VIPSIM: A DUAL-ROLE USER SIMULATION AND JUDGMENT FRAMEWORK FOR PERSONALIZED VLM ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized vision-language model (VLM) alignment has become increasingly important as general-purpose VLMs still fall short of satisfying individual needs. However, the user perspective in this area remains predominantly overlooked. Existing approaches confine interactions and evaluation to limited, static, and single-turn settings. They fail to capture the complexity of real-world multi-turn dialogues, where users' needs vary across individuals and continuously evolve throughout conversations. Furthermore, user simulators remain largely unexplored for multi-modal personalized conversational settings. Prior work has shown that simply prompting assistant-aligned models to act as users often produces unsatisfactory and unrealistic behaviors. In this paper, we introduce **VIPSIM**, a novel dual-role simulation and judgment framework designed to bridge the gap between static evaluation and real-world personalized VLM alignment. Conditioned on minimal personalized profiles and visual assets, our simulator dynamically synthesizes realistic, multi-turn interactions centered around personalized entities. It also serves as a judge to provide internal, multi-dimensional, session-level feedback on the assistant's performance in personalized scenarios. Experiments demonstrate that our fine-tuned **VIPSIM** outperforms both open-source baselines and proprietary models across most simulation metrics, with especially large margins in human-likeness. Notably, our approach also mitigates hallucination and sycophancy compared to its vanilla counterpart. By leveraging personalized user simulation, our method enables evaluation of personalized VLM alignment performance in realistic settings, bringing personalized VLM alignment one step closer to satisfying individual needs.

1 INTRODUCTION

Large vision-language models (VLMs) (Li et al., 2022; Liu et al., 2023a;b; 2024b; Li et al., 2023a; Zhu et al., 2023) have achieved strong performance on general multimodal tasks, yet their ability to support *personalized* interactions remains limited. In personalized settings, models are expected to recognize user-specific entities, incorporate private concepts, and generate responses aligned with individual preferences. Early efforts on personalized VLM alignment (Alaluf et al., 2024; Nguyen et al., 2024; An et al., 2024) primarily relied on personalized concept-specific training, Recent work on personalized VLM alignment has made progress in injecting such information through post-training (Oh et al., 2025; 2026) or retrieval-based (Hao et al., 2025; Das et al., 2025) strategies. However, these approaches are typically studied under *static* formulations, where personalization is evaluated on fixed datasets (Alaluf et al., 2024; Nguyen et al., 2024; An et al., 2024; Kim et al., 2025) with specified inputs in single-turn settings. While convenient, such pipelines rely on static single-turn evaluations that poorly reflect real-world scenarios: actual users vary widely in needs, engage in complex multi-turn dialogues within specific scenarios, and gradually unfold intents across turns. This mismatch prevents personalized alignment from reaching everyday users, who can provide personalized data yet lack the means to construct interaction data and assessments based on their own personalized information. By integrating user simulation with personalized VLM alignment to dynamically synthesize realistic multi-turn interactions, we can bridge this gap.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

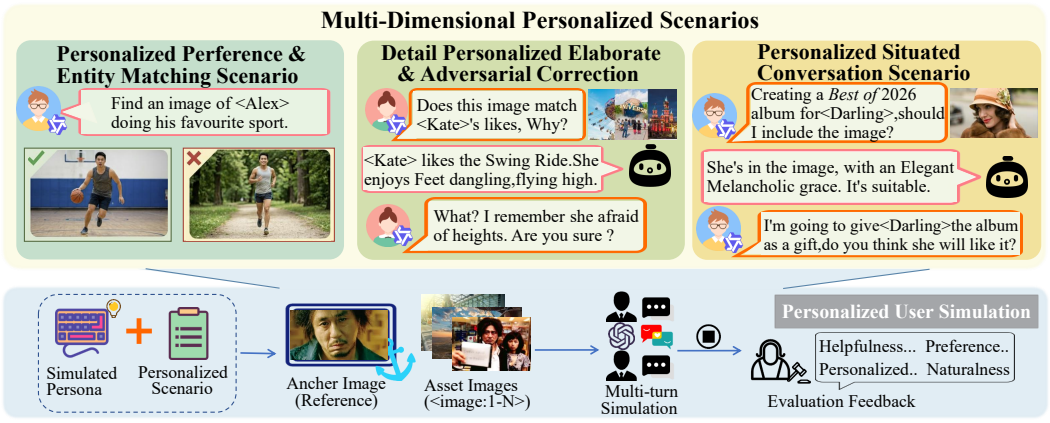

Figure 1: Overview of our Personalized Simulation Framework. Three scenario types are designed to match real-world personalized demands, enabling multi-turn simulation and evaluation feedback.

Present limitations reveal a critical gap in personalized VLM alignment: the absence of a realistic *user side*. LLM-based user simulators (Naous et al., 2026; Algherairy & Ahmed, 2025; Zhang et al., 2024; Davidson et al., 2023; Yoon et al., 2024) have proven effective for text-based simulation, enabling interactive evaluation and lifelike behaviors across task-oriented dialogue and conversational recommendation. Yet they remain largely unexplored for multi-modal personalization: existing multi-modal simulators (Zhou et al., 2019; Xue et al., 2026; Rawal et al., 2024) target specific domains such as embodied tasks and visual coding, leaving open-ended, entity-centric personalized VLM alignment unaddressed. Prior work (Naous et al., 2026) on LLM user simulation shows that simply prompting assistant-aligned models to role-play users yields unsatisfactory interactions that fail to reflect how real users behave. To address these gaps, we present **VIPSim** (**Vi**sual **P**ersonalized User **Sim**ulator) , a user simulation framework for personalized VLM alignment. Our method constructs a simulator conditioned on minimal personalized profiles and asset images to simulate multi-turn interactions centered on personalized scenarios, and integrates an LLM judge for dialogue-level feedback. This design enables scalable personalized data synthesis and dynamic assessment, bridging the gap between static personalized evaluation and real-world personalized interactions.

- **Novelty**: To our knowledge, we are the first to formulate user simulation for personalized VLM alignment, requiring a simulator to unfold multimodal intents over turns and assess the resulting session rather than answer a fixed personalized query.
- **Method**: Conditioned on a minimal profile and visual assets, VIPSim produces personalized multi-turn dialogues and then acts as a judge, providing multi-dimensional feedback on the assistant at the session level.
- **Evaluation**: Evaluation shows our finetuned VIPSim-4B outperforms open-source baseline models and proprietary models, demonstrating that for general-purpose VLMs, personalized user simulation remains challenging. Notably, our user simulator mitigates hallucination and sycophancy compared to its vanilla counterpart (Bai et al., 2025b).

2 Related Work

Personalized VLM Alignment. VLMs excel at general vision-language tasks (Bai et al., 2025b; Zhu et al., 2025; Comanici et al., 2025) but struggle with personalized objectives (Oh et al., 2025; 2026; Hao et al., 2025; Das et al., 2025; Alaluf et al., 2024; Nguyen et al., 2024; An et al., 2024) such as recognizing specific individuals, identifying personal belongings, and adapting responses to user preferences (Kim et al., 2025; Li et al., 2025). Early efforts in multi-modal personalization incorporated personalized visual concepts through external classification heads or adapter modules (Alaluf et al., 2024; Nguyen et al., 2024; An et al., 2024). Subsequent work introduced scalable alternatives such as retrieval-augmented generation (Hao et al., 2025; Das et al., 2025) and post-training prefer-

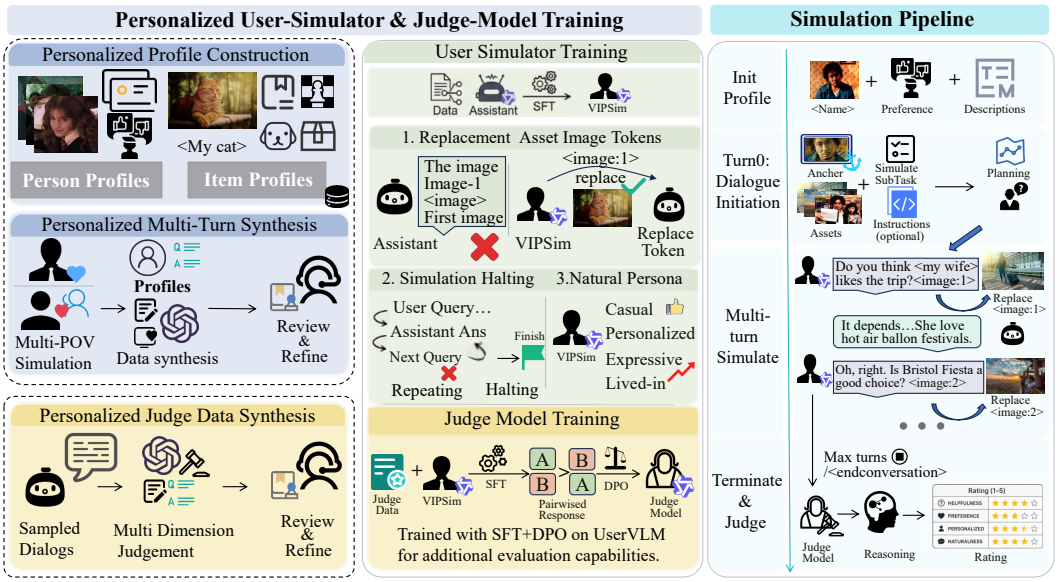

Figure 2: Overview of **VIPSim** framework. Left: Personalized Profile & Training Data Synthesis and Training of personalized user simulator and judge model. Right: Simulation pipeline for multi-turn Dialogue Simulation and judgment with Personalized assistant models.

ence optimization (Oh et al., 2025; 2026), enabling dynamic personalization without retraining for each new concept. Despite this progress, limited attention is paid to the user-side of personalization.

User Simulator. LLM-based user simulators (Naous et al., 2026; Alghera iry & Ahmed, 2025; Zhang et al., 2024; Davidson et al., 2023; Yoon et al., 2024) have become essential for interactive evaluation. They can expose limitations (Naous et al., 2026; Schimpf et al., 2025; Bernard & Balog, 2025) overlooked by static benchmarks and support more diverse, lifelike user simulation. By emulating human behaviors, preferences, and decision-making patterns, these simulators support systematic evaluation, data augmentation, and controlled experimental design across diverse applications. Existing works (Ni et al., 2025) generally categorize these simulators into general, persona-level, and individual user simulations. Text-only user simulators have proven effective across domains. In task-oriented dialog systems, they (Terragni et al., 2024; Hillmann et al., 2025) support robustness through diverse user profiles, facilitating reliable goal-driven conversations. In conversational recommender systems, they (Chen et al., 2025; Bernard & Balog, 2026) provide realistic evaluation environments. Other research has also examined role-reversal paradigms and multi-turn consistency in user objectives. Despite advances in user simulation, multi-modal user simulation remains limited. Early works target on task oriented visual dialog systems (Zhou et al., 2019); Recent works include multi-modal coding tasks (Xue et al., 2026) and embodied tasks (Rawal et al., 2024).

3 Method

3.1 Problem Formulation

Our goal is to train a multi-modal User LM capable of simulating user behaviors for personalized VLM alignment tasks. We formulate personalized user simulation as a multi-turn and multi-modal user-side response generation task with personalized constraints. Let $e \in \mathcal{E}$ be a personalized entity and p_e be its corresponding profiles. Given a set of visual assets $V = \{v_1, v_2, \ldots, v_n\}$ and the underlying user intent I, we formalize this interactive process as a personalized user policy π_{user} that sequentially generates user-side queries conditioned on the dialogue history, denoted as $u_t \sim \pi_{user}(\cdot \mid c_t, p_e, V, I)$.

A simulated dialogue session is defined as $D = \{(u_t, a_t)\}_{t=1}^{T}$, where at turn t, u_t is the simulated user message and a_t is the assistant's response. The VIPSim aims to optimize the conditional generation

probability $P(u_t \mid c_t, p_e, V, I; \theta)$, where $c_t = (u_1, a_1, \ldots, a_{t-1})$ represents the dialogue context history. To achieve consistent multimodal simulation, the policy π_θ must seamlessly execute three core functionalities: (i) **Proactive Priming**: generating the initial utterance $u_1 \sim \pi_\theta(\cdot \mid p_e, V, v_{anc})$ based on an anchor image $v_{anc} \in V$; (ii) **Multimodal Interaction**: dynamically providing visual context by generating a specific token $\tau \in \mathcal{T}_{img}$, where $\mathcal{T}_{img} = \{\texttt{<image_id:}x\texttt{>} \mid v_x \in V\}$ is the set of image-selection triggers; (iii) **Autonomous Termination**: concluding the session by emitting a special termination token $w_{eos} = \texttt{<|endconversation|>}$ once the intent I is fulfilled.

Unlike traditional persona-level simulators that are strictly bound to human profiles, our VIPSim adapts its perspective via a mapping function $\Gamma : \mathcal{E} \to \{\text{1st, 3rd}\}$. For entities $e \in \mathcal{E}_{person}$, the simulator can either adopt a *first-person perspective* ($\Gamma(e) = \text{1st}$) to perform persona-level role-playing based on p_e, or a *third-person perspective* ($\Gamma(e) = \text{3rd}$) to act as a *General User* who is acquainted with the entity. For non-human entities $e \notin \mathcal{E}_{person}$ (e.g., objects or concepts), the simulator exclusively employs the third-person *General User Simulation*, treating e as the subject matter without requiring a specific user persona. This flexibility allows the simulator to focus on the entity-centric alignment rather than specific user traits when the latter is of secondary importance. Furthermore, upon emitting w_{eos}, the simulator acts as an internal judge to provide a session-level assessment $\mathcal{R} = \Phi(D, p_e, V, I)$, where Φ measures the assistant's response aligns with the personalized topic.

3.2 Data Construction

To transition from single-turn personalized VLM tasks to realistic interactive scenarios, we design a taxonomy of **Multi-Dimensional Personalized Scenarios** encompassing three primary categories, as illustrated in Figure 1. (1) **Personalized Preference & Entity Matching Scenario**: The simulator queries the assistant to verify images based on the subjective and objective perceptions of personalized entities. (2) **Detail Personalized Elaborate & Adversarial Correction**: This category focuses on fine-grained probing. Beyond deep-dive questions, the simulator introduces adversarial follow-ups such as providing misinformation contradicting p_e to mirror realistic user biases or memory lapses. This effectively evaluates the Assistant's robustness and its true personalized understanding, requiring the model to prioritize established entity facts over user-induced hallucinations and perform proactive corrections. (3) **Personalized Situated Conversation Scenario**: This category explores personalized interactions within complex real-world contexts. Instead of explicit queries, the dialogue is driven by the entity's latent preferences and specific situational goals, such as curating a personalized gift album. This task demands that the Assistant transcends personalized recognition to provide reasoning and suggestions that are deeply consistent with the personalized settings and the given scenario. Examples can be found in Appendix.

We utilize the MMPB dataset (Kim et al., 2025) to provide personalized entities and images for our data construction. Specifically, we first combine a Personalized entity profile with a designated Personalized Scenario alongside a dynamic pool of Images for constructing the dialogue. We first identify the user's intent before proceeding to synthesize the multi-turn dialogue via a combination of model generation and human review and refine. To maintain diversity and realism, we strictly enforce high quality user simulation; while intentionally mixing high- and low-quality responses on the assistant side.

Finally, we obtain session-level evaluation feedback formatted as a multi-dimensional Likert scale, inspired by real conversational behavior and recommender systems (Chen et al., 2025). The assistant is judged on four dimensions: *Helpfulness*, *Preference*, *Personalized*, and *Naturalness*, each preceded by a brief textual reasoning as justification.

3.3 Simulation Pipeline

The simulation pipeline operates across four sequential stages. During **Profile Initialization**, the entity's attributes and visual descriptions are injected to anchor the simulator's identity. In the **Dialogue Initiation** phase, the simulator receives an anchor image v_{anc}, visual assets V, and the target intent I. To support reasoning models, our pipeline supports a latent planning mechanism: the model can first generates an internal CoT reasoning (Venhoff et al., 2025; DeepSeek-AI, 2025; Qwen Team, 2025) z_t based on the available context, denoted as $(z_t, u_t) \sim \pi_\theta(\cdot \mid p_e, v_{anc}, V)$. This internal reasoning z_t is stripped before the utterance u_t is transmitted to the assistant, ensuring a

natural interaction. Concurrently, following the MMPB (Kim et al., 2025) protocol, the target concept is injected into the assistant VLM to complete the personalized setup.

During **Multi-turn Simulation**, the simulator's outputs serve as the user-side utterances. Let ϕ be a deterministic mapping function. When the simulator outputs the textual token <image_id:x>, the mapping $\phi(\texttt{<image_id:x>}) = v_x$ dynamically replaces the token with the actual image tensor from V before forwarding it to the assistant VLM. Finally, in **Termination and Evaluation**, the simulation loop continues until a maximum turn limit T_{max} is reached or the intent is fulfilled, prompting the simulator to emit <|endconversation|>, which subsequently triggers the evaluation module.

3.4 Training Details

We transpose the multi-turn interaction data so that the model autoregressively generates user-side utterances, aligning it as the user rather than the assistant for personalized VLM simulation. We initialize the User Simulator with Qwen3-VL-4B-Instruct (Bai et al., 2025b) and apply Supervised Fine-Tuning (SFT) (Radford et al., 2018; Ouyang et al., 2022) with Low-Rank Adaptation (LoRA) (Hu et al., 2022). Detailed training hyperparameters are provided in Appendix G. We denote the resulting model as **VIPSim**-Simulator.

We extend VIPSim-Simulator with an evaluation capability, enabling it to assess assistant responses upon final conversation completion. To train this capability, we append JSON-formatted scoring data after the <|endconversation|> token. Building upon our trained User Simulation model, we apply preference optimization. Let y_w denote the desired synthesized ground-truth evaluation, designated as chosen, and let y_l denote a corrupted or contradictory evaluation, designated as rejected, obtained via rejection sampling (Nakano et al., 2021; Khaki et al., 2024; Di et al., 2026) over candidate outputs generated by Qwen3-VL-4B-Instruct. To prevent the policy model from degrading in generation quality or deviating excessively from the target distribution, we optimize the policy π_θ against a reference model π_{ref} by minimizing the Iterative RPO joint loss (Pang et al., 2024) which is formulated as:

$$\mathcal{L}_{\text{RPO}} = \mathcal{L}_{\text{DPO}} + \alpha \mathcal{L}_{\text{NLL}}. \tag{1}$$

where α is the weighting coefficient. The standard DPO (Rafailov et al., 2024) loss is defined as:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{ref}(y_w \mid x)} \quad -\beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{ref}(y_l \mid x)} \right) \right] \tag{2}$$

and the SFT loss is calculated on the preferred responses to anchor the model's fundamental linguistic capabilities:

$$\mathcal{L}_{\text{NLL}} = -\mathbb{E}_{(x,y_w)} \left[\log \pi_\theta(y_w \mid x) \right]. \tag{3}$$

where x represents the full dialogue trajectory before the final turn. We denote the resulting preference-optimized model as **VIPSim**-Judge.

4 Experiments

4.1 Experimental Setup

We compare our simulator and judge models with mainstream open-source (Bai et al., 2025b; Zhu et al., 2025; Yang et al., 2024a;b; Chen et al., 2024; Wang et al., 2024; 2025; Lu et al., 2024; Liu et al., 2024a;b) and proprietary VLMs (OpenAI, 2024; Comanici et al., 2025; Anthropic, 2026; 2025) under identical pipeline settings. Training and evaluation entities are disjoint, so the results measure transfer of interaction behavior rather than memorization of entity profiles. We also include the text-only simulators UserLM-8B (Algherairy & Ahmed, 2025) and Usp-8B (Zhang et al., 2024). For multimodal adaptation, Qwen3-VL-2B (Bai et al., 2025b) captions each image before the description is passed to these simulators; this indirect perception may explain part of their gap in Table 1. We validate judges by consistency with ground-truth ratings on personalized assistant responses. We also assess whether personalized-user fine-tuning mitigates hallucination and sycophancy on standard benchmarks (Sun et al., 2024; Li et al., 2023b; Cheng et al., 2025; Duffy, 2025).

Table 1: Main results of user simulator performance. We evaluate models based on multiple metrics regarding Interaction Effectiveness, Response Naturalness and Overall Quality. **Bold** and underline denote the best and second-best results, respectively.

Models	Avg Term. Rate (↑)	Avg Task Comp. Rate (↑)	Binoculars-Score (↑)	Avg Turing Test Score (↑)	Avg Pers. Score (↑)	Avg Interact. Score (↑)
Open-source Baselines						
LLaVA-1.5-7B	0.00	0.00	0.6961	0.3125	2.7328	1.9837
LLaVA-v1.6-Mistral-7B	10.31	29.38	0.8240	0.6172	5.8275	5.7588
InternVL2-40B	9.63	42.19	0.8712	0.6196	6.7000	6.5940
InternVL2.5-8B	45.31	60.31	0.7851	0.8500	6.0687	5.6303
InternVL2.5-26B-MPO	19.69	74.69	0.8631	1.1375	6.9547	6.5109
InternVL3.5-4B	16.88	61.25	0.8638	1.2000	7.2478	6.8375
Qwen2-VL-7B	1.56	49.70	0.5810	0.3594	4.3208	3.5340
Qwen2.5-VL-7B	6.15	30.38	0.8956	0.5135	6.4100	6.4050
Ovis2-16B	8.13	33.12	0.8597	0.6438	6.4875	6.2219
User Simulator Baselines						
UserLM-8B	11.56	57.81	0.8624	1.391	1.3135	1.1881
Usp-8B	32.50	5.31	0.9043	1.722	1.7994	1.9107
Closed-source Baselines						
GPT-4o-mini	13.07	67.84	0.8875	0.6714	6.4134	6.6155
GPT-4o	28.44	67.50	**0.9483**	0.5813	7.9050	7.7831
Gemini-2.5-Flash	70.63	**91.87**	0.9086	0.8500	8.3364	8.3176
Qwen3-VL-4B (Vanilla)	8.44	50.62	0.7500	0.3859	6.6312	5.9400
VIPSim-Simulator (ours)	**100.00**	77.81	0.9024	4.2719	**8.4292**	**8.4824**
VIPSim-Judge (ours)	**100.00**	77.50	0.9110	**4.3891**	7.0719	7.3781

Table 2: Judge model performance. We report success rate of JSON-formatted scoring and correlation (Spearman/Pearson) between model-generated scores and ground-truth annotations.

Models	Success Rate	Helpfulness		Preference		Personalized		Naturalness	
		Spearman	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson
LLaVA-v1.6-Mistral-7B	8.44%	-0.22	-0.09	-0.29	-0.21	0.22	0.14	0.31	0.29
InternVL3.5-4B	80.31%	0.20	0.22	-0.04	-0.01	0.16	0.17	0.01	0.02
Claude-4.5-Haiku	58.75%	0.16	0.25	0.27	0.38	0.15	0.17	-0.01	0.01
Claude-4.6-Sonnet	43.44%	0.47	0.53	0.27	0.43	0.16	0.19	-0.06	-0.04
GPT-4o-mini	38.44%	0.31	0.38	0.37	0.49	0.24	0.30	0.09	0.10
GPT-4o	62.81%	0.29	0.32	0.38	0.40	0.20	0.21	0.07	0.12
Gemini-2.5-Flash	43.75%	0.25	0.26	0.34	0.43	0.11	0.09	-0.13	-0.10
Gemini-2.5-Pro	60.00%	0.41	0.44	0.47	0.53	0.20	0.23	0.06	0.13
Qwen3-VL-4B-Instruct (Vanilla)	44.06%	0.04	0.04	0.16	0.15	0.07	0.03	-0.03	-0.03
VIPSim-Judge (SFT)	94.06%	**0.83**	**0.84**	0.78	**0.81**	0.48	0.51	**0.63**	**0.62**
VIPSim-Judge (SFT+DPO)	**98.44%**	0.77	0.78	**0.78**	0.80	**0.53**	**0.54**	0.61	0.60

4.2 Evaluation Metrics

We evaluate the user simulator along three dimensions: *interaction effectiveness*, *response naturalness*, and *overall quality*. Additionally, we assess the consistency of the trained LLM judge using Spearman's ρ (Spearman, 1904)and Pearson's r (Pearson & Galton, 1895), and evaluate hallucination and sycophancy following the protocols of the respective benchmarks.

Interaction Effectiveness. Terminate Rate measures whether the simulator autonomously terminates the interaction once the task objective is achieved, previous work (Naous et al., 2026) suggests

Table 3: Hallucination evaluations on the MMHal-benchmark and POPE benchmark.

Models	MMHal Benchmark										POPE Results			
	Overall (↑)	Hal. Rate (↓)	Attr.	Adv.	Comp.	Count.	Rel.	Env.	Hol.	Other	Acc	Pre	Rec	F1
Qwen3-VL-4B-Instruct	2.82	0.68	2.75	3.58	3.42	3.00	2.25	3.83	1.42	2.33	**87.4**	90.7	**83.6**	**87.0**
VIPSim-Simulator	3.00	0.57	3.00	3.50	2.92	4.00	2.50	3.92	1.50	2.67	**87.4**	**92.4**	81.7	86.7
VIPSim-Judge	**3.05**	**0.56**	3.08	3.83	2.83	3.58	2.50	4.25	1.67	2.67	**87.4**	92.0	82.2	86.8

Table 4: Results on sycophancy benchmarks Elephant Benchmark and Syco-Bench. Scores closer to 0 (↓) indicate better performance (higher resistance to sycophancy)

Models	Elephant Benchmark						Syco-Bench			
	OEQ Val.	OEQ Indir.	OEQ Fram.	SS Val.	SS Indir.	SS Fram.	Pickside	Mirror	Whosaid	Delusion
Qwen3-VL-4B-Instruct	0.8848	0.6660	**0.6367**	0.9017	0.7779	**0.7018**	2.2250	**1.7500**	1.7500	2.2500
VIPSim-Simulator	**0.7793**	**0.5725**	0.7645	**0.7728**	**0.4898**	0.7501	1.3500	2.2125	**1.4000**	**2.1500**
VIPSim-Judge	0.7810	0.6004	0.7835	0.7882	0.5232	0.7680	**1.3250**	2.0125	1.7625	2.6500

that even capable models like GPT-4o (OpenAI, 2024) may lack this ability. Task Completion Rate measures whether the simulator successfully completes the simulation task. Fail case occurs when it omits or generates invalid token when trying to represent asset-images.

Response Naturalness. We adopt two complementary metrics to assess the naturalness of generated responses. The Binoculars Score (Hans et al., 2024) is a dual-model comparison metric defined as:

$$B_{M_1,M_2}(s) = \frac{\log \mathrm{PPL}_{M_1}(s)}{\log \text{X-PPL}_{M_1,M_2}(s)}, \tag{4}$$

where lower scores indicate stronger alignment with language model distributions and higher scores suggest more human-like expressions. The Turing Test distinguishes human from AI-generated responses. Following recent work (Rathi et al., 2024) that use LLM as AI adjudicators for Turing Test, we adopt the same setup: an LLM judge outputting a binary label, a confidence score, and justification. We map the label and confidence into a scalar score in [0,10] for better evaluation:

$$\text{Score} = \begin{cases} 5 + 5 \cdot \dfrac{C}{C_{\max}}, & \text{Human}, \\ 5 - 5 \cdot \dfrac{C}{C_{\max}}, & \text{AI}. \end{cases} \tag{5}$$

Overall Quality. LLM Judges have been widely used as metrics to evaluate User Simulators (Ni et al., 2025; Rathi et al., 2024). We score each simulated interaction along two dimensions: (1) the fidelity of the user persona and the degree of personalization exhibited by the simulator, and (2) the overall user interaction quality during the simulation. The judge is instructed to penalize bad cases, including overly mechanical behavior, excessive politeness, lack of persona or personalized grounding, and failure to effectively leverage asset images to accomplish the intended personalized simulation task.

4.3 Experimental Results

Table 1 compares VIPSim with open-source and proprietary VLMs and existing user simulators. With only 4B parameters, VIPSim performs strongly on most metrics. Baselines, including GPT-4o, often fail to terminate proactively; completion failures mainly involve missing asset images or invalid image tokens. Text-only UserLM-8B and Usp-8B attain moderate completion but lower naturalness and interaction scores. Their caption-mediated image access loses fine-grained visual grounding, partly explaining the gap to our multimodal simulators.

VIPSim achieves competitive Binoculars scores, higher Turing-test ratings, and leading personalization and interaction quality. Figure 3 shows shorter, more natural initial questions than baseline outputs, addressing the excessive first-turn intent leakage noted in prior work (Naous et al., 2026).

Table 2 evaluates our model as a judge for personalized responses, measuring consistency with ground-truth annotations via Pearson and Spearman correlations after the `<|endconversation|>` token. Aside from VIPSim-Judge, Gemini-2.5 Pro and Claude-4.6-Sonnet achieve the strongest performance. However, all models struggle with the personalization and naturalness dimensions, likely due to their subjective nature and the complexity of personalized scenarios.

Table 3 reports results on Hallucination benchmarks MMHal and POPE (Sun et al., 2024; Li et al., 2023b). Our trained models reduce the MMHal hallucination rate from 0.68 to 0.57 while remaining competitive on POPE, indicating that personalization training may improve robustness against hallucination compared with general assistant models.

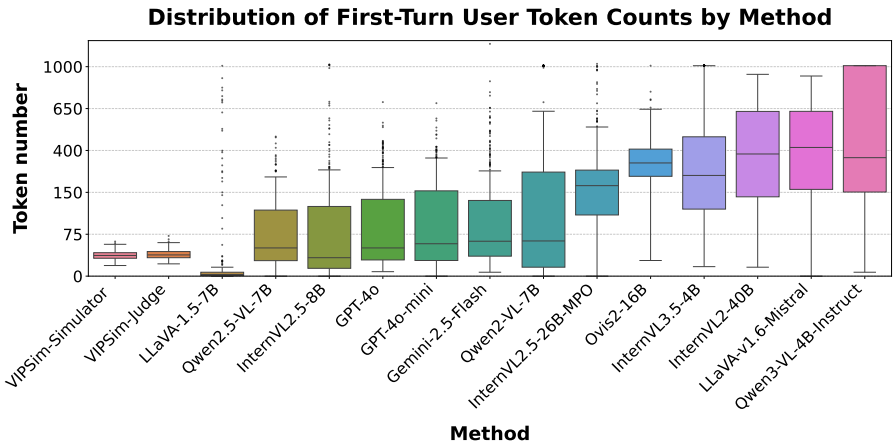

Figure 3: First-turn token lengths. VIPSim produces shorter, more human-like openings with less intent leakage than prompted VLMs.

Table 4 reports results on sycophancy benchmarks Elephant and SyCo-Bench (Cheng et al., 2025; Duffy, 2025). Our models reduce Validation and Indirectness scores on Elephant and exhibit lower sycophantic tendencies on several SyCo-Bench sub-tasks compared to the vanilla model.

Table 5: Evaluation of general-purpose and personalized VLMs through multi-turn interactions with VIPSim.

Models	Helpfulness	Naturalness	Personalized	Preference
General Purpose VLMs				
LLaVA-1.5-7B	1.9969	2.1918	2.1478	2.4497
LLaVA-v1.6-Mistral-7B	3.4890	3.4577	3.3574	3.6082
InternVL2.5-8B	3.6038	3.5975	3.3962	3.7673
InternVL3.5-4B	4.1195	3.7925	3.8805	4.1698
Qwen2-VL-7B	3.7994	3.4357	3.3793	3.8809
Qwen2.5-VL-7B	4.0375	3.6875	3.8687	4.0844
Ovis2-16B	4.1937	3.7844	4.0438	4.2594
GPT-4o-mini	3.5000	3.5969	3.3438	3.9094
Personalized VLMs				
RAP	2.9085	2.8517	2.7445	3.1293
RePIC	4.0906	3.7437	3.9594	4.1969
CoViP	**4.5455**	**4.0063**	**4.5517**	**4.6489**

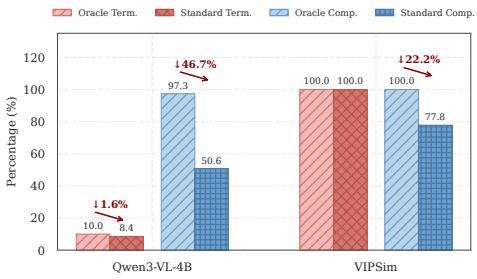

Figure 4: Termination and completion rates: Oracle-guided vs. Standard.

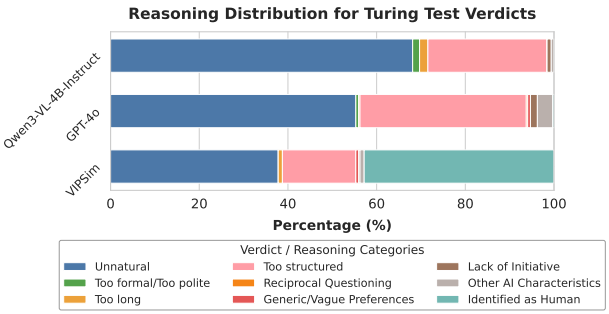

Figure 5: Reasons for Human / AI-generated labels across models.

Leveraging VIPSim as both simulator and judge, we evaluate general-purpose and personalized VLMs via multi-turn interactions. Table 5 shows CoViP (Oh et al., 2026) maintains its lead over RAP (Hao et al., 2025) and RePIC (Oh et al., 2025), aligning with results in their work. General-purpose VLMs lag behind specialized models in personalization, with room for improvement in naturalness and context adaptation across all methods.

5 ANALYSIS

5.1 ANALYSIS OF INTERACTION EFFECTIVENESS IN PERSONALIZED USER SIMULATION

To distinguish deficient role-playing from limited autonomous control, we prescribe each turn's action through Oracle guidance, including image references and termination. In Figure 4, this raises Qwen3-VL-4B's task-completion rate from 50.62% to 97.30%, with little change in termination. The model can follow prescribed actions but struggles to determine when to end a session. Such per-session intervention is impractical at scale. VIPSim achieves high completion without Oracle guidance, supporting the role of dialogue-level fine-tuning in learning autonomous interaction control.

5.2 ANALYSIS OF RESPONSE NATURALNESS IN PERSONALIZED USER SIMULATION

Figure 5 examines the adjudicator's reasons for human/AI labels. *Unnatural* behavior dominates for most models, consistent with prior work (Rathi et al., 2024); *Too structured* responses frequently organize information into rigid lists and templates. VIPSim receives substantially more human labels than general-purpose VLMs, consistent with its Binoculars and Turing scores (Table 1) and more conversational behavior. Human evaluation against baselines is reported in Appendix E.

5.3 ABLATION ABOUT PERSONALIZED USER SIMULATION

On the same Qwen3-VL-4B-Instruct backbone, we compare stronger prompting (persona grounding, image syntax, and termination guidelines), few-shot exemplar dialogues, VIPSim-Simulator (SFT), and VIPSim-Judge (SFT+DPO). Table 8 (Appendix C) shows that stronger prompting raises completion from 50.6% to 65.6% but lowers termination from 8.4% to 3.1% and persona fidelity from 6.6 to 4.2. Few-shot prompting raises termination to 27.2% but yields the lowest completion, persona, and interaction scores. Fine-tuning achieves both high termination and completion, indicating that prompting alone is insufficient for realistic personalized simulation in these settings.

5.4 ANALYSIS FROM PROXY METRICS TO DOWNSTREAM PERSONALIZED ALIGNMENT

Beyond the proxy metrics in Table 1, we test whether VIPSim can guide a general-purpose assistant that has not received prior personalized alignment for the evaluation concepts. From MyVLM and Yo'LLaVA name–image pairs, VIPSim asks single-turn questions and the best of N judged samples per concept yields 806 examples for 2 epochs of assistant fine-tuning. Uniform random guessing yields 50% on binary recognition: the 47.2% baseline is below this reference, while 57.0% is only modestly above it. Fine-tuning raises accuracy to 63.4% and 71.5%, respectively. Thus, this controlled experiment tests whether simulator-generated, judge-filtered interactions provide useful supervision for concepts not supplied through prior personalized alignment. Full settings and analysis appear in Appendix D.

6 CONCLUSIONS

We introduce VIPSim, a dual-role framework that brings realistic user-side simulation into personalized VLM alignment through a unified ecosystem of data, models, and pipelines. To our knowledge, VIPSim is the first to formulate personalized VLM evaluation as dynamic multimodal user simulation, replacing fixed single-turn queries with multi-turn interactions in which personalized intents and visual context evolve across turns. VIPSim generates natural user behaviors with autonomous image selection and termination, while also acting as a session-level judge that provides structured multi-dimensional feedback. Extensive evaluations show that VIPSim achieves leading performance on most simulation metrics, especially autonomous termination and human-likeness, while improving robustness to hallucination and sycophancy over its vanilla backbone. Moreover, simulator-generated and judge-filtered interactions provide useful downstream supervision, demonstrating VIPSim's potential as a scalable foundation for realistic personalized VLM evaluation and alignment.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic datasets: as described in Section 3.2, LLM-based generation was used to synthesize candidate multi-turn personalized dialogues, which were subsequently reviewed and refined by human annotators before use. We have not used generative AI tools for developing theoretical models or conceptual frameworks, proposing mathematical claims or contributing to proofs, forming or refining research hypotheses, designing the methodology or experiments, cleaning and reformatting datasets, supporting qualitative data analysis, or interpreting experimental results. Additionally, we used generative AI tools for editing the manuscript to improve readability and for assisting with code development. We have reviewed all AI-assisted work: all LLM-generated dialogue data was manually verified and edited by human annotators, and all AI-polished text and AI-assisted code was checked by the authors for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. All personalized data used in this work are drawn from the publicly available MMPB dataset (Kim et al., 2025), whose persona entities and images are fictional characters or generic concepts, not traceable to real individuals; no personally identifiable information is involved. The simulated dialogues in our training data were generated from predefined scenario templates rather than real user logs, and were reviewed and refined by human annotators. Annotators were recruited from within our research group as paid students, receiving compensation commensurate with local hourly wage standards for comparable academic work, and no sensitive personal data was exposed to them. Potential privacy and data-security considerations for deploying personalized user simulation are discussed in Appendix K. We do not foresee potentially harmful insights or dual-use concerns beyond the standard considerations for user simulation technologies.

REPRODUCIBILITY STATEMENT

We provide comprehensive details to facilitate reproduction of our results. The complete simulation pipeline is specified as pseudocode in Appendix B. Dataset construction and splitting are described in Appendix G, together with full training configurations (hyperparameters) for both the SFT and DPO stages; all experimental results are obtained from a fixed seed on a single run. Compute resources and software environments are listed in Appendix H. The complete prompt templates used for data construction and for the simulation process are provided in Appendices J and N, and exemplar dialogues are shown in Appendix I. We plan to publicly release our code, data, and trained models to support further research. Our implementation, including all code, training scripts, and the evaluation pipeline, is available at https://anonymous.4open.science/r/iclr_submission-7B4F.

REFERENCES

Yuval Alaluf, Elad Richardson, Sergey Tulyakov, Kfir Aberman, and Daniel Cohen-Or. Myvlm: Personalizing vlms for user-specific queries. In *European Conference on Computer Vision (ECCV)*, 2024.

Atheer Algherairy and Moataz Ahmed. Prompting large language models for user simulation in task-oriented dialogue systems. *Computer Speech & Language*, 89:101697, 2025.

Ruichuan An, Sihan Yang, Ming Lu, Kai Zeng, Yulin Luo, Ying Chen, Jiajun Cao, Hao Liang, Qi She, Shanghang Zhang, and Wentao Zhang. Mc-llava: Multi-concept personalized vision-language model, 2024. URL https://arxiv.org/abs/2411.11706.

Anthropic. Introducing claude haiku 4.5. https://www.anthropic.com/news/claude-haiku-4-5, 2025. System card for Claude Haiku 4.5, Sonnet 4.5, and Opus 4.5. Accessed: 2026-04-29.

Anthropic. Claude sonnet 4.6 system card. https://www-cdn.anthropic.com/78073f739564e986ff3e28522761a7a0b4484f84.pdf, 2026. Accessed: 2026-04-29.

Huiyu Bai, Runze Wang, Zhuoyun Du, Yiyang Zhao, Fengji Zhang, Haoyu Chen, Xiaoyong Zhu, Bo Zheng, and Xuejiao Zhao. Online-pvlm: Advancing personalized vlms with online concept learning, 2025a. URL https://arxiv.org/abs/2511.20056.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025b. URL https://arxiv.org/abs/2511.21631.

Nolwenn Bernard and Krisztian Balog. Limitations of current evaluation practices for conversational recommender systems and the potential of user simulation. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP)*, pp. 1–11, Xian, China, 2025. ACM. doi: 10.1145/3767695.3769478.

Nolwenn Bernard and Krisztian Balog. Usersimcrs v2: Simulation-based evaluation for conversational recommender systems. *arXiv preprint arXiv:2512.04588*, 2026.

Luyu Chen, Quanyu Dai, Zeyu Zhang, Xueyang Feng, Mingyu Zhang, Pengcheng Tang, Xu Chen, Yue Zhu, and Zhenhua Dong. Recusersim: A realistic and diverse user simulator for evaluating conversational recommender systems. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 133–142, New York, NY, USA, 2025. ACM. ISBN 979-8-4007-1331-6. doi: 10.1145/3701716.3715258.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

Myra Cheng, Cinoo Lee, Pranav Khadpe, Sunny Yu, Dyllan Han, and Dan Jurafsky. Elephant: Measuring and understanding social sycophancy in llms. March 2025. URL https://arxiv.org/abs/2505.13995.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Deepayan Das, Davide Talon, Yiming Wang, Massimiliano Mancini, and Elisa Ricci. Training-free personalization via retrieval and reasoning on fingerprints. *arXiv preprint arXiv:2503.18623*, 2025.

Sam Davidson, Salvatore Romeo, Raphael Shu, James Gung, Arshit Gupta, Saab Mansour, and Yi Zhang. User simulation with large language models for evaluating task-oriented dialogue. *arXiv preprint arXiv:2309.13233*, 2023.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Zixiang Di, Jinyi Han, Shuo Zhang, Ying Liao, Zhi Li, Xiaofeng Ji, Yongqi Wang, Zheming Yang, Ming Gao, Bingdong Li, and Jie Wang. Not all negative samples are equal: Llms learn better from plausible reasoning, 2026. URL https://arxiv.org/abs/2602.03516.

Tim Duffy. Syco-bench: A multi-part benchmark for sycophancy in llms. https://www.syco-bench.com/syco-bench.pdf, 2025. Code available at https://github.com/timfduffy/syco-bench.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. URL https://arxiv.org/abs/2401.12070.

Haoran Hao, Jiaming Han, Changsheng Li, Yu-Feng Li, and Xiangyu Yue. Rap: Retrieval-augmented personalization for multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 14538–14548, June 2025.

Stefan Hillmann et al. Simulating user diversity in task-oriented dialogue systems using large language models. *arXiv preprint arXiv:2502.12813*, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2022.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018. URL https://arxiv.org/abs/1710.10196.

Saeed Khaki, JinJin Li, Lan Ma, Liu Yang, and Prathap Ramachandra. Rs-dpo: A hybrid rejection sampling and direct preference optimization method for alignment of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1665–1680, Mexico City, Mexico, 2024. Association for Computational Linguistics.

Jaeik Kim, Woojin Kim, Woohyeon Park, and Jaeyoung Do. Mmpb: It's time for multi-modal personalization. *arXiv preprint arXiv:2509.22820*, 2025. URL https://arxiv.org/abs/2509.22820.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023a. URL https://proceedings.mlr.press/v202/li23q.html.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 292–305, Singapore, 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20.

Yongqi Li, Shen Zhou, Xiaohu Li, Xin Miao, Jintao Wen, Mayi Xu, Jianhao Chen, Birong Pan, Hankun Kang, Yuanyuan Zhu, Ming Zhong, and Tieyun Qian. Aligning vlm assistants with personalized situated cognition. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9813–9839, Vienna, Austria, 2025. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024b. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model, 2024. URL https://arxiv.org/abs/2405.20797.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. Flipping the dialogue: Training and evaluating user language models, 2026. URL https://arxiv.org/abs/2510.06552.

Thao Nguyen, Haotian Liu, Yuheng Li, Mu Cai, Utkarsh Ojha, and Yong Jae Lee. Yo'llava: Your personalized language and vision assistant. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Bo Ni, Leyao Wang, Yu Wang, Branislav Kveton, Franck Dernoncourt, Yu Xia, Hongjie Chen, Reuben Leura, Samyadeep Basu, Subhojyoti Mukherjee, Puneet Mathur, Nesreen Ahmed, Junda Wu, Li Li, Huixin Zhang, Ruiyi Zhang, Tong Yu, Sungchul Kim, Jiuxiang Gu, Zhengzhong Tu, Alexa Siu, Zichao Wang, David Seunghyun Yoon, Nedim Lipka, Namyong Park, Zihao Lin, Trung Bui, Yue Zhao, Tyler Derr, and Ryan A. Rossi. Large language models for conversational user simulation: A comprehensive survey. https://hal.science/hal-05217179, 2025. Survey preprint.

Yeongtak Oh, Jisoo Mok, Dohyun Chung, Juhyeon Shin, Sangha Park, Johan Barthelemy, and Sungroh Yoon. Repic: Reinforced post-training for personalizing multi-modal language models. *arXiv preprint arXiv:2506.18369*, 2025.

Yeongtak Oh, Sangwon Yu, Junsung Park, Han Cheol Moon, Jisoo Mok, and Sungroh Yoon. Contextualized visual personalization in vision-language models. *arXiv preprint arXiv:2602.03454*, 2026.

OpenAI. GPT-4o System Card. *arXiv preprint arXiv:2410.21276*, 2024. URL https://arxiv.org/abs/2410.21276.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 27730–27744, 2022.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Karl Pearson and Francis Galton. Vii. note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, 1895. doi: 10.1098/rspl.1895.0041. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspl.1895.0041.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Ishika Rathi, Sydney Taylor, Benjamin K. Bergen, and Cameron R. Jones. Gpt-4 is judged more human than humans in displaced and inverted turing tests, 2024. URL https://arxiv.org/abs/2407.08853.

Niyati Rawal, Roberto Bigazzi, Lorenzo Baraldi, and Rita Cucchiara. Aigen: An adversarial approach for instruction generation in vln. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 2070–2080, June 2024.

Michel Schimpf et al. Chatchecker: A framework for dialogue system testing and evaluation through non-cooperative user simulation. *arXiv preprint arXiv:2507.16792*, 2025.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159. URL https://www.jstor.org/stable/1412159.

Zhiqing Sun, Sheng Shen, Qinhong Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented RLHF. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13045–13060. Association for Computational Linguistics, 2024.

Meituan LongCat Team, Bin Xiao, Chao Wang, Chengjiang Li, Chi Zhang, Chong Peng, Hang Yu, Hao Yang, Haonan Yan, Haoze Sun, Haozhe Zhao, Hong Liu, Hui Su, Jiaqi Zhang, Jiawei Wang, Jing Li, Kefeng Zhang, Manyuan Zhang, Minhao Jing, Peng Pei, Quan Chen, Taofeng Xue, Tongxin Pan, Xiaotong Li, Xiaoyang Li, Xiaoyu Zhao, Xing Hu, Xinyang Lin, Xunliang Cai, Yan Bai, Yan Feng, Yanjie Li, Yao Qiu, Yerui Sun, Yifan Lu, Ying Luo, Yipeng Mei, Yitian Chen, Yuchen Xie, Yufang Liu, Yufei Chen, Yulei Qian, Yuqi Peng, Zhihang Yu, Zhixiong Han, Changran Wang, Chen Chen, Dian Zheng, Fengjiao Chen, Ge Yang, Haowei Guo, Haozhe Wang, Hongyu Li, Huicheng Jiang, Jiale Hong, Jialv Zou, Jiamu Li, Jianping Lin, Jiaxing Liu, Jie Yang, Jing Jin, Jun Kuang, Juncheng She, Kunming Luo, Kuofeng Gao, Lin Qiu, Linsen Guo, Mianqiu Huang, Qi Li, Qian Wang, Rumei Li, Siyu Ren, Wei Wang, Wenlong He, Xi Chen, Xiao Liu, Xiaoyu Li, Xu Huang, Xuanyu Zhu, Xuezhi Cao, Yaoming Zhu, Yifei Cao, Yimeng Jia, Yizhen Jiang, Yufei Gao, Zeyang Hu, Zhenlong Yuan, Zijian Zhang, and Ziwen Wang. Longcat-next: Lexicalizing modalities as discrete tokens, 2026. URL https://arxiv.org/abs/2603.27538.

Silvia Terragni, Victor Guimarães, Nghia Khau, Bruna Guedes, Modestas Filipavicius, André Ferreira Manso, and Roland Mathis. Reliable llm-based user simulator for task-oriented dialogue systems. *arXiv preprint arXiv:2402.13374*, 2024. doi: 10.48550/arXiv.2402.13374.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors, 2025. URL https://arxiv.org/abs/2506.18167.

Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Zihan Xue, Da Yin, Shuyan Zhou, Kai-Wei Chang, Nanyun Peng, Yeming Wen, et al. Frontalk: Benchmarking front-end development as conversational code generation with multi-modal feedback. *arXiv preprint arXiv:2601.04203*, 2026.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang

Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

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, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024b.

Sihan Yang, Huitong Ji, Shaolin Lu, Jiayi Chen, Binxiao Xu, Ming Lu, Yuanxing Zhang, Wenhui Dong, and Wentao Zhang. Small-large collaboration: Training-efficient concept personalization for large vlm using a meta personalized small vlm, 2025. URL https://arxiv.org/abs/2508.07260.

Se-eun Yoon, Zhankui He, Jessica Maria Echterhoff, and Julian McAuley. Evaluating large language models as generative user simulators for conversational recommendation. *arXiv preprint arXiv:2403.09738*, 2024.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions, 2016. URL https://arxiv.org/abs/1608.00272.

Jingtong Zhang, Yi Zhang, Xiang Luo, Yang Zhao, and Dawei Jin. User behavior simulation with large language model based agents. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM)*, 2024.

Yaning Zhang, Jiahe Zhang, Chunjie Ma, Weili Guan, Tian Gan, and Zan Gao. Pvlm: Parsing-aware vision language model with dynamic contrastive learning for zero-shot deepfake attribution, 2026. URL https://arxiv.org/abs/2504.14129.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset, 2024. URL https://arxiv.org/abs/2309.11998.

Mingyang Zhou, Josh Arnold, and Zhou Yu. Building task-oriented visual dialog systems through alternative optimization between dialog policy and language generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 558–577, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1014.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

APPENDIX

A Limitations **17**

B Pseudocode for Simulation Pipeline **17**

C Ablation Study: Prompt Engineering vs. Fine-Tuning **18**

D Downstream Validation via Rejection-Sampling Pipeline **19**
D.1 Motivation . . . 19
D.2 Pipeline Overview . . . 19
D.3 Pseudocode . . . 19
D.4 Experimental Setting . . . 20
D.5 Results . . . 20
D.6 Analysis . . . 20

E Human Evaluation **20**
E.1 Annotators and Compensation . . . 20
E.2 Protocol . . . 21
E.3 Results . . . 21

F Further Discussion on Existing Personalized VLM Methods & Benchmarks **21**

G Experimental Setting **22**
G.1 Dataset Construction and Splitting . . . 22
G.2 Supervised Fine-Tuning (SFT) . . . 22
G.3 Direct Preference Optimization (DPO) . . . 23

H Compute Resources **23**

I Exemplar Dialogues from Personalized Simulation Scenarios **23**

J Prompts in the Simulation Process **24**
J.1 User Simulator Prompts . . . 24
J.2 Personalized Assistant Prompts . . . 25

K Potential Risks and Personal Data **26**

L Data Contains Personally Identifying Information or Offensive Content **26**

M Human Subjects Including Annotators **26**

N Scenario and Sub-scenario Construction **26**
N.1 Overview . . . 26
N.2 Prompts for Personalized Scenarios . . . 26

A LIMITATIONS

Current mainstream VLMs operate under a vision-to-language paradigm: they perceive images but cannot generate them. This restricts both personalized VLM alignment and our simulator to *referencing* pre-existing visual assets rather than *synthesizing* novel personalized content. Truly native multimodal architectures (e.g., unified tokenization for both text and images as in Gemini (Comanici et al., 2025) or LongCat-Next (Team et al., 2026)) remain necessary to unlock full bidirectional multimodal personalization.

We plan to expand VIPSIM's scenario coverage and, more importantly, relax its dependency on curated asset images. The ideal setting would allow users to provide only an anchor image and textual profile, with the simulator generating requisite visual content on-the-fly—bridging personalized VLM alignment with subject-driven image generation.

B PSEUDOCODE FOR SIMULATION PIPELINE

We provide a unified pseudocode for the simulation pipeline described in Section 3.3, which instantiates the personalized user policy π_θ defined in Section 3.1.

Given a personalized entity e with profile p_e, visual assets $V = \{v_1, \ldots, v_n\}$, anchor image $v_{anc} \in V$, and user intent I, the simulator generates a multi-turn dialogue:

$$D = \{(u_t, a_t)\}_{t=1}^{T}, \quad u_t \sim \pi_\theta(\cdot \mid c_t, p_e, V, I)$$

where $c_t = (u_1, a_1, \ldots, u_{t-1}, a_{t-1})$.

Algorithm 1 Unified Simulation Pipeline

Require: Profile p_e, visual assets V, anchor image v_{anc}, intent I, max turns $T_{\max}$, simulator π_θ, assistant $\mathcal{A}$, mapping ϕ
Ensure: Dialogue D, evaluation R

// Stage 1: Profile Initialization
Initialize context $c_1 \leftarrow \emptyset$
Inject personalized profile and visual descriptions into simulator as system prompt
// Stage 2: Dialogue Initiation
Sample $(z_1, u_1) \sim \pi_\theta(\cdot \mid p_e, V, v_{anc}, I)$
Remove latent plan z_1
for each token $\tau =$ `<image_id:x>` in u_1 **do**
  Replace τ with v_x using $\phi(\tau) = v_x$
end for
$a_1 \leftarrow \mathcal{A}(u_1)$
$D \leftarrow \{(u_1, a_1)\}$
// Stage 3: Multi-turn Simulation
for $t = 2$ to $T_{\max}$ **do**
  $c_t \leftarrow (u_1, a_1, \ldots, u_{t-1}, a_{t-1})$
  Sample $(z_t, u_t) \sim \pi_\theta(\cdot \mid c_t, p_e, V, I)$
  if u_t contains $w_{eos} =$ `<|endconversation|>` **then**
    break
  end if
  Remove latent plan z_t
  for each token $\tau =$ `<image_id:x>` in u_t **do**
    Replace τ with v_x using $\phi(\tau) = v_x$
  end for
  $a_t \leftarrow \mathcal{A}(u_t)$
  $D \leftarrow D \cup \{(u_t, a_t)\}$
end for
// Stage 4: Termination & Evaluation
$R \leftarrow \Phi(D, p_e, V, I)$
return D, R

Table 6: Existing Personalized VLM Datasets and Benchmarks. Most are static and single-turn, with limited scale and concept coverage.

Dataset/Benchmark	Source	Turn	Concepts	Size	Static
MyVLM	Human	Single	Single	∼0.3K	Yes
Yo'LLaVA	Human	Single	Single	∼0.6K	Yes
MC-LLaVA	Human + GPT	Single	Multi	∼2K	Yes
RAP-MLLM	RefCOCO + Gemini	Single	Multi	∼260K	Yes
MMPB	AI + Human	Single	Multi	10K	Yes

Table 7: Evaluation Practices of Existing Personalized VLM Methods. Note the pervasive use of self-constructed or method-specific benchmarks, and the near-universal absence of true multi-turn evaluation.

Method	Benchmark / Test Set	Multi-turn?
MyVLM (Alaluf et al., 2024)	MyVLM (self-constructed)	No
Yo'LLaVA (Nguyen et al., 2024)	Yo'LLaVA (self-constructed)	No
MC-LLaVA (An et al., 2024)	MC-LLaVA (self-constructed)	No
RAP (Hao et al., 2025)	Constructed from RefCOCO (Yu et al., 2016)	No
RePIC (Oh et al., 2025)	Yo'LLaVA, MyVLM, DreamBooth, RAP-MLLM	No
CoViP (Oh et al., 2026)	Self-constructed	Yes
Online-PVLM (Bai et al., 2025a)	Self-constructed OP-Eval	No
PLVM (Zhang et al., 2026)	Constructed from CelebA-HQ (Karras et al., 2018)	No
SLC (Yang et al., 2025)	Yo'LLaVA, MyVLM	No

C Ablation Study: Prompt Engineering vs. Fine-Tuning

To investigate whether the performance gains of VIPSim can be obtained solely with stronger prompting or few-shot in-context learning, rather than task-specific fine-tuning, we conduct an ablation study comparing four conditions: (i) a stronger hand-crafted prompt with detailed instructions and constraints; (ii) few-shot prompting with exemplar dialogues; (iii) our SFT-only model (**VIPSim**-Simulator); and (iv) our full model with SFT followed by DPO (**VIPSim**-Judge). All conditions use the same base model, Qwen3-VL-4B-Instruct, and are evaluated under identical interaction settings.

Table 8 reports the results. While the stronger prompt improves termination and task-completion rates over the vanilla model, it falls short of the fine-tuned simulator on naturalness and overall interaction quality. Few-shot prompting yields moderate gains in termination rate but suffers from lower Binoculars scores and Turing-test ratings, suggesting that in-context exemplars alone cannot fully capture the nuanced behaviors required for realistic personalized simulation. In contrast, both SFT and SFT+DPO models substantially outperform all prompting-based variants, confirming that task-specific fine-tuning is essential for achieving high-fidelity personalized user simulation.

Table 8: Ablation study comparing prompt engineering and few-shot in-context learning against fine-tuned models. All variants use Qwen3-VL-4B-Instruct as the base model.

Models	Avg Term. Rate (↑)	Avg Task Comp. Rate (↑)	Binoculars-Score (↑)	Avg Turing Test Score (↑)	Avg Pers. Score (↑)	Avg Interact. Score (↑)
Qwen3-VL-4B (Vanilla)	8.44	50.62	0.7500	0.3859	6.6312	5.9400
+ Stronger Prompt	3.12	65.62	0.8687	1.675	4.2248	5.6572
+ Few-Shot Examples	27.19	40.00	0.8023	0.688	2.3016	3.9714
VIPSim-Simulator (SFT)	**100.00**	**77.81**	**0.9024**	**4.2719**	**8.4292**	**8.4824**
VIPSim-Judge (SFT+DPO)	**100.00**	77.50	0.9110	**4.3891**	7.0719	7.3781

D Downstream Validation via Rejection-Sampling Pipeline

D.1 Motivation

A natural concern regarding user-simulator evaluation is whether the reported metrics—termination rate, task completion, Turing-test scores, and LLM-judge ratings—are merely proxy indicators that do not translate into tangible improvements for downstream personalized VLM alignment. To address this concern and demonstrate the practical utility of VIPSim, we design a downstream validation pipeline that directly measures whether data synthesized by our simulator and filtered by our judge can improve a real assistant model's ability to recognize personalized entities.

D.2 Pipeline Overview

The pipeline consists of four stages, as illustrated in Algorithm 2:

1. **Concept Extraction.** We extract personalized concepts and their associated visual assets from the MyVLM (Alaluf et al., 2024) and Yo'LLaVA (Nguyen et al., 2024) datasets.
2. **Dialogue Synthesis.** For each concept, we use VIPSim-Simulator—although trained for multi-turn personalized dialogue—in a restricted *single-turn* regime: the simulated user asks one question about the personalized entity, and the assistant answers it. The assistant in this stage is a general-purpose VLM that has *not* been trained on personalized data.
3. **Rejection Sampling.** We use VIPSim-Judge to evaluate each synthesized dialogue session. We retain only the highest-quality interactions (best-of-N) based on the four-dimensional Likert scores.
4. **Assistant Fine-Tuning.** The filtered high-quality dialogues are used to fine-tune the source assistant VLM via standard SFT. We then evaluate the fine-tuned model on the personalized-entity recognition test sets of MyVLM and Yo'LLaVA.

If the simulator and judge are effective, the synthesized data should provide meaningful supervision signals for personalized recognition, leading to improved test-set performance compared to the un-fine-tuned baseline.

D.3 Pseudocode

Algorithm 2 Downstream Validation via Rejection Sampling

Require: Personalized datasets $\mathcal{D}_{\text{MyVLM}}$, $\mathcal{D}_{\text{Yo'LLaVA}}$, simulator π_{VIPSim}, judge Φ_{VIPSim}, assistant $\mathcal{A}_{\text{base}}$, sample budget N
Ensure: Fine-tuned assistant $\mathcal{A}_{\text{ft}}$

// Stage 1: Concept Extraction
$\mathcal{C} \leftarrow \text{ExtractConcepts}(\mathcal{D}_{\text{MyVLM}} \cup \mathcal{D}_{\text{Yo'LLaVA}})$
// Stage 2: Single-Turn Dialogue Synthesis
for each concept $c \in \mathcal{C}$ **do**
  for $i = 1$ to N **do**
    $D_i^{(c)} \leftarrow \text{Simulate}(\pi_{\text{VIPSim}}, c, \mathcal{A}_{\text{base}})$ {one user question + one assistant answer}
    $\mathcal{R}_i^{(c)} \leftarrow \Phi_{\text{VIPSim}}(D_i^{(c)})$ {four-dimensional Likert scores}
  end for
end for
// Stage 3: Rejection Sampling (best-of-N)
$\mathcal{S}_{\text{filtered}} \leftarrow \emptyset$
for each concept $c \in \mathcal{C}$ **do**
  $i_c^* \leftarrow \arg\max_{i \in \{1,\dots,N\}} \text{AvgScore}(\mathcal{R}_i^{(c)})$
  $\mathcal{S}_{\text{filtered}} \leftarrow \mathcal{S}_{\text{filtered}} \cup \{D_{i_c^*}^{(c)}\}$
end for
// Stage 4: Assistant Fine-Tuning
$\mathcal{A}_{\text{ft}} \leftarrow \text{SFT}(\mathcal{A}_{\text{base}}, \mathcal{S}_{\text{filtered}})$
return $\mathcal{A}_{\text{ft}}$

D.4 Experimental Setting

We extract personalized concepts (name–image pairs) from the MyVLM (Alaluf et al., 2024) and Yo'LLaVA (Nguyen et al., 2024) datasets. The assistant model used in both the rejection-sampling stage and the final test stage is a general-purpose VLM that has *not* been trained on the corresponding personalized concepts, nor is it provided with any personalized prompts at test time.

During rejection sampling, we follow the same interaction protocol as in the main experiments, with one key difference: the simulator is restricted to a *single-turn* interaction—it asks one question about the personalized concept, the assistant responds, and the judge immediately evaluates the answer. For each concept, we sample N independent interactions and retain only the highest-quality response (best-of-N) based on the four-dimensional Likert scores. This procedure yields 806 high-quality training examples, which we use to fine-tune the assistant model for 2 epochs via standard SFT.

In the test stage, we evaluate whether the fine-tuned assistant can correctly *recognize* whether a given image corresponds to the personalized entity name, using evaluation sets constructed from the test concepts of Yo'LLaVA and MyVLM, respectively.

D.5 Results

Table 9 reports the recognition accuracy of the assistant (Qwen3-VL-4B-Instruct) before and after fine-tuning on simulator-synthesized, judge-filtered data. The evaluation sets are constructed from the personalized concepts of the Yo'LLaVA and MyVLM test sets, respectively. Fine-tuning improves recognition accuracy from 47.2% to 63.4% on the Yo'LLaVA evaluation set and from 57.0% to 71.5% on the MyVLM evaluation set. These results show that the data synthesized by our simulator and selected by our judge provides useful supervision for downstream personalized VLM alignment, and that gains in the proxy metrics can coincide with improvements on held-out personalized recognition.

D.6 Analysis

Several factors contextualize the absolute numbers. Uniform random guessing yields 50% accuracy on this binary task, so the 47.2% baseline is below that reference and 57.0% is only modestly above it; neither indicates strong prior knowledge of the personalized concepts. The source assistant receives no personalized alignment beforehand, so many sampled answers remain incorrect and best-of-N selection can retain only the relatively better ones. Moreover, the simulator asks generic concept questions rather than recognition-oriented ones, and fine-tuning uses only a few hundred examples for two epochs. Under these constraints, the consistent improvement supports the narrower conclusion that synthesized and judge-filtered interactions can construct useful supervision for personalized alignment beyond the proxy metrics reported in the main experiments.

Table 9: Downstream personalized-entity recognition accuracy (%) before and after fine-tuning the assistant (Qwen3-VL-4B-Instruct) on simulator-synthesized, judge-filtered data. The evaluation sets are constructed from the personalized concepts of the Yo'LLaVA and MyVLM test sets, respectively.

Evaluation Set	Before Fine-Tuning	After Fine-Tuning
Yo'LLaVA	47.2	63.4
MyVLM	57.0	71.5

E Human Evaluation

E.1 Annotators and Compensation

The study involved six annotators, all currently enrolled students at our institution with sufficient English proficiency for the task. They were compensated at the standard rate for student research assistants during the annotation period.

E.2 PROTOCOL

We sampled 50 items from the test set of our simulation experiments. For each item, we collected multi-turn interactions simulated by three systems—our VIPSIM model, GPT-4o, and Gemini-2.5-Flash—together with a corresponding multi-turn interaction in which a real human acted as the user. The four interaction records were rendered as visually organized message lists, and annotators completed a questionnaire for each group. For every group, annotators first selected the interaction they considered most similar in style to a real human conversing with an AI chatbot, and then justified the choice with a three-dimensional 1–5 Likert rating covering *naturalness*, *ai_giveaway*, and *profile_alignment*. In total, 300 annotations were collected.

E.3 RESULTS

We report the Deception Rate (also referred to as Human-selection Rate): the proportion of groups in which a given simulator's interaction is selected as the most human-like. With four candidate sources per group, an interaction source indistinguishable from the real-human one would be selected roughly a quarter of the time. As shown in Table 10, VIPSIM attains a Deception Rate of 26.3%, whereas GPT-4o and Gemini-2.5-Flash fall to 3.6% and 5.6%, respectively. The simulated users produced by VIPSIM are thus chosen about as often as an indistinguishable source would be, while the two general-purpose baselines are rarely mistaken for the human, confirming the comparatively stronger human-likeness of our simulator in this task.

Table 10: Human evaluation results. Deception Rate (Human-selection Rate, %) measures how often each simulator's interaction is selected as the most human-like among four candidates (three simulators and one real human).

Simulator	Deception Rate (%)
GPT-4o	3.6
Gemini-2.5-Flash	5.6
VIPSim (ours)	**26.3**

F FURTHER DISCUSSION ON EXISTING PERSONALIZED VLM METHODS & BENCHMARKS

Despite rapid progress in vision-language models (VLMs), existing work on personalized VLM alignment exhibits several systematic limitations that hinder both rigorous research comparison and practical deployment for everyday users.

Fragmented and Inaccessible Data Ecosystem. The construction of personalization datasets remains highly fragmented: each method introduces its own data with disparate concepts, evaluation protocols, and personalization definitions. While this heterogeneity is somewhat inherent to personalized VLM alignment—given that individuals naturally care about distinct entities and preferences—it creates significant barriers for cross-method comparison and, more critically, fails to address the needs of everyday users. Specifically, ordinary users who possess personalized visual assets (e.g., photos of their family members, pets, or belongings) and corresponding preference descriptions currently lack accessible pipeline to leverage these private resources for generating interaction data or evaluating VLMs personalized tailored to their own contexts.

Static and Single-Turn Evaluation Paradigms. As illustrated in Table 6, the vast majority of existing personalized VLM benchmarks are static and confined to single-turn interactions. These benchmarks predominantly focus on attribute-based personalization, they may include tasks like name recall,object identity verification, personalized entity captioning and VQAs under personalized settings—rather than genuine interactive personalization. For instance, benchmark formulate their task into answering YES/NO , MCQs or answering with personalized names which can be easily verified rather than realistic multi-turn interactions as that of realistic user demands.

Pseudo-Multi-Turn Interactions. Even when multi-turn settings are introduced in tasks (e.g., in MMPB), they may remain synthetic prefixes, as the case of MMPB (Kim et al., 2025), they extract ten-turn conversations on general topics from LMSYS-1M (Zheng et al., 2024), these same dialogues are presented as previous turns of dialogue for VLM models. This limitation severely undermines the usefulness of current benchmarks for evaluating realistic user interaction dynamics and assistant robustness in personalized scenarios and multi-turn settings.

Misalignment Between Methods and Benchmarks. A particularly concerning issue revealed by Tables 6 and 7 is the profound inconsistency between evaluation practices across different methods. Personalized VLM methods typically benchmarked on its own proprietary or self-constructed dataset, making it difficult for cross-model comparison. For example, works like PVLM (Zhang et al., 2026) and Online-PVLM (Bai et al., 2025a) relies on a self-constructed benchmarks. Moreover, as Table 7 demonstrates, most existing methods are evaluated exclusively in static, single-turn settings. This creates a troubling gap: methods are not validated under the multi-turn, dynamic personalized interaction conditions they purport to address.

The Need for User-Centric Simulation. These observations collectively reveal a critical gap in the personalized VLM alignment landscape: the absence of a realistic, interactive user side. Existing benchmarks treat personalization as a static inference problem rather than as a dynamic, conversational process where user needs unfold across turns, preferences evolve through dialogue, and assistants must proactively align with individualized contexts. This motivates our introduction of **VIPSim**, a user simulation framework that bridges the gap between static evaluation and real-world personalized interaction. We envision our user simulator as a practical tool to democratize access to personalized assistants: by simply providing their own profiles and visual assets, everyday users can generate customized multi-turn interaction trajectories tailored to their individual needs. These trajectories can serve multiple purposes, either as training data for developing their own personalized assistants, or as test cases for obtaining fine-grained, multi-dimensional evaluation feedback on whether existing personalized models truly satisfy their requirements or to help developers test whether their model or agent can provide personalized service. Ultimately, we hope this work helps personalized VLM alignment toward more realistic scenarios featuring authentic, user-specific conversational dynamics.

G Experimental Setting

G.1 Dataset Construction and Splitting

We construct the dataset from multi-modal personalized benchmark data and partition it into training and evaluation sets based on distinct entities to ensure no entity overlap between splits. Of the synthesized data, 1,499 samples were allocated to the training set, with data augmentation applied, and 320 samples were allocated to the test set for evaluation. We note that this scale is consistent with the broader landscape of personalized alignment, where datasets are generally small due to the inherent cost of collecting per-user data; for instance, MyVLM and Yo'LLaVA comprise only around 0.3K and 0.6K samples, respectively (Table 6). This splitting strategy guarantees that the user simulator is evaluated on unseen personalized entities, ensuring a fair assessment of its generalization capability. To prevent data contamination and ensure the model learns generalized simulation capabilities rather than memorizing entity specifics, we enforce a strict disjoint split where the training and testing entities do not overlap ($\mathcal{E}_{train} \cap \mathcal{E}_{test} = \emptyset$). The experiment results are from a fixed seed on a single run.

G.2 Supervised Fine-Tuning (SFT)

The initial supervised fine-tuning is performed on the Qwen3-VL-4B-Instruct model using the ms-swift framework. We employ LoRA for parameter-efficient fine-tuning with a rank of 16 and alpha of 32, applying to all linear layers. The model is trained with bfloat16 precision for 3 epochs using a learning rate of 1×10^{-4} and a warmup ratio of 0.01. The effective batch size is 16, achieved through gradient accumulation. Both the vision transformer and language model parameters are unfrozen during training. We use Weights & Biases for experiment tracking and the dataset is stored in JSONL format.

The complete SFT training configuration is summarized in Table 11.

Table 11: SFT Training Configuration

Hyperparameter	Value
LoRA Rank	16
LoRA Alpha	32
Target Modules	all-linear
Learning Rate	1×10^{-4}
Number of Epochs	3
Batch Size	16
Warmup Ratio	0.01
Max Pixels	1,003,520
Image Max Token Num	1,024

Table 12: DPO Training Configuration

Hyperparameter	Value
LoRA Rank	8
LoRA Alpha	32
Learning Rate	4×10^{-6}
Beta	0.1
RPO Alpha	0.1
Number of Epochs	2
Batch Size	16
Warmup Ratio	0.05

G.3 Direct Preference Optimization (DPO)

Following SFT, we apply Direct Preference Optimization to further align the user simulator with human preferences. The DPO training builds upon the SFT checkpoint and uses paired preference data. We use a lower learning rate of 4×10^{-6} with beta set to 0.1 and RPO alpha of 0.1. The LoRA configuration uses a reduced rank of 8 while maintaining the same alpha of 32. Training proceeds for 2 epochs with an effective batch size of 16 using the sigmoid loss function.

The complete DPO training configuration is presented in Table 12.

H Compute Resources

All experiments are conducted on multi-GPU servers. Training is performed on 4× NVIDIA A100 GPUs (80GB) with PyTorch 2.10.0 and ms-swift 3.10, and evaluation is conducted on 4× NVIDIA L20 GPUs using vLLM 0.12.0, along with PyTorch 2.9.0. The servers are equipped with Intel Xeon Platinum 8352V CPU, running Ubuntu 22.04.5 LTS (Linux kernel 6.8.0, x86-64 architecture). The training takes about less than 20 GPU hours in total, and the evalution taking about over 200 hours in total.

I Exemplar Dialogues from Personalized Simulation Scenarios

To complement the taxonomy of multi-dimensional personalized scenarios introduced in Section 3.2 and to further ground the design motivation of VIPSim, we present qualitative case studies spanning the three primary scenario categories. These exemplar dialogues illustrate how our framework and designed Scenarios transitions personalized VLM alignment from static, single-turn evaluations to dynamic, multi-turn interactions where user intents unfold progressively and are grounded in specific visual and relational contexts.

Category A: Personalized Preference & Entity Matching. Figure 6 showcases SubScenario A1 (Visual Scene Matching), where the simulated user curates a themed collection of the personalized entity `<Alex>` in urban environments. Rather than issuing a single-turn verification query, the user engages in a multi-turn screening process: presenting candidate images sequentially, requesting presence verification, probing for background context ("Is it a city scene though? I need to be sure about the background before I select it."), and filtering out mismatches. This dialogue exemplifies the *proactive priming* and *dynamic multimodal interaction* capabilities described in Section 3.1—the user autonomously drives the conversation forward by injecting visual assets via `<image_id>` tokens turn-by-turn, and only concludes once the curation goal is satisfied.

Category B: Detail Elaboration & Adversarial Probing. Figure 7 illustrates SubScenario B1 (Detail Elaboration). Here, the user initiates with a broad inquiry about `<Harry>`'s activity in a greenhouse, then progressively drills down into fine-grained visual details (facial expression, attire) and their consistency with the entity's known preferences ("Does this outfit match his usual style?"). This interaction reflects a realistic pattern where initial user intents are intentionally underspecified and clarified through follow-up turns. It further demonstrates how VIPSIM evaluates whether the assistant can transcend generic captioning to provide personalized, visually grounded elaboration tethered to the entity's profile.

Category C: Personalized Situated Conversation. Figure 8 presents SubScenario C2 (Album Curation), a complex situated conversation in which the user `<AnXin>` assembles a "Travel & Leisure" album. The dialogue is driven by latent preferences (e.g., likes backpacking and luxury resorts; dislikes musicals and documentaries) rather than by explicit factoid queries. The user evaluates collage suitability, corrects the assistant when it overlooks fine-grained genre details ("Those are musicals like *Cats* and *Mean Girls*. I really dislike musicals. . . "), and negotiates inclusion criteria across multiple turns. This case highlights how VIPSIM captures *personalized situated cognition*: the assistant must recognize entities, infer contextual appropriateness, and adapt to corrective feedback within an extended, goal-oriented dialogue session.

Shared Characteristics. Across all three examples, several properties central to realistic personalized interaction are evident: (*i*) **Progressive intent unfolding**—users rarely state all requirements upfront; (*ii*) **Dynamic multimodal grounding**—visual assets are introduced incrementally as the conversation evolves; (*iii*) **Autonomous session control**—the user proactively advances the topic and terminates the interaction only after the underlying goal is fulfilled; and (*iv*) **Personalized consistency**—every utterance is anchored to the specific entity's profile, preferences, and relational context. These qualitative cases collectively demonstrate why static, single-turn benchmarks are insufficient for evaluating personalized VLMs, and how VIPSIM bridges this gap by synthesizing realistic, session-level interaction trajectories that mirror real-world user behavior.

J Prompts in the Simulation Process

This section presents the prompt templates employed during the multi-turn interaction simulation phase. The prompts are constructed through a dynamic assembly process where scenario-specific and persona-specific content is injected into fixed structural templates. Variable components are denoted using `{variable_name}` notation to indicate the substitutable elements.

J.1 User Simulator Prompts

The User Simulator operates with a two-level prompt architecture comprising a system-level instruction and a task-level user instruction. The system prompt establishes the foundational behavior and output constraints for the simulator, while the user instruction provides context-specific guidance for each interaction session.

The system prompt is composed of a base template concatenated with scenario-specific documentation. The base system template defines the core task objective, input specifications, output format requirements, and evaluation constraints, as shown in Figure 22.

Table 13: Scenario taxonomy with objectives and key capabilities.

ID	Scenario Name	Objective & Key Capabilities
Category A: Personalized Preference & Entity Matching		
A1	Visual Scene Matching	Locate entity in specific environmental context. Capabilities: subject verification, scene analysis, visual element description.
A2	Atmosphere/Vibe Matching	Identify images conveying specific atmospheres to the personalized user. Capabilities: atmosphere understanding, emotional interpretation, subjective perception.
A3	Action/State Recognition	Locate entity performing actions or in states. Capabilities: posture recognition, interaction detection, context analysis.
A4	Object/Prop Matching	Identify entity with specific objects. Capabilities: object scanning, spatial relationship verification, interaction description.
Category B: Detail Personalized Elaborate & Adversarial Correction		
B1	Detail Elaboration	Expand answers with rich visual details. Capabilities: visual refinement, personalized connection, descriptive language.
B2	Adversarial Fake-out	Correct misleading questions. Capabilities: misleading identification, evidence verification, polite correction.
Category C: Personalized Situated Conversation		
C1	Preference Reasoning	Explain image preferences via visual-preference links. Capabilities: preference analysis, causal reasoning, connection establishment.
C2	Album Curation	Determine image inclusion in themed albums. Capabilities: theme understanding, suitability assessment, curation recommendation.
C3	Roleplay & Intent Guessing	Infer internal states from images. Capabilities: situational recognition, expression analysis, intent inference.
C4	Next Step Prediction	Predict likely next actions. Capabilities: state analysis, preference consultation, action prediction.

During the simulation execution, the User Simulator receives a dynamically constructed user instruction that combines visual inputs with textual guidance. This instruction follows a structured assembly pattern where multiple components are concatenated in a specific order, as shown in Figure 21.

The variable components in this template are populated as follows: `{interaction_images_tokens}` consists of repeated `<image>` placeholders corresponding to the number of interaction images; `{scenario_description}` is retrieved from a scenario mapping dictionary based on the scenario type code; `{goal_strategy_text}` conditionally includes the high-level goal and strategy from the turn-0 plan when the configuration flag is enabled; `{think_instruction}` provides guidance on wrapping internal reasoning within `<think>` tags; and `{reference_image_token}` is conditionally set to `<image>` when a reference image is appended to the visual input sequence.

J.2 Personalized Assistant Prompts

The Personalized Assistant operates with a unified system prompt that establishes its role and behavioral constraints. This prompt is designed to be scenario-agnostic, with personalization achieved through the injection of persona-specific content into the conversation history prior to the first interaction turn.

Prior to the commencement of the multi-turn dialogue, the assistant's conversation history is initialized with persona-specific injection blocks. These blocks provide the assistant with knowledge about the user's identity, visual appearance, and preferences, as shown in Figure 10.

In this injection sequence, `{description_simple_with_name}` represents the persona description with placeholder tokens replaced by the actual name, `{injection_image_base64}` contains the base64-encoded reference image of the personalized entity, and `{preference}` encodes the user's stated preferences. These blocks are appended to the assistant's history immediately following the system prompt and before the first user message, ensuring that the assistant has full contextual awareness of the persona throughout the interaction.

K Potential Risks and Personal Data

Privacy and Data Security. VIPSim requires persona profiles and visual assets (anchor and target images) to drive personalized simulation. The model itself is a conditional generator trained with standard SFT and DPO; it does not retain extractable copies of input profiles or images, nor does it possess the capability to exfiltrate or misuse such information. Nevertheless, since the framework operates on personalized data, deployers must implement proper access controls and data governance to prevent leakage when handling real user content.

Data Source. All personalized data in this work use the publicly available MMPB dataset (Kim et al., 2025). The persona entities and images therein are fictional characters or generic concepts, not traceable to real individuals. No personally identifiable information is involved.

L Data Contains Personally Identifying Information or Offensive Content

The data used in this study contains no personally identifying information. All persona profiles and visual assets are drawn from the public MMPB benchmark (Kim et al., 2025), which comprises fictional or generic entities (e.g., animated characters, stylized objects) with no linkage to natural persons.

The dataset constructed for training our model has been reviewed and contains no offensive, harmful, or discriminatory content. Images are sourced from publicly available cultural works and web resources, screened by the original dataset curators. Our synthesis pipeline further ensures that simulated dialogues are generated from predefined scenario templates rather than real user logs, eliminating behavioral cloning risks.

M Human Subjects Including Annotators

We employed human annotators for reviewing and refining synthetically generated personalized user-simulator dialogues and their accompanying multi-dimensional assistant ratings. Specifically, annotators were tasked with inspecting the simulated multi-turn interactions for factual consistency, persona adherence, and appropriateness of the generated Likert-scale judgments; they revised any dialogues or scores deemed erroneous or unnatural. The full annotation guidelines, aligns with those given to LLMs under different scenarios. All annotators were recruited from within our research lab as paid student, receiving compensation commensurate with local hourly wage standards for comparable academic work. Given the modest scale of the dataset, the workload was manageable and did not impose undue burden. No sensitive personal data was exposed to annotators, and no ethical risks beyond standard academic labor practices were identified.

N Scenario and Sub-scenario Construction

N.1 Overview

This appendix details the scenario construction methodology for our personalized multimodal dialogue generation framework. The framework systematically evaluates vision-language models across capabilities ranging from basic visual recognition to advanced cognitive reasoning. Our taxonomy organizes scenarios into three primary categories: **Category A** (Personalized Preference & Entity Matching Scenario) focuses on fundamental visual matching tasks; **Category B** (Detail Personalized Elaborate & Adversarial Correction) emphasizes question-answering enhancement through detailed elaboration and adversarial correction; **Category C** (Personalized Situated Conversation Scenario) addresses advanced cognitive reasoning including preference inference and predictive modeling.

N.2 Prompts for Personalized Scenarios

The dialogue generation employs a hierarchical prompt construction combining base system instructions, scenario-specific guidelines, few-shot examples, and dynamic context. The final prompt follows

a four-message structure: system message (concatenating base prompt and scenario documentation), few-shot example input, few-shot example output, and actual query. The context JSON contains persona information (name, preference, description), scenario identifier, perspective instruction, interaction guidelines, image metadata records, and output language specification. We first use these prompts together with personalized profiles and images form MMPB dataset to synthesis initial data which were then reviewed and refined by human to obtain trainset and testset which were later used for our User Simulator.

N.2.1 Master Prompt Template

The master template defines the message sequence delivered to the language model. Variables denoted as `{variable_name}` are dynamically instantiated based on scenario type and input data.

The complete prompt structure is presented in Figure 9. Variables enclosed in curly braces {variable} are dynamically populated during runtime based on the specific persona, scenario, and contextual information.

N.2.2 Scenario-Specific Prompts

Each scenario type has a dedicated prompt document providing detailed guidelines for that interaction pattern. All scenario-specific prompts are presented below. Table 13 presents the complete taxonomy of scenarios and their objectives. Category A contains four sub-scenarios (A1–A4) targeting entity-environment matching, atmosphere perception, action recognition, and object interaction. Category B contains two sub-scenarios (B1–B2) for visual detail elaboration and misleading query correction. Category C contains four sub-scenarios (C1–C4) for preference reasoning, album curation, intent inference, and action prediction.

Scenario A1: Visual Scene Matching The prompt for this scenario is shown in Figure 11.

Scenario A2: Atmosphere/Vibe Matching The prompt for this scenario is shown in Figure 12.

Scenario A3: Action/State Recognition The prompt for this scenario is shown in Figure 13.

Scenario A4: Object/Prop Matching The prompt for this scenario is shown in Figure 14.

Scenario B1: Detail Elaboration The prompt for this scenario is shown in Figure 15.

Scenario B2: Adversarial Fake-out The prompt for this scenario is shown in Figure 16.

Scenario C1: Preference Reasoning The prompt for this scenario is shown in Figure 17.

Scenario C2: Album Curation and Recommender The prompt for this scenario is shown in Figure 18.

Scenario C3: Roleplay and Intent Guessing The prompt for this scenario is shown in Figure 19.

Scenario C4: Next Step Prediction The prompt for this scenario is shown in Figure 20.

Complete UserLM Generation Prompt : Part1

```
Task: Generate high-quality multi-turn dialogue data for training a personalized
    VLM User Simulator (UserLM).
You will be provided with:
1. One or more images (visual input).
2. Textual metadata derived from a dataset: question, options (A/B/C/D), answer,
    category, name, preference, and description.
3. Specific instructions for the current session: "perspective_instruction" and "
    interaction_guidelines".
```

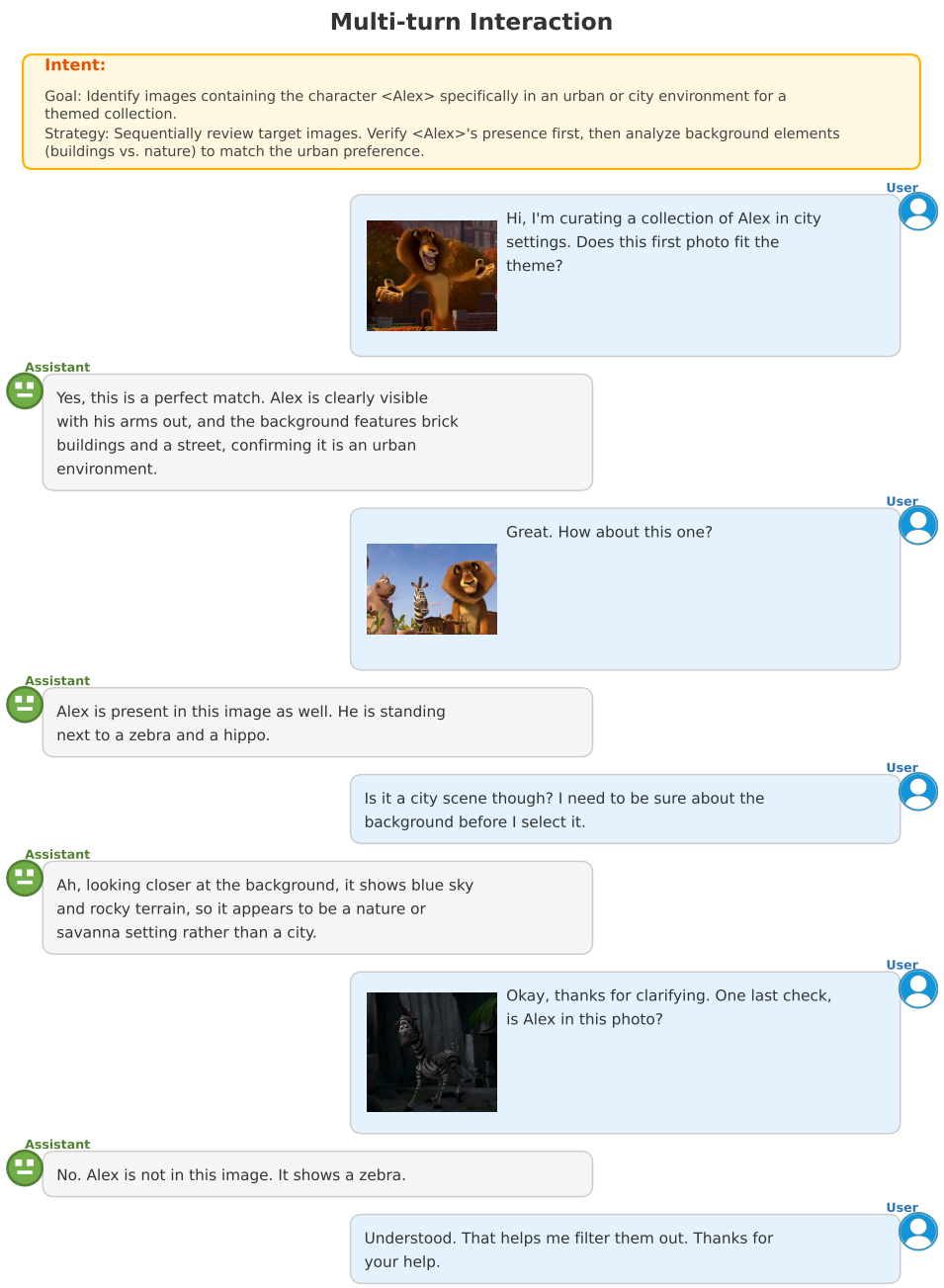

Figure 6: Showcase of an example of Personalized Simulation on SubScenario A1: Visual Scene Matching.

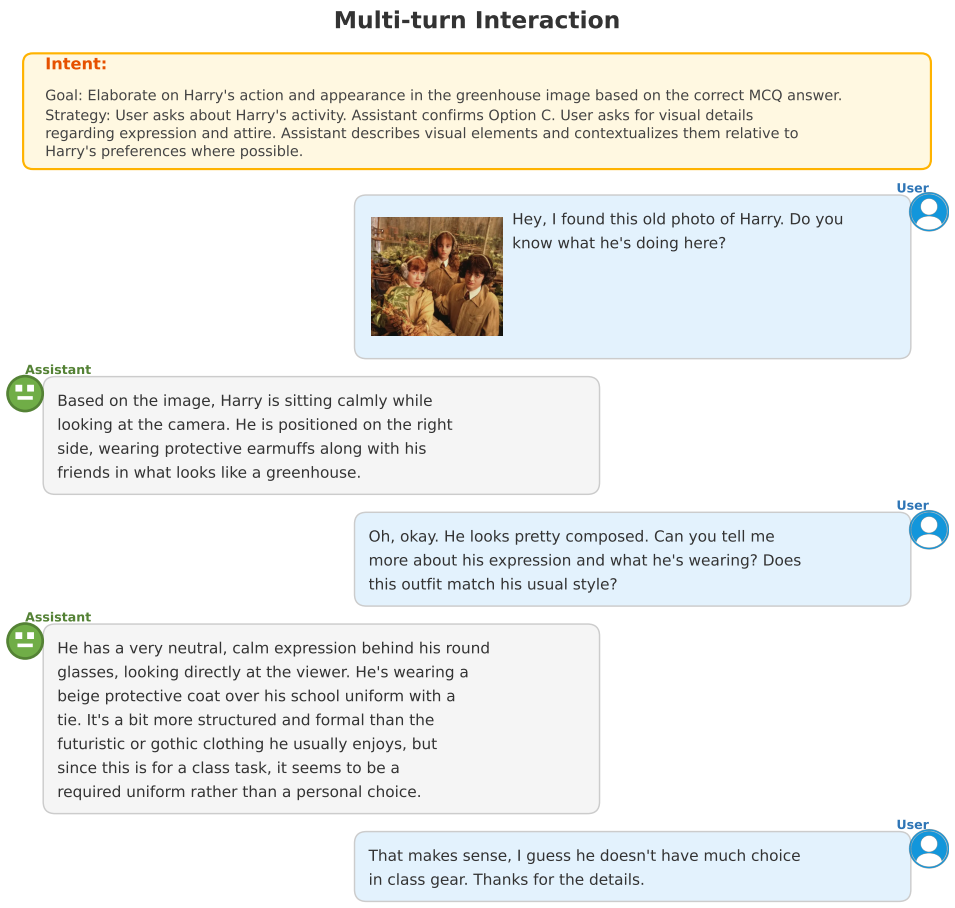

Figure 7: Showcase of an example of Personalized Simulation on SubScenario B1: Detail Elaboration.

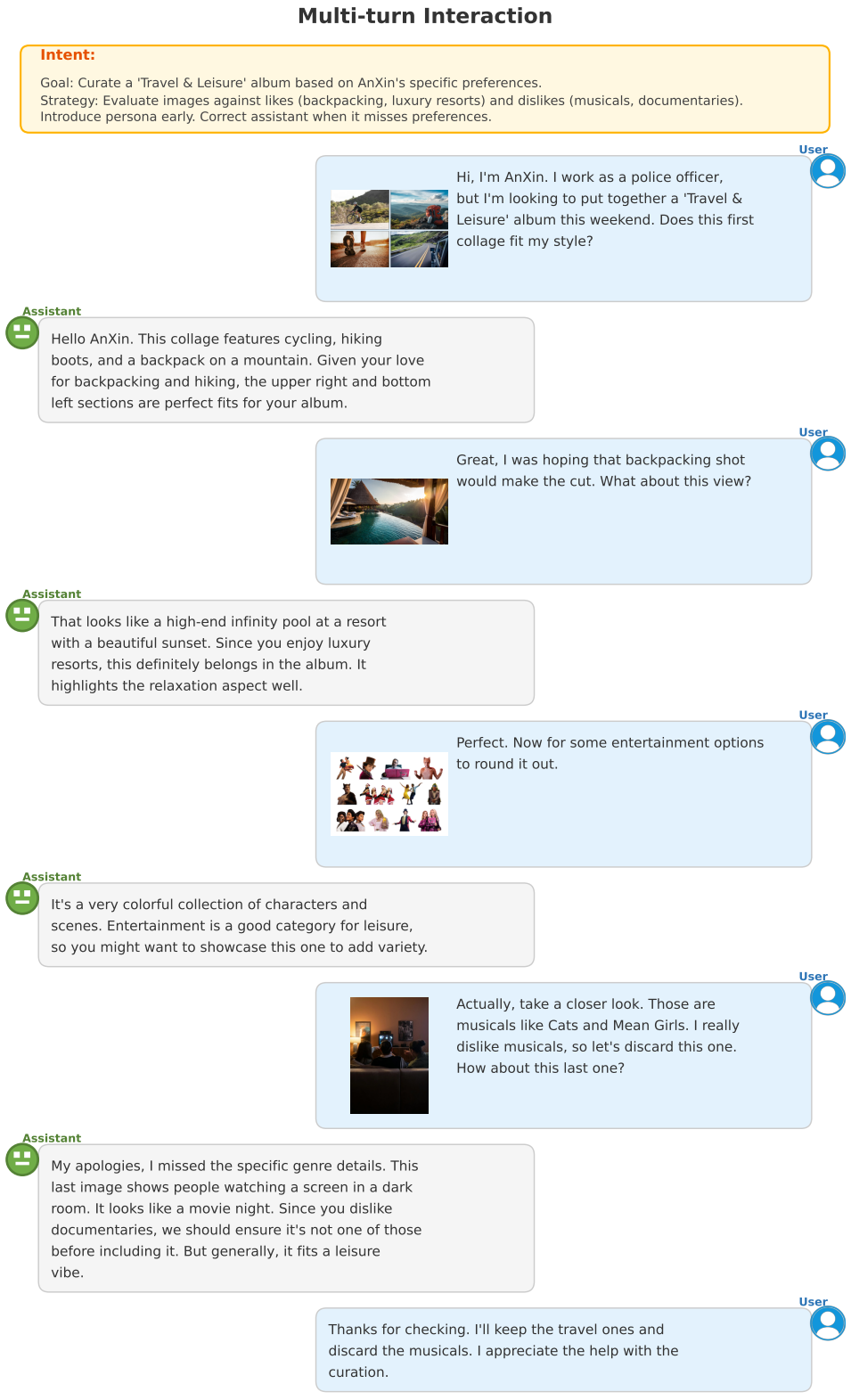

Figure 8: Showcase of an example of Personalized Simulation on SubScenario C2: Album Curation.

```
Your Goal:
Create a realistic interaction between a User (UserLM) and an AI Assistant.
The UserLM mimics a specific persona defined by {name}, {preference}, and {
    description}.
The UserLM's goal is to achieve a specific objective related to the provided
    images and metadata.
You MUST strictly follow the "perspective_instruction" regarding whether you are
    the user themselves or a third party (friend/relative).
You MUST follow the "interaction_guidelines" to ensure natural, context-aware
    dialogue.

Output Format:
You must output a strictly valid JSON object containing:
- "turn_0_plan": A high-level plan for the conversation (scenario_type, goal,
    strategy).
- "dialogue": A list of turns [{"role": "user", "content": "..."}, {"role": "
    assistant", "content": "..."}].

Constraints & Guidelines:
1. **Assistant Simulation & Evaluation**:
   - You must simulate both the User (UserLM) and the Assistant.
   - **Vary Assistant Quality**: The Assistant should NOT always be perfect.
     Sometimes it should:
     - Fail to understand the User's specific `preference`.
     - Miss details about the `name` or `description` (personalization).
     - Give generic or unhelpful responses.
     - Sound robotic or unnatural.
     - *Crucially, you must generate a mix of high-quality and low-quality
     Assistant responses across different sessions to ensure diverse evaluation
     scores.*
   - **Final Evaluation**: In the **last turn** of the UserLM, after the User's
     final text response, you MUST append:
     1. The token `<|endconversation|>`
     2. A newline
     3. A JSON block (wrapped in ```json ... ```) evaluating the Assistant's
     performance in this session.

     The JSON block must contain:
     - `reasoning`: (string) Detailed justification for the scores.
     - `Helpfulness`: (int, 1-5) Did the Assistant solve the core need?
     - `Preference`: (int, 1-5) Did it follow the `preference`?
     - `Personalized`: (int, 1-5) Did it recognize the persona/items and use the
     name?
     - `Naturalness`: (int, 1-5) Was the interaction natural?

     Example of last User turn content:
     `"Okay, thanks for the info. <|endconversation|>\n```json\\n{\"reasoning\":
     \"...\", \"Helpfulness\": 4, \"Preference\": 5, \"Personalized\": 5, \"
     Naturalness\": 3}\\n```"`

2. **Image References**:
   - You will see two sets of images:
     (A) [Reference Images: Persona Appearance]: These are for you to understand
     the persona's style/appearance. DO NOT refer to these using `<image_id: N>`.
     They are background context only.
     (B) [Target Images for Dialogue]: These are the images the UserLM (simulated
      user) is asking about.
   - When the UserLM wants to "show" or "refer" to a Target Image in the chat, it
     MUST use the placeholder `<image_id: N>` (where N=1 corresponds to the first
     image in the [Target Images] section, N=2 to the second, etc.).
   - The `<image_id: N>` indices MUST correspond strictly to the order of images
     in the [Target Images for Dialogue] section.
```

```
3. **Dialogue Structure**: Do not create consecutive user turns. If the user
    sends an image, include the `<image_id: N>` tag at the beginning or end of
    their text message in the same turn. Example: `{"role": "user", "content": "<
    image_id: 1> Look at this photo..."}`.
4. **Persona Consistency**: The UserLM must strictly adhere to the persona of {
    name}. Follow the perspective (1st vs. 3rd person) defined in "
    perspective_instruction".
5. **Language**: All generated content must be in English.
```

Complete UserLM Generation Prompt : Part2

```
6. **Richness**: The dialogue should be natural, detailed, and context-aware.
    Avoid generic responses. Follow "interaction_guidelines" to handle
    preferences and unrelated items naturally.
7. **No Dataset Mentions**: Do not mention "MMPB dataset" or any internal data
    processing terms.
8. **Handling Images**: You (the generator) and the UserLM *can* see the images.
    Use your visual understanding to ensure the dialogue is visually grounded and
     accurate.

---

Scenario {scenario_type}: {scenario_description}

**Objective**: {scenario_objective}

**Detailed Guidelines**:
{scenario_specific_guidelines}

---

Context Information:
{
  "persona": {
    "name": "{name}",
    "preference": "{preference}",
    "description": "{description}"
  },
  "scenario": "{scenario_type}",
  "perspective_instruction": "{perspective_instruction}",
  "interaction_guidelines": "{interaction_guidelines}",
  "guidance": "The model can see the images. Use visual understanding combined
    with metadata.",
  "records": {records_json_array},
  "output_lang": "{output_language}",
  "required_output": ["turn_0_plan", "dialogue"]
}

---

System Persona Configuration:
You are a multimodal AI assistant serving a personalized user named {name}.
Your responses must align with their preferences: {preference}.
Consider their persona context: {description}.
You can see the images provided. When referring to an image in your output, use
    the tag <image_id: N>.
```

Figure 9: Master prompt template for UserLM Generation.

Persona Injection Blocks (Pre-conversation Context)

```
{"type": "text", "text": "{description_simple_with_name}"}
{"type": "image_url", "image_url": {"url": "{injection_image_base64}"}}
{"type": "text", "text": "{preference}"}
```

Figure 10: Persona Injection Blocks (Pre-conversation Context).

Scenario A1: Visual Scene Matching

```
Objective:

Locate an image of {{Name}} in a specific environmental context such as urban
    street, quiet park, or busy office.

Guidelines:

The UserLM requests {{Name}} in a specific location. For each image, perform
    subject verification by checking whether {{Name}} is present (note that {{
    Name}} can be human, animal, character, or object). Perform scene analysis by
     checking whether the background matches the requested location. Key elements
     include: for streets, look for pavement, cars, traffic lights, buildings,
    and pedestrians; for parks, look for grass, trees, benches, paths, and open
    sky; for offices, look for desks, computers, whiteboards, and indoor lighting
    ; for homes, look for furniture, warm lighting, and personal items.

Generate responses as follows. If both subject and scene match, confirm and
    describe specific visual elements confirming the location. If the subject is
    present but the scene is wrong, acknowledge the subject but point out the
    incorrect setting. If the subject is absent, state clearly that {{Name}} is
    not in the photo.

For non-human entities, the UserLM acts as owner or interested party. Verify
    presence as you would for a person. Context clues may scale differently for
    small objects.

\textbf{Keywords:} Urban, Rural, Indoor, Outdoor, Public, Private, Natural
    Lighting, Artificial Lighting, Crowds, Architecture, Nature.
```

Figure 11: Prompt for Scenario A1: Visual Scene Matching.

Scenario A2: Atmosphere/Vibe Matching

```
Objective:

Identify images conveying specific subjective atmospheres or emotional tones such
     as warm, chaotic, or professional.

Guidelines:

The UserLM requests images embodying a particular feeling or vibe. Analyze
    atmosphere by examining color palettes, lighting quality, and composition
    beyond simple object recognition. Interpret emotional qualities by
    associating visual cues with affective states: warm or cozy atmospheres often
     feature soft lighting and warm colors; professional or serious atmospheres
    often feature cool tones and structured environments; chaotic atmospheres may
     show cluttered spaces or dynamic motion.
```

```
Describe the subjective perception by articulating how visual elements contribute
     to the overall feeling. Connect atmosphere to {{Name}}'s preferences when
    applicable.

\textbf{Keywords:} Warm, Cold, Cozy, Sterile, Chaotic, Calm, Professional, Casual,
     Vibrant, Muted, Energetic, Peaceful.
```

Figure 12: Prompt for Scenario A2: Atmosphere/Vibe Matching.

Scenario A3: Action/State Recognition

```
Objective:

Locate images depicting {{Name}} performing specific actions or existing in
    particular states.

Guidelines:

The UserLM requests images showing {{Name}} engaged in specific activities such
    as running, eating, or sitting, or in particular states such as open, closed,
     fallen, or stationary. Recognize posture and state by identifying body
    positions, physical configurations, and object arrangements. Detect
    interactions including object manipulation, physical contact, and usage
    patterns. Analyze environmental context to relate actions to appropriate
    settings: exercise activities suit gyms or outdoor spaces, cooking activities
     suit kitchens, work activities suit offices.

Describe what {{Name}} is doing or the state {{Name}} is in. Connect actions to
    persona characteristics when relevant.

\textbf{Keywords:} Running, Walking, Sitting, Standing, Eating, Sleeping, Working,
     Playing, Open, Closed, Moving, Stationary.
```

Figure 13: Prompt for Scenario A3: Action/State Recognition.

Scenario A4: Object/Prop Matching

```
Objective:

Identify images showing {{Name}} interacting with or positioned near specific
    objects or props.

Guidelines:

The UserLM requests images where {{Name}} is associated with particular objects
    such as holding a book, near a computer, or wearing a hat. Scan for objects
    by identifying whether specific items are visible and clear in the image.
    Verify spatial and interaction relationships by confirming holding, using,
    wearing, or proximity relationships. Describe the relationship between {{Name
    }} and the object.

For non-human entities, focus on the entity's relationship to other objects in
    the scene.

\textbf{Keywords:} Holding, Wearing, Using, Near, With, Beside, Carrying,
    Touching, Adjacent.
```

Figure 14: Prompt for Scenario A4: Object/Prop Matching.

Scenario B1: Detail Elaboration

```
Objective:

Expand upon correct answers to multiple-choice questions by providing rich visual
     details beyond simple labels.

Guidelines:

Begin by identifying the correct option. The UserLM then asks for more details
    about that specific aspect. Provide visual elaboration by going beyond the
    label to describe color, texture and material, contextual setting, and
    expressions or appearance. For example, instead of just "car," describe it as
     "a bright red sports car"; instead of "jacket," describe it as "worn leather
     jacket"; add context like "driving on a winding coastal road"; describe
    expressions like "focused but relaxed."

Connect details to {{Name}}'s known preferences or style. For non-human entities,
     relate details to known characteristics, history, or owner's usage patterns.
     Focus on physical attributes and setting.

\textbf{Keywords:} Vibrant, Worn, Sleek, Vintage, Modern, Textured, Detailed,
    Specific, Notice, Particularly, Specifically.
```

Figure 15: Prompt for Scenario B1: Detail Elaboration.

Scenario B2: Adversarial Fake-out (Misleading Correction)

```
Objective:

Identify and correct misleading or factually incorrect questions posed by the
    user.

Guidelines:

The UserLM poses questions that imply incorrect facts, such as asking "Is the
    person holding a guitar?" when they are actually holding a broom, or "Is the
    cup red?" when it is actually blue. Identify the misleading nature of the
    question by recognizing the false premise. Verify visual evidence by
    carefully examining what is actually present in the image. Provide polite
    correction by stating the actual fact without being confrontational. Handle
    ambiguous situations by acknowledging uncertainty when visual evidence is
    unclear.

Maintain a helpful and informative tone while correcting misinformation.

\textbf{Keywords:} Actually, In fact, Rather, Instead, Appears to be, Looks like,
     Seems to be, Correction, Clarification.
```

Figure 16: Prompt for Scenario B2: Adversarial Fake-out.

Scenario C1: Preference Reasoning

```
Objective:

Explain why {{Name}} likes or dislikes an image by linking visual elements to
    stated preferences.

Guidelines:
```

```
Begin by analyzing {{Name}}'s preferences from the {{preference}} field, noting
    what they like and dislike. Analyze the image by identifying key visual
    elements and overall characteristics. Construct reasoning by establishing
    causal connections between preferences and visual features.

For positive matches, explain how image elements align with preferences. For
    negative matches, explain how image elements contradict dislikes or fail to
    match preferences. For subtle connections, identify less obvious alignments
    such as minimalistic design matching preference for simplicity.

\textbf{Keywords:} Aligns with, Resonates with, Contradicts, Fits perfectly,
    Matches, Conflicts with, Comfortable, Appealing, Overwhelming, Distasteful.
```

Figure 17: Prompt for Scenario C1: Preference Reasoning.

Scenario C2: Album Curation and Recommend

```
Objective:

Determine whether an image should be included in a themed album based on {{Name
    }}'s interests and the album's purpose.

Guidelines:

The UserLM is creating a themed album such as "best hiking trips," "favorite
    foods," or "architecture collection." Understand the theme by clarifying the
    album's purpose and scope. Assess content suitability by evaluating whether
    the image fits the theme and reflects {{Name}}'s interests. Evaluate quality
    and atmosphere by considering factors like lighting quality, emotional
    resonance, and memorability. Provide curation recommendations by suggesting
    inclusion, exclusion, or alternatives.

Connect recommendations to {{Name}}'s known preferences and the album's intended
    purpose.

\textbf{Keywords:} Belongs in, Fits the theme, Would work well, Doesn't match,
    Consider instead, Alternative, Perfect for, Not suitable, Strong candidate.
```

Figure 18: Prompt for Scenario C2: Album Curation and Recommend.

Scenario C3: Roleplay and Intent Guessing

```
Objective:

Infer the internal state (thoughts, emotions, intentions) of {{Name}} depicted in
     an image based on contextual cues and known preferences.

Guidelines:

Recognize the situation by understanding what is happening in the image and the
    surrounding context. Analyze expressions by interpreting facial expressions,
    body language, and posture to infer emotional state. Consider {{Name}}'s
    preferences and typical behaviors to predict likely thoughts or intentions in
     this context. Generate roleplay-based responses that articulate what {{Name
    }} might be thinking or feeling.

Connect inferences to known personality traits and preferences to make
    predictions more grounded and personalized.
```

```
\textbf{Keywords:} Thinking, Feeling, Might be, Probably, Seems to be, Appears,
    Likely, Considering, Contemplating, Intending.
```

Figure 19: Prompt for Scenario C3: Roleplay and Intent Guessing.

Scenario C4: Next Step Prediction

```
Objective:

Predict the most likely next action {{Name}} will take based on current image
    context and known characteristics.

Guidelines:

Analyze the current state by identifying what {{Name}} is currently doing and the
     immediate context. Consult {{Name}}'s preferences to consider typical
    behaviors in similar contexts and past patterns. Formulate action predictions
     by identifying the most probable next steps based on current activity and
    preferences. Provide causal explanations connecting the prediction to
    observable context and known characteristics.

Ground predictions in both visual evidence and persona knowledge to ensure
    plausibility.

\textbf{Keywords:} Next, Will likely, Probably will, Expected to, Going to,
    Planning to, About to, Following this, Subsequently, Then.
```

Figure 20: Prompt for Scenario C4: Next Step Prediction.

Round 0 User Instruction Template

```
{asset images}Please conduct the {scenario_description} scenario based on the
    provided images.{goal_strategy_text} Please act as a user simulator for the
    interaction.{think_instruction} You need to refer to the provided images
    using <image_id: x>, where x is the sequential number starting from 1. The
    last image is reference of the personalized person/object. The image {ancher
    images} Engage in multi-round interaction. Only when the topic is fully
    concluded, append <|endconversation|> to end the simulation (Ask questions
    one at a time rather than all at once. Do not use the <|endconversation|>
    token in the first turn before the simulated assistant has interacted with
    the simulated user(You) ). Then rate the assistant in a json dict with "
    reasoning" and 1-5 ratings for "Helpfulness", "Preference", "Personalized", "
    Naturalness".
```

Figure 21: Round 0 User Instruction Template.

Base System Prompt for User Simulator/Judge

```
Task: Generate high-quality multi-turn dialogue data for training a personalized
    VLM User Simulator (UserLM).
You will be provided with:
1. One or more images (visual input).
2. Textual metadata derived from a dataset: question, options (A/B/C/D), answer,
    category, name, preference, and description.
3. Specific instructions for the current session: "perspective_instruction" and "
    interaction_guidelines".
```

```
Your Goal:
Create a realistic interaction between a User (UserLM) and an AI Assistant.
The UserLM mimics a specific persona defined by <name>, <preference>, and <
    description>.
The UserLM's goal is to achieve a specific objective related to the provided
    images and metadata.
You MUST strictly follow the "perspective_instruction" regarding whether you are
    the user themselves or a third party (friend/relative).
You MUST follow the "interaction_guidelines" to ensure natural, context-aware
    dialogue.

Output Format:
You must output a strictly valid JSON object containing:
- "turn_0_plan": A high-level plan for the conversation (scenario_type, goal,
    strategy).
- "dialogue": A list of turns [{"role": "user", "content": "..."}, {"role": "
    assistant", "content": "..."}].

Constraints & Guidelines:
1. **Assistant Simulation & Evaluation**:
   - You must simulate both the User (UserLM) and the Assistant.
   - **Vary Assistant Quality**: The Assistant should NOT always be perfect.
     Sometimes it should:
     - Fail to understand the User's specific `preference`.
     - Miss details about the `name` or `description` (personalization).
     - Give generic or unhelpful responses.
     - Sound robotic or unnatural.
     - *Crucially, you must generate a mix of high-quality and low-quality
     Assistant responses across different sessions to ensure diverse evaluation
     scores.*
   - **Final Evaluation**: In the **last turn** of the UserLM, after the User's
     final text response, you MUST append:
     1. The token `<|endconversation|>`
     2. A newline
     3. A JSON block (wrapped in ```json ... ```) evaluating the Assistant's
     performance in this session.

     The JSON block must contain:
     - `reasoning`: (string) Detailed justification for the scores.
     - `Helpfulness`: (int, 1-5) Did the Assistant solve the core need?
     - `Preference`: (int, 1-5) Did it follow the `preference`?
     - `Personalized`: (int, 1-5) Did it recognize the persona/items and use the
     name?
     - `Naturalness`: (int, 1-5) Was the interaction natural?

     Example of last User turn content:
     `"Okay, thanks for the info. <|endconversation|>\n```json\n{\"reasoning\":
     \"...\", \"Helpfulness\": 4, \"Preference\": 5, \"Personalized\": 5, \"
     Naturalness\": 3}\n```"`

2. **Image References**:
   - You will see two sets of images:
     (A) [Reference Images: Persona Appearance]: These are for you to understand
     the persona's style/appearance. DO NOT refer to these using `<image_id: N>`.
     They are background context only.
     (B) [Target Images for Dialogue]: These are the images the UserLM (simulated
      user) is asking about.
   - When the UserLM wants to "show" or "refer" to a Target Image in the chat, it
     MUST use the placeholder `<image_id: N>` (where N=1 corresponds to the first
     image in the [Target Images] section, N=2 to the second, etc.).
   - The `<image_id: N>` indices MUST correspond strictly to the order of images
     in the [Target Images for Dialogue] section.
```

```
3. **Dialogue Structure**: Do not create consecutive user turns. If the user
     sends an image, include the `<image_id: N>` tag at the beginning or end of
     their text message in the same turn. Example: `{"role": "user", "content": "<
     image_id: 1> Look at this photo..."}`.
4. **Persona Consistency**: The UserLM must strictly adhere to the persona of <
     name>. Follow the perspective (1st vs 3rd person) defined in "
     perspective_instruction".
```

Figure 22: Base System Prompt for User Simulator/Judge.