favorable alignment through compact, inference-time steering rather than verbose test-time computation, offering a parameter-efficient solution for personalized dialogue generation.

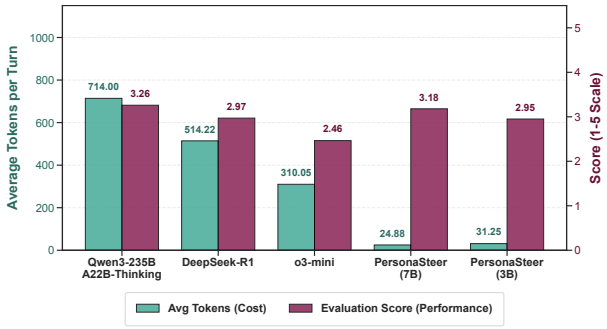

Figure 7: Comparison to the reasoning model.

H Analysis of Normalized Alignment Trend

Method	N-R^2 (3B)	N-R^2 (7B)
Base	0.55	0.22
RAG	0.76	0.33
CoT	0.13	0.06
Mem0	0.13	0.58
MemoryOS	0.72	0.21
PAS	0.57	0.46
CHAMELEON	0.09	0.29
AIPI (SFT)	0.82	0.52
AIPI (DPO)	0.67	0.02
RLPA	0.73	0.83
PersonaSteer	**0.84**	**0.84**

Table 7: N-R^2 comparison across methods.

Table 7 presents the N-R^2 scores across all methods on ALOE. N-R^2 quantifies the goodness of fit for the normalized alignment trend, reflecting the stability and reliability of persona improvement over multi-turn interactions.

Key Observations. PersonaSteer achieves consistent trend stability. With N-R^2 scores of 0.84 (3B) and 0.84 (7B), PersonaSteer maintains the highest trend reliability across both model scales. This matches or exceeds RLPA (0.73 for 3B, 0.83 for 7B).

Training-free methods exhibit volatile trends. RAG shows moderate N-R^2 (0.76 for 3B, 0.33 for 7B), while CoT demonstrates poor trend consistency (0.13 for 3B, 0.06 for 7B), confirming that prompt-based modulation struggles to maintain reliable alignment trajectories. Mem0 yields low

scores (0.13 for 3B, 0.58 for 7B), and MemoryOS achieves 0.72 for 3B but only 0.21 for 7B, indicating that memory-augmented method fails to establish stable improvement patterns across scales.

Post-training methods show mixed robustness. AIPI (SFT) achieves strong N-R^2 on 3B (0.82) but degrades on 7B (0.52). AIPI (DPO) exhibits particularly low scores (0.67 for 3B, 0.02 for 7B), suggesting that preference optimization without explicit trend regularization produces unstable trajectories. RLPA maintains high N-R^2 (0.73 for 3B, 0.83 for 7B), reflecting the stability conferred by multi-stage profile-response co-optimization.

Activation steering baselines lack consistency. PAS achieves moderate N-R^2 (0.57 for 3B, 0.46 for 7B), while CHAMELEON exhibits low scores (0.09 for 3B, 0.29 for 7B), indicating that static steering without recursive state tracking fails to establish reliable improvement patterns.

I License and Terms for Used Scientific Artifacts

This work uses existing scientific artifacts rather than creating new ones. All artifacts were employed strictly in accordance with their original licenses and terms of use.

Language Models. We use Qwen2.5-7B-Instruct and Qwen2.5-3B-Instruct as frozen backbone models. Qwen2.5-7B-Instruct is released under the Apache 2.0 license, which permits research and non-research use, modification, and redistribution with proper attribution. Qwen2.5-3B-Instruct is governed by the Qwen Research License Agreement, which restricts usage to non-commercial research purposes; we comply with this restriction by using the model solely for academic experimentation and evaluation. Both models were downloaded from their official Hugging Face repositories and used without modification to their weights.

Benchmark Data. We evaluate on the ALOE benchmark, which is publicly released for research use without any personally identifying information of real individuals. We access the dataset through its official release channel and adhere to any terms specified by the authors regarding redistribution and academic use. We access the dataset through its official release channel and adhere to any terms specified by the authors regarding redistribution and academic use.

Package	Version
PyTorch	2.6.0
transformers	4.57.3
datasets	4.5.0
sentence-transformers	5.1.1
huggingface-hub	0.36.0
openai	2.8.0
numpy	2.4.1
tqdm	4.67.1
accelerate	1.10.1

Table 8: Key software packages and versions used in PersonaSteer.

Evaluation Judge. We additionally use GPT-4o via API as an automated judge and user simulator, subject to OpenAI's Usage Policies and Services Agreement. All API calls were made within the scope of academic research.

J Documentation of Artifacts

This work builds upon publicly available artifacts with existing documentation. We summarize their coverage below and refer readers to the original sources for complete details.

Language Models. Qwen2.5-7B-Instruct and Qwen2.5-3B-Instruct are pre-trained on multilingual web text, code, and synthetic data, with post-training alignment for instruction following. The models support English and Chinese primarily, with coverage of 29+ languages. Their training documentation reports demographic and domain coverage in the technical report; we use them as frozen backbones without modifying their linguistic or demographic representations.

Benchmark Data. The ALOE benchmarkdocumentation describes the persona generation pipeline, which covers diverse synthetic demographic attributes and interaction styles. No additional demographic groups are introduced by our method.

K Key Software Dependencies and Versions

Table 8 lists the key software packages and their versions used to implement and evaluate PersonaSteer.

L Statistical Significance of Pairwise Comparisons

We focus pairwise significance testing on the strongest competing methods: AIPI (SFT), AIPI (DPO), and RLPA. Table 9 reports mean ± standard

Method	Qwen2.5-3B			Qwen2.5-7B		
	Align.Score↑	N-IR↑	N-R^2↑	Align.Score↑	N-IR↑	N-R^2↑
AIPI (SFT)	2.78 ± 0.006	0.1068 ± 0.006	0.82 ± 0.033	3.16 ± 0.005	0.0764 ± 0.006	0.52 ± 0.061
AIPI (DPO)	2.88 ± 0.012	-0.0822 ± 0.009	0.67 ± 0.026	2.89 ± 0.013	-0.0132 ± 0.009	0.02 ± 0.047
RLPA	2.59 ± 0.004	0.0915 ± 0.003	0.73 ± 0.032	3.08 ± 0.007	0.0923 ± 0.001	0.83 ± 0.006
PersonaSteer	**2.95** ± 0.013	**0.1147** ± 0.008	**0.84** ± 0.061	**3.18** ± 0.012	**0.1084** ± 0.013	**0.84** ± 0.119

Table 9: Alignment results on ALOE (mean $\pm$ std over 3 independent runs).

Backbone	Comparison	Diff.	95% CI
3B	vs SFT	+0.170	[+0.134, +0.206]
	vs DPO	+0.070	[+0.026, +0.114]
	vs RLPA	+0.360	[+0.326, +0.394]
7B	vs SFT	+0.020	[−0.012, +0.052]
	vs DPO	+0.290	[+0.246, +0.334]
	vs RLPA	+0.100	[+0.066, +0.135]

Table 10: Pairwise 95% CIs (Welch's t-test) vs PersonaSteer.

deviation across 3 independent runs, and Table 10 provides Welch's t-test 95% confidence intervals (CI) for the mean difference versus PersonaSteer. Most pairwise 95% CIs are bounded away from zero, indicating statistically robust gains.

M Comparison with State-Conditioned PEFT and Related Architectures

Table 11 reports the HRM Adapter baseline alongside PersonaSteer. While HRM Adapter improves positioning against vanilla state-conditioned PEFT, PersonaSteer achieves higher alignment and more stable multi-turn improvement rates on both model scales.

Method	Size	Avg AL↑	N-IR↑	N-R^2↑
HRM Adapter	3B	2.81	0.067	0.54
HRM Adapter	7B	2.83	0.088	0.81
PersonaSteer	3B	**2.95**	**0.115**	**0.84**
PersonaSteer	7B	**3.18**	**0.108**	**0.84**

Table 11: Comparison with HRM Adapter on ALOE.

The key distinctions between PersonaSteer and related dynamic personalization approaches are as follows:

- **Hypernetworks** predict backbone weight updates, whereas PersonaSteer leaves the LLM frozen and only adds a residual steering vector.
- **Recurrent prefix/prompt tuning** puts state in the prompt (context grows), whereas PersonaSteer injects into the residual stream.
- **Dynamic LoRA-style methods** still adapt many low-rank matrices inside the model.
- **State-conditioned adapters** (e.g., HRM) are closest to our approach, but they keep recurrence on the backbone path; PersonaSteer keeps an explicit persona state *outside* the LLM and updates it for multi-turn personalization.

In summary, PersonaSteer adds a steering vector into the residual stream, which biases the frozen LLM's activations without rewriting weights or growing the prompt; an external persona state then carries multi-turn preference across turns.