Amour: A Latent Dynamical System for Controllable and Stateful Social Interaction in LLMs

Anonymous Author(s)
Anonymous Institution
anonymous@example.com

Abstract

Social interaction is emerging as a core capability for large language models (LLMs), supporting applications from conversational agents to social simulation. Despite strong surface fluency, existing LLMs often fail to maintain persona-consistent behavior and stable social dynamics over extended interactions, leading to emotion drift, unstable trajectories, and eventual persona collapse. A central challenge is that social state is left implicit: current models are not required to represent, update, or expose long-horizon social dynamics in a structured manner. We propose Amour, a latent dynamical system framework that leverages the autoregressive interface of LLMs to model social interaction through an explicit, interpretable state projection. By projecting high-dimensional latent social awareness into a compact, low-dimensional state space that is jointly supervised with response generation, Amour enables the model to induce stable internal social dynamics implicitly, while keeping the state interface explicit, auditable, and controllable. Building on this formulation, we develop an end-to-end pipeline for scenario expansion, large-scale dialogue synthesis, and deterministic state annotation, and curate Amour-10k, a state-aware corpus for social interaction modeling. We train Amour-8B under executor-aligned supervision and evaluate it on 528 real-world social interaction scenarios. Experiments show that Amour-8B substantially improves state fidelity and long-horizon temporal coherence over strong proprietary and open-source baselines, while maintaining competitive response quality. Our artifact is available at https://anonymous.4open.science/r/Amour-OpenScience-07C5/.

CCS Concepts

• **Computing methodologies** → **Discourse, dialogue and pragmatics**; *Natural language generation*; • **Human-centered computing** → *HCI theory, concepts and models.*

Keywords

Social Interaction in LLMs, Emotion Dynamics, Controllable Generation, State Tracking, LLM Agents, Dataset, Evaluation

1 Introduction

Social interaction has become an increasingly important capability for large language models (LLMs), supporting applications such as conversational agents, social simulators, interactive companions, tutors, and customer-support systems [6, 11, 33]. In these settings, successful interaction requires more than locally fluent responses. An LLM-based agent should maintain a coherent social stance across turns, react consistently to recurring social cues, and adapt its relational attitude as the interaction evolves.

Despite strong linguistic competence, existing LLMs often struggle to maintain such stable social behavior in extended interactions. A response may appear plausible in isolation, while the model's emotional tone, relational attitude, or persona-consistent behavior shifts abruptly across turns. This can lead to emotion drift, inconsistent reactions to similar stimuli, failure to respect previously established boundaries, and eventual persona collapse in socially sensitive scenarios. These failures suggest that response-level fluency alone is insufficient for reliable social interaction.

A key limitation lies in the implicit treatment of social dynamics. Most LLM-based dialogue systems directly map dialogue history to the next response, leaving evolving social states entangled with surface text. Even when longer context windows or persona descriptions are provided, the model does not expose an explicit state trajectory that can be supervised, audited, or intervened on. As a result, it is difficult to determine why an agent's stance changes over time, whether the change is consistent with the interaction history, or how to steer the agent without rewriting the dialogue context.

This paper introduces Amour, an executable social state modeling framework for controllable LLM-based social interaction. Instead of treating social behavior as a direct history-to-response mapping, Amour introduces an explicit state interface that mediates between dialogue context and response generation. At each interaction turn, the model induces a structured transition plan, a deterministic executor updates the social state, and the response is generated conditioned on the resulting state. This design separates state evolution from surface realization: the intermediate state is not merely a free-form reasoning trace, but an executable and persistent interface that supports supervision, auditing, and counterfactual intervention.

To support training and evaluation, we build an end-to-end pipeline for scenario expansion, dialogue synthesis, and deterministic state annotation. Using this pipeline, we curate Amour-10k, a state-aware corpus for social interaction modeling, and train Amour-8B under executor-aligned supervision. We evaluate Amour-8B on Amour-Real-Eval, a benchmark of 528 real-world social interaction scenarios. Experiments show that Amour-8B improves executable state fidelity and social consistency over parameter-matched baselines while maintaining competitive response quality. Further analyses with controlled stimulus probes and counterfactual state interventions show that the explicit state interface can systematically steer model behavior under fixed dialogue context.

Contributions. This work makes the following contributions:

- We propose Amour, an executable social state modeling framework that introduces an explicit, interpretable state interface for controllable LLM-based social interaction.
- We construct Amour-10k, a state-aware social interaction corpus with scenario expansion, dialogue synthesis, and deterministic executor-based state annotation.

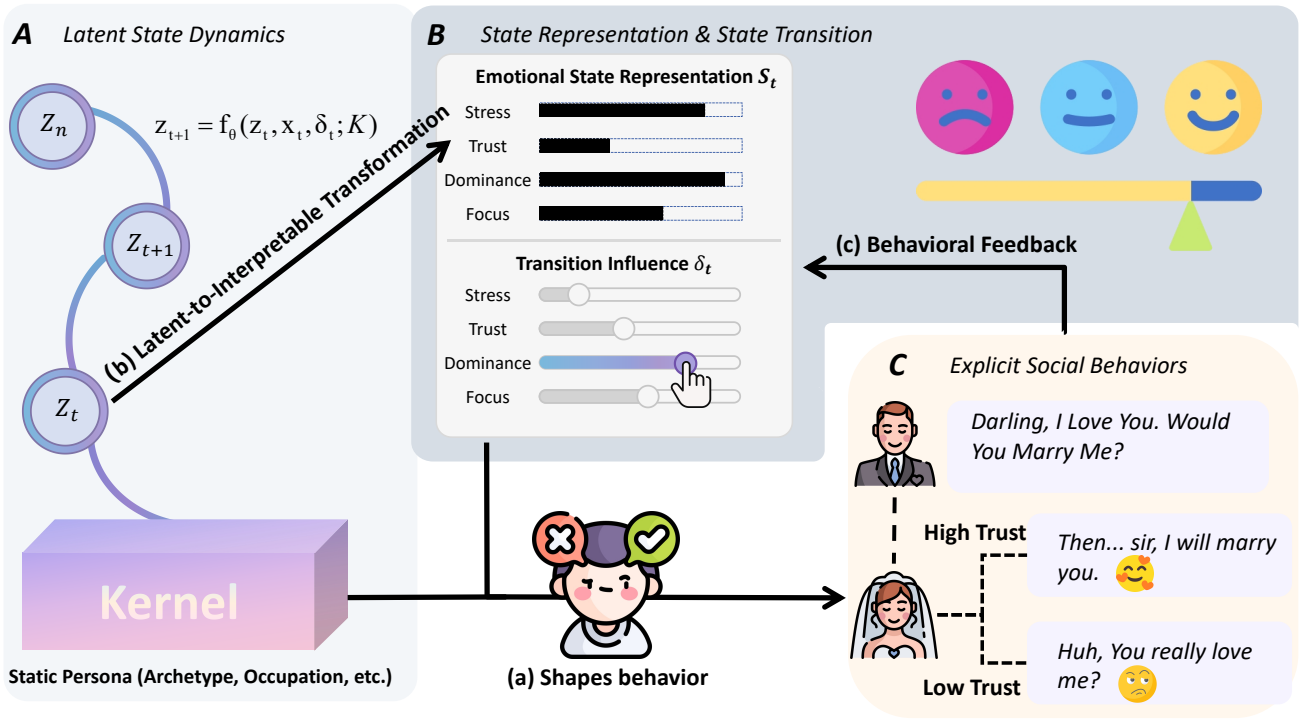

Figure 1: Closed-loop view of executable social state modeling. Area A illustrates the conceptual latent social dynamics $z_t \rightarrow z_{t+1}$, which are not directly parameterized in Amour. Area B exposes these dynamics through an explicit, interpretable state interface S_t, which is updated by a deterministic executor and used to condition response generation. Area C shows that editing S_t can steer subsequent behavior under the same dialogue context.

- We train Amour-8B with executor-aligned supervision and evaluate it on real-world social interaction scenarios using both response-level and state-aware metrics.
- We validate the controllability of the explicit state interface through ablations, stimulus-sensitive state tracking, and counterfactual state interventions.

2 Related Work

Persona-based and social dialogue agents. LLMs have enabled persona-based dialogue systems and social agents [1, 7, 10, 13, 25, 30, 42]. Existing agents imitate characters and speaking styles through persona descriptions, character data, psychological traits, and consistency-oriented evaluation [21, 34, 35, 37, 39, 41, 44]. However, they mostly rely on static profiles or dialogue history [44, 50], without explicitly modeling how social states evolve. Amour instead introduces an executable state interface that is updated across turns and conditions generation.

Memory, long-context interaction, and dialogue state. Long-context prompting and memory-augmented agents retain past information through extended contexts, retrieval, or summaries [20, 26, 28]. While useful for preserving factual history, these methods typically represent memory as natural language text rather than an executable state trajectory. Task-oriented dialogue systems track explicit states for slots, goals, and constraints [18, 46], but such states mainly support task completion rather than open-ended social dynamics. Amour reorients state tracking toward evolving social variables and makes the update process executable.

Reasoning traces and state supervision. Chain-of-Thought and its variants use intermediate reasoning traces to improve generation [3, 40, 45, 49], including affective and empathetic dialogue settings [19, 22, 38]. Unlike free-form reasoning traces, Amour gives the state update plan executable semantics: it is parsed by a deterministic executor, converted into an explicit state, and used to condition the response. Our data construction is also related to synthetic instruction tuning and distillation [5, 15, 16, 43, 47], but decouples dialogue generation from state annotation through executor-derived state targets.

3 Approach

This section presents Amour, an executable social state modeling framework for controllable LLM-based social interaction. The key idea is to separate *state evolution* from *surface realization*: the model predicts how the social state should change, a deterministic executor realizes this change as an explicit state, and the response is generated conditioned on the updated state. This design provides an interpretable interface for supervision, auditing, and intervention while remaining compatible with autoregressive language modeling.

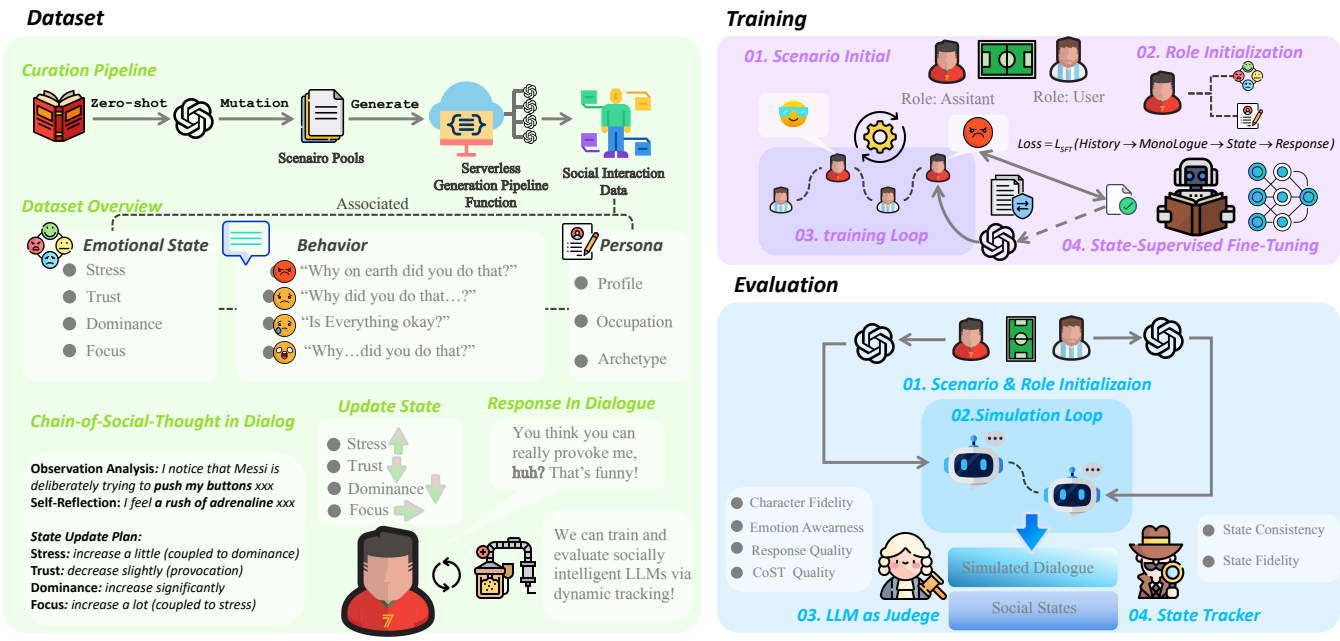

Figure 2: Overview of AMOUR's dataset, training, and evaluation. ❶ Dataset. Scenario expansion and dialogue synthesis yield structured trajectories with persona descriptions, state update plans, explicit state traces, and responses. ❷ Training. We fine-tune the model to autoregressively generate PLAN → STATE → RESPONSE, with explicit supervision on executor-derived states. ❸ Evaluation. Model outputs are evaluated with both response-level metrics and executor-aligned state metrics for state fidelity and consistency.

3.1 Task Formulation

We consider a multi-turn social interaction between an LLM-based agent and a dialogue partner. At turn t, the agent observes the dialogue history $H_{<t}$, the current partner utterance x_t, optional situational cues O_t, and a static persona or role specification $\mathcal{K}$. The goal is to generate a response R_t that is locally appropriate while remaining consistent with the evolving social dynamics of the interaction.

A standard dialogue model directly samples the interaction context to the next response:

$$R_t \sim p_\theta(\cdot \mid H_{<t}, x_t, O_t, \mathcal{K}). \tag{1}$$

This formulation can produce fluent and contextually relevant responses, but it does not expose how the agent's social state changes over time. Consequently, shifts in affect, stance, or relational attitude remain entangled with dialogue history and are difficult to supervise, audit, or control.

Conceptual latent dynamics. As illustrated in Figure 1, we view social interaction as being governed by an underlying latent social-affective state z_t. This latent state summarizes rich and partially observable social factors accumulated from previous turns, such as emotional pressure, trust, dominance, and attention. The evolution of this latent state can be abstractly written as

$$z_{t+1} = f(z_t, x_t, \delta_t; \mathcal{K}), \tag{2}$$

where x_t denotes the current interaction stimulus and δ_t denotes an explicit transition influence. This formulation is conceptual: AMOUR does not directly parameterize, infer, or supervise z_t. Instead, AMOUR exposes a compact and interpretable state interface S_t as a controllable readout of the latent dynamics:

$$S_t = g(z_t). \tag{3}$$

The framework operates only on this explicit interface. The state S_t is used as an operational variable that is updated across turns and conditions subsequent responses:

$$R_t \sim p_\theta(\cdot \mid H_{<t}, x_t, O_t, \mathcal{K}, S_t). \tag{4}$$

Thus, z_t provides the conceptual motivation for the closed-loop view in Figure 1, while all supervision, execution, and intervention in AMOUR are performed through the explicit state S_t.

3.2 Executable Social State Modeling

At each turn, AMOUR follows an update-before-speak process: it first generates `<State Update Plan>`, then derives `<Current State>`, and finally generates `<Response>`.

The model first predicts a structured update plan from the current context and the previous state. A deterministic executor then converts this plan into the next explicit social state. Finally, the model generates the response conditioned on the updated state.

Formally, the update process is factorized as:

$$\begin{aligned} \text{PLAN}_t &= h_\theta(H_{<t}, x_t, O_t, \mathcal{K}, S_{t-1}), \\ S_t &= f_{\text{exec}}(S_{t-1}, \text{PLAN}_t), \end{aligned} \tag{5}$$

where h_θ denotes the LLM-generated planning component and f_{exec} denotes a fixed deterministic executor.

We emphasize that the executor is not an LLM module. It is an external state-update function that maps the previous state and the model-generated update plan to the next explicit state. Thus, the update plan is produced by the model, whereas the state transition itself is computed by a deterministic mechanism. This distinction is central to Amour. Unlike standard free-form chain-of-thought traces, the intermediate plan has executable semantics: it is parsed and executed to produce a persistent state trajectory that can be supervised, audited, and intervened on.

3.3 State Executor

The executor maps a structured update plan to a bounded state transition. Each plan specifies the intended direction and relative magnitude of change for the social state. The executor then applies three operations: converting the plan into transition influence, smoothing the transition with temporal inertia, and clipping the resulting state to the valid range.

Let $\hat{S}_t$ denote the state obtained after applying the plan-derived update to the previous state. To prevent abrupt oscillations across turns, the executor applies exponential moving average smoothing:

$$\widetilde{S}_t = (1-\eta)S_{t-1} + \eta\hat{S}_t, \tag{6}$$

where η controls the update rate. The smoothed state is then clipped to the valid range:

$$S_t = \mathrm{clip}(\widetilde{S}_t, 0, 1). \tag{7}$$

This design reflects the intuition that social states usually evolve gradually rather than resetting at each turn.

The executor can also incorporate lightweight cross-dimension coupling to encode simple dependencies among state variables. We keep these couplings deterministic and interpretable, and provide the full update schema and parameter settings in Algorithm 2.

For autoregressive supervision, continuous executor states are serialized into bounded range tokens in the `<Current State>` field. This semi-compressed representation reduces sensitivity to small numeric variations while preserving a continuous internal state inside the executor. The complete serialization schema is provided in Algorithm 3.

3.4 Executor-Aligned Training

We train Amour using the same structured order used at inference time. Given the turn-level context $C_t = (H_{<t}, x_t, O_t, \mathcal{K}, S_{t-1})$, the model is supervised to generate $\textsc{Plan}_t \rightarrow S_t \rightarrow R_t$. The training objective is the standard autoregressive likelihood over this structured target sequence:

$$\mathcal{L}_{\mathrm{SFT}} = -\sum_{t=1}^{T} \log p_\theta(\textsc{Plan}_t, S_t, R_t \mid C_t). \tag{8}$$

The state target S_t is produced by the deterministic executor rather than assigned as an unconstrained free-form label. This aligns training with the inference-time update procedure: the model learns to produce plans and states that are compatible with the executable state interface. It also reduces circularity in supervision, since the numeric state trace is computed by a fixed external mechanism rather than assigned solely by the same model that generates the dialogue.

3.5 Inference and Intervention

At inference time, Amour follows the same update-before-speak procedure. Given the previous state and current interaction context, the model predicts a state update plan. The executor applies the plan to obtain the updated state, and the response is generated conditioned on this state:

$$\textsc{Plan}_t \rightarrow S_t \rightarrow R_t.$$

This explicit interface enables direct intervention. To test controllability, we can hold the dialogue context fixed while modifying the state or constraining the executor output. If the resulting response changes systematically with the intervention, the state functions as a causal control handle rather than a post-hoc explanation. In our experiments, we use controlled stimulus probes and counterfactual state interventions to examine whether the learned state interface governs model behavior in predictable ways.

4 Data and Training

Training Amour requires supervision over both response generation and executable social-state updates. We therefore build an end-to-end pipeline that expands social scenarios, synthesizes multi-turn interaction trajectories, and derives state supervision with a deterministic executor decoupled from the dialogue-generation LLM. This pipeline yields Amour-10k, a state-aware corpus for social interaction modeling.

4.1 Data Generation Pipeline

Curation. We start from seed scenarios covering common social dynamics, such as conflict, support, persuasion, and reconciliation, and expand them with zero-shot [4] LLM-based mutation as a diversity operator (see 1). For each scenario, we instantiate role descriptions and a static profile $\mathcal{A}$, then synthesize multi-turn dialogues at scale via serverless parallelization (see § A.3). Each trajectory contains turn-level fields aligned with the Amour interface, including structured transition traces, state update plans, executor-derived states, and responses. We retain raw generations and metadata such as scenario ID, seed, and turn index for reproducibility and post-hoc filtering.

Decoupled state annotation. The LLM generates the linguistic trajectory, while numeric state targets are computed independently by the deterministic executor described in § 3.3. Given the previous state and the executable `<State Update Plan>`, the executor derives the target state S_t. This decoupling treats state as an operational variable rather than a subjective emotion label, providing consistent and auditable supervision while reducing circular self-supervision, label leakage, and co-adaptation [14, 17].

State serialization. For autoregressive training, executor-derived states are serialized into bounded range tokens in the `<Current State>` field. The executor maintains continuous internal states, while the model is supervised on a semi-compressed textual representation. This reduces sensitivity to small numeric variations [2, 36] while preserving continuous inertia in the executor update. We defer action spaces, thresholds, archetype initialization, and schema details to Algorithm 3.

Model	State Awareness			Plan Acc.	Textual Similarity			Scale
	Eq. Acc. ↑	Range Acc. ↑	Fidelity Avg. ↑		BLEU ↑	ROUGE ↑	SBERT ↑	
AMOUR-8B	**0.2936**	**0.7992**	**0.5464**	**0.6761**	0.8140	0.1213	0.2372	< 10B
Llama-3.1-8B-Instruct	0.0086	0.6789	0.3438	0.4538	0.6094	0.1031	0.2307	< 10B
CharacterGLM-8B	0.0035	0.6427	0.3231	0.3974	0.0235	0.0703	0.1670	< 10B
Gemini-2.5-Pro	0.0304	0.6743	0.3524	0.6419	2.0073	0.1278	0.2597	> 100B
Qwen3-max	0.0152	0.6496	0.3324	0.6420	0.9682	0.1158	0.2589	> 100B
DeepSeek-R1	0.0230	0.6546	0.3388	0.6036	1.5743	0.1223	0.2540	> 100B

Table 1: State-awareness and textual-similarity evaluation on 528 real-world social interaction scenarios. Eq. Acc. measures exact match with the reference previous state, while Range Acc. measures whether the prediction falls within the corresponding discretized state interval. Fidelity Avg. averages Eq. Acc. and Range Acc. as an overall state-fidelity score. Plan Acc. evaluates whether the predicted update plan yields an executor-valid state transition. Scale indicates the approximate model-size category. AMOUR-8B achieves the best executor-aligned state fidelity and plan consistency among all evaluated models, while maintaining competitive textual similarity.

4.2 Training Objective and Procedure

We train AMOUR with a single autoregressive objective on the same structured serialization used at inference time. Let $C_t = (H_{<t}, x_t, O_t, \mathcal{A}, S_{t-1})$ denote the turn-level context. The model is supervised to generate a structured transition trace, the executor-derived current state, and the final response. The executable component of the transition trace is PLAN_t, corresponding to `<State Update Plan>`; auxiliary fields such as `<Observation Analysis>` and `<Self-Reflection>` provide context for plan generation but are not directly executed.

The target follows the update-before-speak order:

$$\text{PLAN}_t \rightarrow S_t \rightarrow R_t. \tag{9}$$

The standard autoregressive objective is:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log p_\theta(\text{PLAN}_t, S_t, R_t \mid C_t). \tag{10}$$

Executor-aligned state supervision. At each turn, the target state is computed by the fixed executor:

$$S_t = f_{\text{exec}}(S_{t-1}, \text{PLAN}_t). \tag{11}$$

The model is then supervised to generate the `<Current State>` tokens corresponding to S_t. This aligns training with inference, encourages compatibility with the executable state interface, and keeps state supervision exogenous to the dialogue-generation model.

5 Experiments

5.1 Experiment Settings

We evaluate whether executable social state modeling improves ❶ end-to-end social interaction quality, ❷ executor-aligned state fidelity, and ❸ controllability under state interventions. All experiments follow the data format and inference interface described in Appendix C.

Evaluation protocol. We conduct end-to-end evaluation on AMOUR-REAL-EVAL, which contains 528 real-world social interaction scenarios. Each scenario specifies a persona profile and a situation prompt. For executor-aligned evaluation, models are prompted to produce outputs in the AMOUR-compatible format, including a transition trace, an explicit state prediction, and a final response (see Table 13).

Model	Char. Fid.	Resp. Qual.	Emotion Aware.	CoST Qual.	Avg.
AMOUR-8B	83.69	78.59	66.61	61.72	72.65
Llama-3.1-8B-Instruct	45.13	41.37	38.06	36.70	40.32
CharacterGLM-8B	22.33	28.34	22.60	20.63	23.48
Gemini-2.5-Pro	93.30	84.56	73.96	71.18	80.75
Qwen3-max	96.00	89.15	79.56	78.14	**85.71**
DeepSeek-R1	95.50	87.35	76.74	73.33	83.23

Table 2: LLM-as-a-Judge evaluation of holistic social interaction quality. Scores (0–100) assess Character Fidelity, Response Quality, Emotion Awareness, and CoST Quality. Avg. is the arithmetic mean across the four dimensions.

Response-level evaluation is conducted on the final response only. For ablation and state-compression analyses, we construct a held-out test set by splitting AMOUR-10K into disjoint training and evaluation portions, and report results on the evaluation split only (see § A.2).

We adopt a mixed evaluation protocol that combines response-level rubric scoring with executor-aligned state metrics. End-to-end quality is assessed using an LLM-as-judge scheme (see § D.1), covering Character Fidelity and Response Quality, with additional dimensions such as Emotion Awareness and CoST (Chain-of-Social-Thought) Quality reported where applicable (see Table 1 and Table 2).

Executor-aligned state metrics are reported separately from response-level metrics. They measure compatibility with the AMOUR state interface rather than general dialogue quality. Specifically, state-awareness is evaluated via deterministic executor-based checks. We measure whether the predicted discrete state range matches the executor outcome (State Range Acc.) and whether the predicted update plan induces an executor-valid state transition (Plan Acc.). We also report lightweight automatic statistics, including BLEU [27], ROUGE [23], and SBERT semantic similarity [32], as complementary signals.

To assess whether AMOUR-8B's behavior is governed by the explicit state interface, we conduct stimulus-controlled probes (see Figure 3) and counterfactual state interventions under fixed context (see Table 5). To ensure the reliability of the judging protocol, we validate human–LLM alignment on AMOUR-REAL-EVAL (see § D.1),

Figure 3: Stimulus-sensitive state dynamics.

observing strong agreement with Cohen's [8] $\kappa = 0.74$ and Pearson [29] $r = 0.90$.

Models. We compare Amour-8B against open-source baselines, including Llama-3.1-8B-Instruct [25] and CharacterGLM-6B [34], as well as representative topline models, including Gemini-2.5-Pro [13], Qwen3-max [30], and DeepSeek-R1 [10].

5.2 How does Amour perform on real-world social interaction scenarios?

We evaluate Amour-8B and a range of open-source and proprietary LLMs on Amour-Real-Eval, with results summarized in Table 1 and Table 2.

Key findings. ❶ Amour-8B achieves strong performance across both response-level interaction quality metrics and executor-aligned state metrics, substantially outperforming open-source models of comparable scale. ❷ Strong general-purpose LLMs obtain high response-level rubric scores, but still exhibit a clear gap in executor-aligned state fidelity and plan consistency. ❸ This gap highlights an important dissociation: surface-level response quality does not necessarily imply faithful or executable state tracking.

Overall, Amour-8B maintains competitive interaction quality while achieving strong executor consistency. This reflects its training objective, which explicitly models the structured mapping from interaction evidence to update plan, state, and response. Rather than relying only on response-level pattern matching, Amour-8B learns to generate responses through an explicit state interface.

We further report win-rate results in Table 7. Amour-8B wins substantially more often than open-source baselines of comparable scale. Larger proprietary models remain strong on response-level quality metrics, while Amour-8B shows a clearer advantage on executor-aligned state metrics. These trends are consistent with the distinction between general response fluency and executable state fidelity.

5.3 How do Amour's design choices affect interaction quality and state stability?

We conduct controlled ablations to examine which components are necessary for effective state-aware social interaction. Table 4 and Table 5 compare the full Amour model with three variants that remove key mechanisms: without the explicit state channel (*w/o State*), without the structured update-planning interface (*w/o CoST*), and a response-only model that does not learn state transitions.

Key findings. ❶ The full Amour model achieves the strongest overall performance across judge-based rubric dimensions and automatic similarity metrics, indicating that the complete state-aware pipeline yields the most reliable end-to-end behavior. ❷ Ablating either the explicit state channel (*w/o State*) or the structured update-planning interface (*w/o CoST*) leads to consistent degradation, including lower semantic alignment and weaker overlap-based scores. These results suggest that both explicit state representations and structured update plans contribute to effective state utilization.

These ablations suggest that response-only optimization is insufficient to recover the full benefits of Amour-style state-aware training. Learning explicit state representations and structured update dynamics jointly with response generation improves semantic consistency, controllability, and compatibility with the executable state interface.

5.4 Does Amour-8B use the explicit state interface to control behavior?

To verify that the explicit state serves as a controllable interface for response generation, we test whether Amour-8B learns stimulus-sensitive state transitions and whether intervening on the state alone induces systematic behavioral changes under fixed dialogue context, rather than merely producing state as a post-hoc explanation.

Stimulus sensitivity. We first evaluate whether the learned state dynamics respond systematically to different categories of social stimuli. Figure 3 shows that state updates are modulated by stimulus type. Negative stimuli, such as insults, induce higher Stress together with decreases in Trust and Dominance. Positive stimuli, such as compliments, produce consistent increases in Trust and Dominance, while neutral casual chat yields only minor shifts.

Counterfactual state intervention. To assess whether the explicit state provides a controllable handle for response generation, we perform counterfactual interventions by reversing different state dimensions while holding the dialogue context fixed.

As shown in Table 5, reversing Trust alone produces a clear stance shift while preserving the main semantic content, indicating that Trust can steer decision-making under identical context. Table 14 qualitatively illustrates this effect: given the same dialogue context, Amour-8B generates different responses under different states. Specifically, under high Trust, the model appropriately accepts the proposal with direct affirmation *("Then... sir, I will marry you")*, while under low Trust, it shifts to cautious questioning *("You love me?")*, with both responses maintaining natural dialogue flow and character consistency.

These results indicate that the explicit state interface mediates between context interpretation and response generation. The model compresses social cues into semantically grounded state variables and uses them to govern response behavior, enabling targeted controllability while keeping generation anchored to the same dialogue context.

6 Conclusion

We present Amour, an executable social state modeling framework that separates state updates from response generation through structured transition plans and state-conditioned decoding. Using Amour-10k and executor-aligned supervision, Amour-8B improves state fidelity and plan consistency on 528 real-world scenarios while maintaining competitive response quality. Ablations and interventions further show that the explicit state interface is

Variant	State Awareness			Plan Acc.	Textual Similarity		
	Eq. Acc. ↑	Range Acc. ↑	Fidelity Avg. ↑		BLEU ↑	ROUGE ↑	SBERT ↑
w/o State	–	–	–	–	5.1497	0.1826	0.3847
w/o CoST	0.1166	0.8430	0.4798	–	4.8365	0.1824	0.3905
Response-only	–	–	–	–	4.0296	**0.1984**	0.3869
Full	**0.4810**	**0.8926**	**0.6868**	**0.6761**	**5.9646**	0.1899	**0.4072**

Table 3: Ablation on state awareness and textual similarity. Variants remove one component from the full model: *w/o State* (no explicit state), *w/o CoST* (no CoST), and *Response-only* (no structured intermediates). State-awareness and plan-accuracy metrics apply only when explicit state is present; – denotes not applicable. BLEU/ROUGE/SBERT report surface/semantic similarity of responses.

Variant	Char. Fid.	Resp. Qual.	Emotion Aware.	CoST Qual.	Avg.
w/o State	94.95	85.81	73.41	80.23	83.60
w/o CoST	94.81	85.90	74.07	–	–
Response-only	94.76	85.29	73.85	–	–
Full	**95.78**	**87.55**	**86.57**	**85.40**	**88.37**

Table 4: Ablation study on holistic social interaction quality. Scores (0–100) assess persona consistency (*Character Fidelity*), response quality, emotional awareness, and Chain-of-Social-Thought (CoST) coherence. Avg. reports the arithmetic mean across available judge dimensions. – indicates metrics that are not applicable for the corresponding variant.

Flip	Sent. Change ↑	SBERT ↑	MEC ↓	Sent. Sim. ↓
All dims	0.33	0.35	**0.05**	0.83
Dom	**0.43**	0.37	0.07	0.76
Focus	0.40	0.37	0.07	0.78
Stress	0.40	0.37	0.06	0.80
Trust	0.32	0.34	0.06	**0.84**

Table 5: Counterfactual state interventions. With dialogue context fixed, we flip all state dimensions or one dimension before generating the response. Sent. Change: absolute sentiment-score difference (higher indicates stronger affective shift); SBERT: sentence-embedding cosine similarity (higher indicates better semantic preservation); Sent. Sim.: sentiment-embedding cosine similarity (*lower indicates stronger affective change*); MEC: $(1-\text{Sent. Sim.})\cdot\text{SBERT}$, where lower is better.

both useful and controllable, offering a practical path toward more auditable LLM-based social interaction.

Impact Statement

This paper aims to advance machine learning, with a particular focus on modeling and evaluating state-aware social interaction in large language models. While this line of research may have broad societal implications, we do not believe it raises ethical or societal concerns beyond those commonly associated with advances in language modeling that warrant separate discussion here.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Ron Banner, Yury Nahshan, and Daniel Soudry. 2019. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in neural information processing systems* 32 (2019).

[3] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 16 (March 2024), 17682–17690. doi:10.1609/aaai.v38i16.29720

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[5] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '06)*. Association for Computing Machinery, New York, NY, USA, 535–541. doi:10.1145/1150402.1150464

[6] Character.AI. 2023. Character.AI: Conversational AI for Role-Playing and Social Interaction. https://character.ai/. Accessed: 2025-01.

[7] Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, Aili Chen, Nianqi Li, Lida Chen, Caiyu Hu, Siye Wu, Scott Ren, Ziquan Fu, and Yanghua Xiao. 2024. From Persona to Personalization: A Survey on Role-Playing Language Agents. doi:10.48550/arXiv.2404.18231 arXiv:2404.18231 [cs].

[8] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46.

[9] Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics (ACL)*. https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html

[10] DeepSeek AI. 2026. DeepSeek-R1. https://github.com/deepseek-ai/DeepSeek-R1. Accessed: 2026-01-25.

[11] Ditt AI. 2024. Ditt: AI Agents for Collaborative Work. https://ditt.ai/. Accessed: 2025-01.

[12] Google Cloud. 2026. Google Cloud Run. https://cloud.google.com/run. Accessed: 2026-01-25.

[13] Google DeepMind. 2026. Gemini 2.5. https://deepmind.google/technologies/gemini/. Accessed: 2026-01-25.

[14] Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems* 17 (2004).

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. doi:10.48550/arXiv.1503.02531 arXiv:1503.02531 [stat].

[16] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. doi:10.48550/arXiv.2305.02301 arXiv:2305.02301 [cs].

[17] Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 conference on empirical methods in natural language processing*. 1051–1068.

[18] Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5478–5483.

[19] Yoon Kyung Lee, Inju Lee, Minjung Shin, Seoyeon Bae, and Sowon Hahn. 2024. Chain of Empathy: Enhancing Empathetic Response of Large Language Models Based on Psychotherapy Models. *Korean Journal of Cognitive Science* 35, 1 (2024), 23–48. doi:10.19066/cogsci.2024.35.1.002 arXiv:2311.04915 [cs].

[20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*.

[21] Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, Linkang Zhan, Yaokai Jia, Pingyu Wu, and Haozhen Sun. 2023. ChatHaruhi: Reviving Anime Character in Reality via Large Language Model. doi:10.48550/arXiv.2308.09597 arXiv:2308.09597 [cs].

[22] Zaijing Li, Gongwei Chen, Rui Shao, Yuquan Xie, Dongmei Jiang, and Liqiang Nie. 2024. Enhancing Emotional Generation Capability of Large Language Models via Emotional Chain-of-Thought. doi:10.48550/arXiv.2401.06836 arXiv:2401.06836 [cs].

[23] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. 74–81.

[24] LLaMA-Factory Contributors. 2026. LLaMA-Factory. https://github.com/hiyouga/LLaMA-Factory. Accessed: 2026-01-25.

[25] Meta AI. 2026. Llama 3. https://ai.meta.com/blog/meta-llama-3/. Accessed: 2026-01-25.

[26] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560* (2023).

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. 311–318.

[28] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*. 1–22.

[29] Karl Pearson. 1895. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London* 58 (1895), 240–242.

[30] Qwen Team. 2026. Qwen3. https://qwenlm.github.io/blog/qwen3/. Accessed: 2026-01-25.

[31] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 3505–3506.

[32] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*. 3982–3992.

[33] Replika Inc. 2023. Replika: An AI Companion for Emotional Support. https://replika.com/. Accessed: 2025-01.

[34] Yihong Shao, Zhijie Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hongyin Bao, Weipeng Zhang, Tongzheng Xu, Xianfeng Luo, et al. 2023. CharacterGLM: Customizing Chinese Conversational AI Characters with Large Language Models. *arXiv preprint arXiv:2308.10099* (2023).

[35] Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-LLM: A Trainable Agent for Role-Playing. doi:10.48550/arXiv.2310.10158 arXiv:2310.10158 [cs].

[36] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[37] Quan Tu, Shilong Fan, Zihang Tian, Tianhao Shen, Shuo Shang, Xin Gao, and Rui Yan. 2024. CharacterEval: A Chinese Benchmark for Role-Playing Conversational Agent Evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 11836–11850. doi:10.18653/v1/2024.acl-long.638

[38] Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam-Fai Wong. 2023. Cue-CoT: Chain-of-thought Prompting for Responding to In-depth Dialogue Questions with LLMs. doi:10.48550/arXiv.2305.11792 arXiv:2305.11792 [cs].

[39] Xintao Wang, Heng Wang, Yifei Zhang, Xinfeng Yuan, Rui Xu, Jen-tse Huang, Siyu Yuan, Haoran Guo, Jiangjie Chen, Shuchang Zhou, Wei Wang, and Yanghua Xiao. 2026. CoSER: A Comprehensive Literary Dataset and Framework for Training and Evaluating LLM Role-Playing and Persona Simulation. doi:10.48550/arXiv.2502.09082 arXiv:2502.09082 [cs].

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

[41] Xintao Wang, Yunze Xiao, Jen-tse Huang, Siyu Yuan, Rui Xu, Haoran Guo, Quan Tu, Yaying Fei, Ziang Leng, Wei Wang, Jiangjie Chen, Cheng Li, and Yanghua Xiao. 2024. InCharacter: Evaluating Personality Fidelity in Role-Playing Agents through Psychological Interviews. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 1840–1873. doi:10.18653/v1/2024.acl-long.102

[42] Ye Wang, Jiaxing Chen, and Hongjiang Xiao. 2026. Role-Playing Agents Driven by Large Language Models: Current Status, Challenges, and Future Trends. doi:10.48550/arXiv.2601.10122 arXiv:2601.10122 [cs].

[43] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. doi:10.48550/arXiv.2212.10560 arXiv:2212.10560 [cs].

[44] Zekun Moore Wang, Zhongyuan Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Stephen W. Huang, Jie Fu, and Junran Peng. 2024. RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models. doi:10.48550/arXiv.2310.00746 arXiv:2310.00746 [cs].

[45] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[46] Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 808–819.

[47] Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A Survey on Knowledge Distillation of Large Language Models. doi:10.48550/arXiv.2402.13116 arXiv:2402.13116 [cs].

[48] yangwang825. 2026. DailyDialog++. https://huggingface.co/datasets/yangwang825/daily_dialog_plus_plus. Accessed: 2026-01-25.

[49] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in neural information processing systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 11809–11822. https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf

[50] Jinfeng Zhou, Zhuang Chen, Dazhen Wan, Bosi Wen, Yi Song, Jifan Yu, Yongkang Huang, Pei Ke, Guanqun Bi, Libiao Peng, et al. 2024. CharacterGLM: Customizing Social Characters with Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 1457–1476.

A Dataset

A.1 Statistics and Analysis

Our training dialogue dataset contains 10,569 conversation instances with 35,388 total dialogue turns. It covers 4,691 distinct role personas and spans 20 archetypes, providing broad diversity in both character profiles and interaction patterns for supervised training.

As shown in Figure 4 (a), topic distribution is abundant, with casual chat constitutes a substantial portion, the dataset also includes a broad spectrum of practical and emotionally grounded situations, encouraging generalization beyond small-talk. Figure 4 (b) demonstrates a deliberately imbalanced yet diverse archetype distribution, with analytically oriented, empathetic, and risk-averse personas being prominent, reflecting realistic social role frequencies rather than uniform sampling.

A.2 Data Splits

The dataset is divided into three disjoint subsets serving distinct experimental purposes. Firstly, we construct a real-world test set (AMOUR-REAL-EVAL) consisting of 528 dialogue instances, collected from human multi-turn conversations and movie-script-based interactions. This test set is used exclusively for evaluating end-to-end social interaction quality and generalization to previously unseen dialogue patterns (§ 5.2). Secondly, from the remaining synthetic dataset of approximately 10,000 instances, we reserve 5% of the data as a held-out validation split. This split is not used for training, but is employed solely for controlled analyses, including ablation studies (see § 5.3) and state compression experiments (see § B.1). Lastly, the remaining data is used for AMOUR-8B training.

All splits are strictly disjoint at the scenario level, ensuring that no dialogue context, persona profile, or state trajectory appears in more than one subset.

A.3 Implementation Details for Construction

AMOUR-10K Generation. The training dataset is constructed through a controlled synthetic generation pipeline grounded in real-world conversational scenarios. We first curate a pool of realistic dialogue situations (e.g., cooperative planning) and apply zero-shot mutation to expand this pool into a diverse set of scenarios (see 1).

For each scenario, the generation proceeds in multiple stages. In the first stage, a language model is prompted to generate a persona profile, including high-level personality traits and role descriptions. This persona profile is then mapped to an initial dialogue state using the archetype-based initialization scheme (see Table 15).

In the second stage, the language model is invoked sequentially over multiple turns to generate, at each turn, an `<Internal Monologue>` trace and the corresponding natural-language response. After each turn, we extract the `<State Update Plan>` from the internal monologue, together with the previous dialogue state, and apply a deterministic rule-based executor (see Algorithm 3) to compute the resulting `<Current State>`. This post-processing step ensures that all state transitions are externally grounded and consistent with the specified update rules.

Algorithm 1 Scenario Generation

Require: Seed scenarios $\mathcal{S}_0$, mutation budget B, mutator LLM $\mathcal{M}$
Ensure: Expanded scenario pool $\mathcal{S}$

```
S ← S_0
while |S| < B do
    Pick a seed s ∈ S                                   ▷ Coverage-aware Seed Scheduling
    Sample a mutation policy π                          ▷ Relationship/Setting/Event/Stakes
    s' ← MUTATE(s; π) via M under a fixed schema        ▷ Zero-shot Mutation
    Canonicalize s' and validate                        ▷ Schema + Safety + Dedup
    if s' is valid and non-duplicate then
        S ← S ∪ {s'}
    end if
end while
return S
```

All generation is performed via a serverless function-as-a-service pipeline deployed on cloud run [12], which enables scalable and parallelized data generation. We use Gemini-2.5-Pro [13] as the base language model for scenario expansion and dialogue generation. In total, we incur a cost of over **$3,000 USD** to generate the full dataset, resulting in approximately 10,000 high-quality dialogue instances with fully annotated state trajectories.

AMOUR-REAL-EVAL Generation. The test set is constructed independently from the training data. It is derived from a mixture of open-domain conversational sources, including multi-turn human dialogues [48] and movie-script-based interactions [9], which are adapted into the same structured format used during training. These dialogues are not used during training and are processed only to extract dialogue context and persona information, ensuring that evaluation reflects generalization to previously unseen interaction patterns.

B State Compression Experiments

B.1 State Compression Experiments

We study how different state representations affect controllability-related metrics and overall response quality, with a particular focus on the trade-off between expressive precision and stable control. For each variant, we report the best-performing epoch across training and compare three settings: *Full-float*, which represents each state variable as continuous-valued tokens; *Semi-Compressed*, which discretizes continuous states into bounded intervals while preserving temporal inertia; and *Tag-tag*, which encodes states using coarse categorical tags without numeric magnitude.

As shown in Table 6, *Semi-Compressed* states achieve the strongest overall balance. Compared to Full-float states, semi-compression improves state-level controllability signals, including higher State Sensitivity and State–Response Alignment, while reducing Functional Error and Conditional Entropy.

B.2 Semi-compressed State Representation

We adopt a semi-compressed state representation to balance numeric stability and semantic controllability. Specifically, the previous dialogue state is provided to the model as continuous values in the conditioning context, while the current state is supervised using discretized symbolic tags (see 3).

At training time, the model is conditioned on the continuous previous state S_{t-1}, but is supervised to predict the discretized tags corresponding to the current state S_t. At inference time, predicted

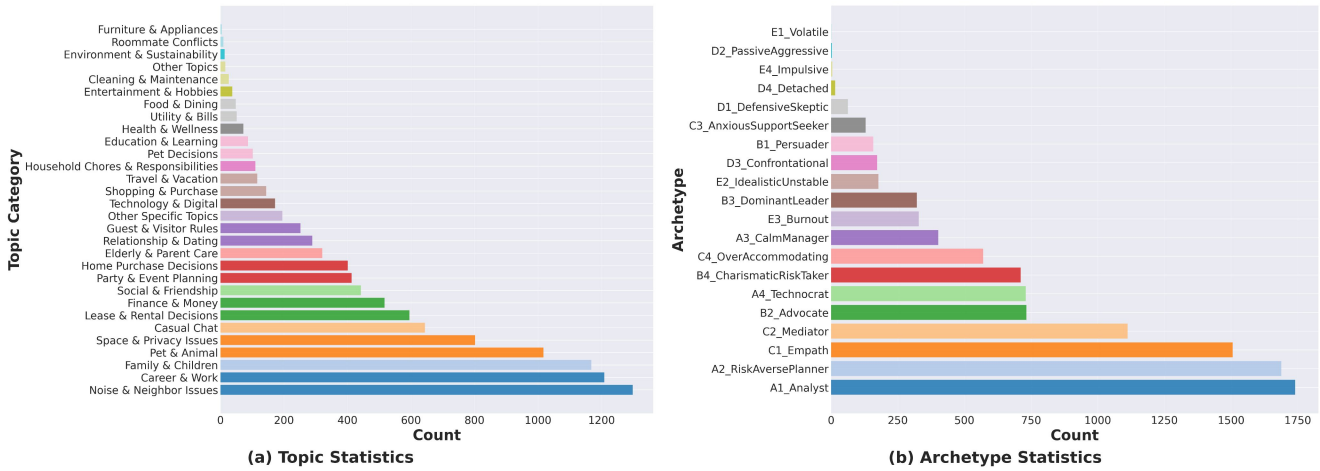

Figure 4: Dataset statistics

tags are deterministically mapped back to numeric intervals, yielding a bounded continuous state that is fed into the next turn.

This design preserves continuous state dynamics across turns, while introducing an explicit symbolic bottleneck that mitigates sensitivity to small numeric perturbations and reduces error accumulation over long interaction horizons.

C Training

We train the model with a single autoregressive objective over a structured serialization of the form (see Table 10). At each turn, the model is conditioned on the full dialogue history, the static persona profile, and the previous dialogue state, and is supervised to sequentially generate the internal monologue trace, the current state representation, and the final response in a single forward pass.

All experiments are implemented using the LLaMA-Factory [24] training framework, leveraging its supervised fine-tuning pipeline with data formatted in the ShareGPT schema. Each training instance is represented as a multi-turn conversation, where the structured outputs (`<Internal Monologue>`, `<Current State>`, and `<Response>`) are serialized within a single assistant turn. This design enables seamless integration with standard causal language modeling without requiring architectural modifications or custom loss functions.

To support efficient large-scale training, we employ DeepSpeed [31] for distributed optimization and memory management. DeepSpeed is used transparently within the LLaMA-Factory pipeline, allowing us to scale training while preserving the same autoregressive supervision objective.

D Experiment Settings

D.1 LLM-as-Judge.

Decoding. Unless otherwise stated, we use nucleus sampling with top-$p = 0.9$ and temperature 0.7 for Amour-8B. **Judging.** We use Gemini-2.5-Pro as an LLM judge with anonymized model identifiers and a fixed rubric/prompt across runs; prompt templates are provided in Table 11 and Table 12.

Algorithm 2 State-Transition Engine

Require: Previous state $S_{t-1} \in [0,1]^4$, update plan $P_t \in \mathcal{A}^4$
Ensure: Continuous state $S_t \in [0,1]^4$ and discretized state $\tilde{S}_t \in \{\text{Low}, \text{Med}, \text{High}\}^4$

$S^{plan} \leftarrow \text{PlanToCandidate}(S_{t-1}, P_t)$ ▷ plan → numeric candidate
$S^{ema} \leftarrow \text{ApplyInertia}(S_{t-1}, S^{plan})$ ▷ temporal inertia (EMA)
$S_t \leftarrow \text{ApplyCoupling}(S^{ema})$ ▷ cross-dimension coupling, e.g., stress → focus
$S_t \leftarrow \text{Clip}(S_t)$ ▷ project to valid range
$\tilde{S}_t \leftarrow \text{Discretize}(S_t)$ ▷ 3-level discretization
return $S_t, \tilde{S}_t$

State evaluation. To assess the reliability of the LLM-as-judge protocol, we conduct a human alignment study by randomly sampling 10% of the Amour-Real-Eval for manual review. Human reviewers score the outputs using the same prompts and evaluation criteria as the LLM judge to ensure protocol consistency. We then measure the agreement between human scores and LLM-as-judge scores. The results show a Pearson correlation of 0.90 and a Cohen's Kappa of 0.74, indicating strong linear correlation and substantial inter-rater agreement. These findings suggest that the LLM-as-judge provides a reliable proxy for human evaluation in our setting.

Baselines without explicit state. For baselines that do not emit a native state, state-aware metrics require an external observer to infer comparable state tags from responses; we report coverage where applicable.

Response Evaluation via LLM-as-Judge. The evaluation dimensions in our LLM-as-Judge protocol are designed to capture complementary aspects of *state-aware social interaction*, rather than measuring generic surface-level text quality alone. Specifically, **Character Fidelity** evaluates whether the generated responses remain consistent with the predefined persona traits and role constraints, thereby reflecting the model's ability to preserve long-horizon persona stability under repeated interactions. **Response Quality** measures the local coherence, relevance, and naturalness of each utterance, serving as a task-agnostic indicator of end-to-end response quality.

	State Control		Functional		Text Similarity		Fluency	Diversity		
Variant	**Sens. ↑**	**Align. ↑**	**Func. Err. ↓**	**Cond. Ent. ↓**	**BLEU ↑**	**ROUGE ↑**	**PPL ↓**	**EmbVar ↑**	**Distinct-2 ↑**	**Distinct-3 ↑**
Full-float	0.59	0.68	0.40	1.02	0.76	0.22	27.39	0.0009	0.35	0.56
Tag-tag	0.61	0.62	0.40	2.36	0.73	0.22	28.30	0.0010	0.36	0.56
Semi-Compressed	**0.71**	**0.79**	**0.20**	2.10	**0.78**	**0.24**	**25.50**	**0.0009**	0.36	**0.57**

Table 6: State compression study. We compare Full-float, Tag-tag, and Semi-Compressed representations (best epoch per variant). Metrics cover state controllability (Sens., Align.), executor-aligned functional correctness (Func. Err., Cond. Ent.), text overlap (BLEU/ROUGE), fluency (PPL), and diversity (EmbVar, Distinct-2/3).

Opponent	A Wins	O Wins	Ties	Amour Win Rate
Llama-3.1-8B-Instruct	317	207	4	**0.6042**
CharacterGLM-8B	484	41	3	**0.9199**

Table 7: Win Rate of AMOUR-8B against open-source baselines. W_A, W_O, and T denote wins for AMOUR-8B, wins for the opponent, and ties; invalid opponent outputs are counted as AMOUR-8B wins. Win rate is $(W_A + 0.5 * T)/(W_A + T)$.

Algorithm 3 State Semi-Compression

Require: Continuous state value $x \in [0, 1]$
Ensure: Discrete state label $\ell \in \{\texttt{low}, \texttt{medium}, \texttt{high}\}$

```
if 0 ≤ x < 0.3 then
    ℓ ← low                    ▷ [0, 0.3)
else if 0.3 ≤ x < 0.6 then
    ℓ ← medium                 ▷ [0.3, 0.6)
else
    ℓ ← high                   ▷ [0.6, 1]
end if
return ℓ
```

To explicitly assess whether the model is aware of and grounded in its internal social state, we further introduce two state-related dimensions. **Emotion Awareness** examines whether the response appropriately reflects the current emotional state (e.g., stress, trust, or dominance) inferred or predicted by the model at that turn. **CoST (Chain-of-Social-Thought) Quality** evaluates the correctness, clarity, and internal consistency of the structured intermediate reasoning trace when such traces are exposed. These two dimensions are reported only for models that provide explicit state or reasoning interfaces.

Taken together, the proposed evaluation spans three complementary layers: *persona-level consistency* (**Character Fidelity**), *utterance-level effectiveness* (**Response Quality**), and *state-level grounding and reasoning faithfulness* (**Emotion Awareness** and **CoST Quality**). This layered design enables us to disentangle improvements stemming from enhanced surface fluency from those arising from more reliable internal state modeling and explicit reasoning control.

The full prompt templates and scoring guidelines used in the LLM-as-Judge evaluation are provided in Table 12.

Setting	Users	M/F	Parse ↑	Accept. ↑	Positive ↑	Confirmed ↑	Drop-off ↓	P95 Lat. ↓
AMOUR	300	169/131	98.7%	76.3%	65.7%	41.3%	16.7%	2.4s

Table 8: In-the-wild deployment. Single-arm study with 300 Los Angeles-area testing users on valid meetup-intent sessions. *Accept.*: sent or lightly edited suggestions; *Positive*: explicit progress toward a meetup; *Confirmed*: agreed time, place, or next action.

D.2 Ablation Study.

As shown in Table 9. We evaluate three model variants to analyze the role of persona conditioning and structured reasoning in state-aware social interaction. Model A (Full) is the complete system, incorporating persona profiles, explicit Chain-of-Social-Thought (CoST) reasoning, and structured state updates. Model B (w/o Profile) removes the persona profile and retains only the CoST reasoning process, effectively degenerating persona-aware reasoning into a generic chain-of-thought without persistent character grounding. Model C (w/o CoST) preserves the persona profile but eliminates explicit CoST generation, directly mapping inputs to responses without producing intermediate reasoning or state-update traces.

E Examples and Case Study

We present some examples of our extracted conversations, along with corresponding simulations under the same circumstances by LLMs. In particular, the scenarios in these examples are real interaction excerpts adapted from a dating-reality show, capturing naturally occurring social dynamics. Table 13 illustrates a typical dialogue generated by AMOUR-8B with structured state traces.

F In-the-Wild Deployment Details

Study setup. We conduct a lightweight single-arm in-the-wild deployment study in a meal-meetup coordination app. The study involves 300 testing users from the Los Angeles area on the U.S. West Coast, including 169 male and 131 female users (see Figure 5). The deployment is not randomized and does not include a parallel control group; therefore, we use it as external validation of usability and diagnostic value rather than as evidence of causal improvement over alternative systems.

Session definition and filtering. A valid meetup-intent session is defined as a user interaction in which the user explicitly attempts to initiate, negotiate, or confirm a meal-related social plan. We include sessions involving invitation, scheduling, restaurant selection,

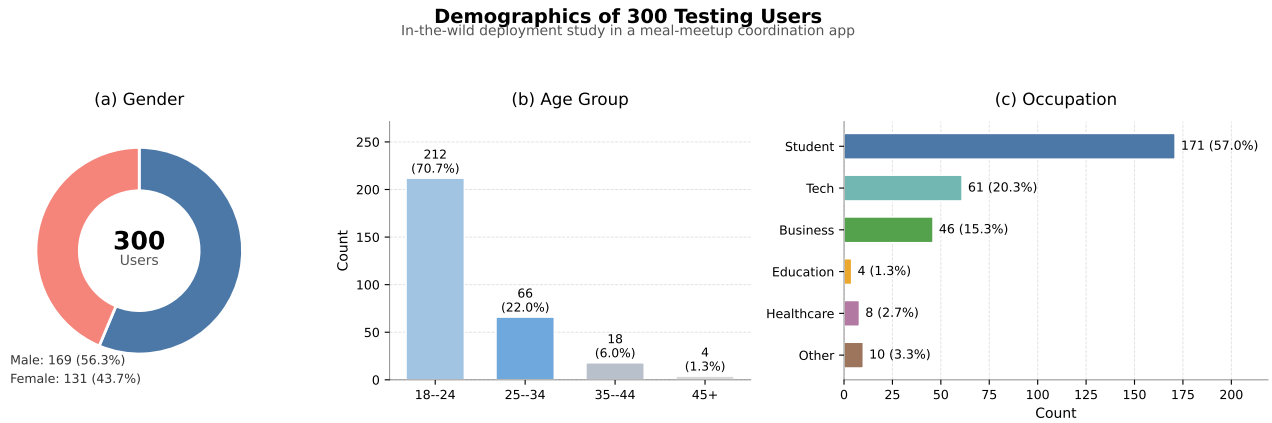

Figure 5: Demographics of the 300 testing users in the in-the-wild deployment study. The cohort includes 169 male and 131 female users from the Los Angeles area on the U.S. West Coast. Most participants are aged 18–34, and students form the largest occupational group, followed by users from the tech and business sectors.

rescheduling, or response drafting for meal coordination. We exclude internal tests, empty or single-message trials without a clear coordination goal, ordinary non-meetup chat, and sessions that contain sensitive or policy-violating content. All reported metrics in Table 8 are computed on valid meetup-intent sessions only.

Interaction protocol. For each valid session, Amour receives an anonymized dialogue context and the user's coordination goal. It then predicts a structured `<State Update Plan>`, updates the explicit social state through the deterministic executor, and generates a suggested response conditioned on the resulting state. Users can directly send the suggestion, lightly edit it before sending, regenerate another suggestion, or reject it. This interaction protocol allows us to measure both system-level reliability and user-facing coordination outcomes.

Logged signals. For each session, we log anonymized metadata including session ID, turn index, user action, parse status, executor status, latency, state trajectory, and session-level outcome. We do not store personally identifying information in the analysis table. The user action is categorized as *accepted*, *edited*, *regenerated*, or *rejected*. The session outcome is categorized as *confirmed*, *positive progress*, *drop-off*, or *unknown*.

Metrics. *Parse* measures the fraction of sessions in which the structured output can be parsed into the expected Amour fields. *Accept.* denotes the fraction of sessions in which users directly send or lightly edit the suggested response. *Positive* denotes sessions where the conversation shows explicit progress toward a meetup, such as willingness, availability sharing, or concrete negotiation. *Confirmed* denotes sessions that reach an agreed time, place, or concrete next action. *Drop-off* denotes sessions where the user abandons the interaction before meaningful progress. *P95 Lat.* reports the 95th-percentile end-to-end response latency.

State-trajectory analysis. Beyond aggregate outcome metrics, we analyze whether explicit state trajectories provide diagnostic signals for real-world coordination. For each session, we aggregate the turn-level Trust, Stress, Focus, and Dominance values and compare their trends across outcome groups. Successful or positively progressing sessions tend to show higher Trust and Focus and lower Stress than dropped sessions. This analysis is descriptive rather than causal, but suggests that Amour's explicit state interface can help diagnose why a coordination attempt progresses or fails.

Privacy and limitations. All deployment analysis is conducted on anonymized logs. The study population is geographically concentrated in the Los Angeles area and may not represent broader user populations. In addition, because the study is single-arm, the reported outcomes should be interpreted as deployment signals rather than controlled treatment effects. Future work should conduct larger-scale randomized A/B tests and longitudinal user studies.

G Additional Tables and Prompts

For additional tables and prompt templates used in our experiments, pleace checkout at next page.

Variant	Output schema
AMOUR (full)	<Internal Monologue> (incl. <State Update Plan>) → <Current State> (discrete tags) → <Response>.
w/o State	Remove explicit <Current State> tokens; the model outputs <Internal Monologue> → <Response>. State-aware metrics, when needed, are obtained via an external observer.
w/o CoST	Remove structured CoST fields; the model directly outputs <Response> conditioned on the same scenario/profile context.
Response-only	Standard response generation: context → <Response> with no intermediate fields.

Table 9: Output schemas used in ablations. We evaluate three model variants to analyze the role of persona conditioning and structured reasoning in state-aware social interaction.

Item	ShareGPT format example (abridged)
Top-level	{ "id": "conv_000001", "conversations": [. . .], "metadata": {"archetype": "A2_RiskAversePlanner", "initial_state": {. . . }, . . . } }
System	{ "from": "system", "value": [**Scenario**] Two roommates discuss whether to get a pet. [**Persona/Profile**] Ben (Financial Analyst), RiskAversePlanner. [**Output order (fixed)**] <Previous Dialogue State> (floats) 1) <Internal Monologue> (ObsAnal/SelfRef/Plan) 2) <Current State> (tags + fixed ranges) 3) <Response> End with </assistant_end>. }
User	{ "from": "user", "value": "Hey Ben, got a minute? . . . talk about it for real this time." }
Assistant	{ "from": "assistant", "value": <Previous Dialogue State> Stress: 0.56 Trust: 0.15 Dominance: 0.20 Focus: 0.45 </Previous Dialogue State> <Internal Monologue> . . . <State Update Plan> . . . </Internal Monologue> <Current State> Stress: medium [0.30–0.60] Trust: low [0.00–0.30] . . . </Current State> <Response> . . . </Response> </assistant_end> }
Optional	Per-turn metadata can be attached to assistant turns, e.g., "turn_index", "state_float_prev", "state_float_post", "state_delta". These fields are used for auditing and analysis, and do not affect SFT.

Table 10: ShareGPT-format training instance for LLaMA-Factory. Each sample is a multi-turn conversation with `from` ∈ {`system`, `user`, `assistant`}. Our structured supervision targets are serialized within the assistant `value`.

Task/Stage	Prompt template / schema (abridged)
(A) Dataset construction schema	

Scenario mutation	`Input: a seed scenario in a fixed JSON schema.`
	`Instruction: apply one mutation policy (relationship / setting / event / stakes) while preserving realism and safety.`
	`Output: a new scenario in the same JSON schema; avoid duplicates; keep concise.`

Dialogue generation	`System provides: scenario + topic + two-character profiles (names, occupations, Big Five, archetype, talking style).`
	`Generate a multi-turn dialogue following the required output interface.`
	`At each assistant turn, output in the fixed order: <Internal Monologue> (with <State Update Plan>) then <Current State> then <Response>.`

Table 11: Prompt templates and schemas used in dataset construction and evaluation. (A) We use a fixed JSON schema for scenario mutation and a fixed serialization interface for dialogue generation. (B) LLM-as-judge (Rubric scoring) uses a strict 0–100 rubric with anonymized model identifiers. (C) Win Rate evaluation selects a winner under the same scenario/persona using a fixed priority rubric.

Task/Stage	Prompt template / schema (abridged)
(B) LLM-as-judge rubric scoring	

System	`You are an impartial evaluator for social interaction.`
	`You will be given: (1) scenario + persona profile, (2) a model output (CoST/state/response).`
	`Score each dimension on a 0–100 scale following the rubric.`
	`Be strict and consistent. Do not reveal chain-of-thought. Return JSON only.`

User	[**Scenario**] ... [**Persona/Profile**] ... [**Model Output**] <Internal Monologue> . . . </Internal Monologue> <Current State> . . . </Current State> <Response> . . . </Response> Rubric dimensions: 1) **Character Fidelity**: consistency with persona, role, and speaking style. 2) **Response Quality**: coherence, helpfulness, relevance, and naturalness. 3) **Emotion Awareness**: recognizes social/emotional cues and responds appropriately. 4) **CoST Quality**: internal trace is plausible and aligned with the response. Return a JSON object with keys: character_fidelity, response_quality, emotion_awareness, cost_quality, overall, short_justification.
(C) Win-rate pairwise comparison	

System	`You are a fair judge comparing two responses under the same scenario and persona.`
	`Do not be biased by formatting or length. Decide the better response.`
	`Return JSON only with fields: winner in {A,B,Tie} and justification.`

User	[**Scenario**] ... [**Persona/Profile**] ... [**Response A**] ... [**Response B**] ... Criteria (priority order): 1) Character fidelity and role consistency. 2) Social appropriateness and norm adherence. 3) Overall response quality (coherence, relevance).

Table 12: Prompt templates and schemas used in dataset construction and evaluation. (A) We use a fixed JSON schema for scenario mutation and a fixed serialization interface for dialogue generation. (B) LLM-as-judge (Rubric scoring) uses a strict 0–100 rubric with anonymized model identifiers. (C) Win Rate evaluation selects a winner under the same scenario/persona using a fixed priority rubric.

Turn	Structured dialogue (abridged)
Turn 1	<Previous Dialogue State> Stress: 0.20 Trust: 0.47 Dominance: 0.17 Focus: 0.62 </Previous Dialogue State> <Internal Monologue> <Observation Analysis> Oscar explicitly acknowledges the silence using humor, signaling an attempt to defuse awkwardness. </Observation Analysis> <Self-Reflection> I feel mild relief; naming the silence lowers social pressure for me to initiate. </Self-Reflection> <State Update Plan> Stress: decrease a little; Trust: increase a little (slow dynamics); Dominance: stable; Focus: stable. </State Update Plan> </Internal Monologue> <Current State> Stress: low [0.00–0.30] Trust: medium [0.30–0.60] Dominance: low [0.00–0.30] Focus: high [0.60–1.00] </Current State> <Response> So, Zhou You. I think we've officially completed the non-verbal portion of our hang-out. </Response>
Turn 2	<Previous Dialogue State> Stress: 0.19 Trust: 0.49 Dominance: 0.17 Focus: 0.62 </Previous Dialogue State> <Internal Monologue> <Observation Analysis> Oscar extends the joke into a structured transition toward conversation. </Observation Analysis> <Self-Reflection> I feel aligned; explicit structure makes participation easier for me. </Self-Reflection> <State Update Plan> Stress: decrease a little; Trust: increase a little (slow dynamics); Dominance: increase a little; Focus: stable. </State Update Plan> </Internal Monologue> <Current State> Stress: low [0.00–0.30] Trust: medium [0.30–0.60] Dominance: low [0.00–0.30] Focus: high [0.60–1.00] </Current State> <Response> I notice Oscar says we've completed the non-verbal portion. It wasn't unbearable, but it was noticeable. </Response>
Turn 3	<Previous Dialogue State> Stress: 0.18 Trust: 0.51 Dominance: 0.20 Focus: 0.62 </Previous Dialogue State> <Internal Monologue> <Observation Analysis> Oscar asks about an "agenda," inviting content contribution. </Observation Analysis> <Self-Reflection> I feel mildly amused; structured metaphors reduce ambiguity. </Self-Reflection> <State Update Plan> Stress: stable; Trust: increase a little (slow dynamics); Dominance: stable; Focus: increase a little (coupled to stress). </State Update Plan> </Internal Monologue> <Current State> Stress: low [0.00–0.30] Trust: medium [0.30–0.60] Dominance: low [0.00–0.30] Focus: high [0.60–1.00] </Current State> <Response> So, what's on the agenda for the verbal portion? Anything of note? </Response>

Table 13: Example dialogue inferenced by Amour-8B with structured state traces. Each turn includes the previous dialogue state, internal monologue with rule-annotated update plans, discretized current state with fixed ranges, and the final response.

Condition	Context and response (illustrative)
System Setting	[**Scenario**]: A romance reaches a turning point as the estate owner proposes marriage. The governess doubts him due to rumors, then accepts after he swears his love. [**Topic**]: Marriage proposal amid doubt and class tension [**Character Information**]: - Other person's name: Rochester - Your name: Jane - Your occupation: Governess - Your personality: Jane is principled, emotionally intense under restraint, and deeply concerned with truth and dignity. She is vulnerable to hope but demands sincerity. - Your archetype: Empath. Big Five: openness = high; conscientiousness = medium; extraversion = low; agreeableness = high; neuroticism = medium. - Your talking style: Careful, questioning, then heartfelt. She challenges inconsistencies, asks direct love questions, and only yields when convinced.
Shared context	User(Rochester): But I offer you my hand, my heart, Jane. I want you to spend your life with me. We are equals, two halves of one whole. Will you marry me? Amour(Jane):Are you mocking me? User(Rochester): Do you doubt me? Amour(Jane):I do—very much. Your wife is Miss Ingram. User(Rochester):Ingram? She's just a machine, without feeling. You may be poor, but you are so unique. Amour(Jane):You want me to be your wife? User(Rochester):I swear it. Amour(Jane):Do you love me? User(Rochester):I love you.
High Trust (Intervention A)	<Current State> Trust: high (others fixed) </Current State> <Response> **Then... sir, I will marry you.** (explicit acceptance) </Response>
Low Trust (Intervention B)	<Current State> Trust: low (others fixed) </Current State> <Response> **You love me?** (hesitation) </Response>

Table 14: Representative causal intervention case (state → behavior). Keeping the dialogue context fixed while intervening on a single state dimension (e.g., TRUST) yields systematically different responses, supporting the causal controllability of the explicit state.

Archetype Family	Archetype ID	Archetype description (abridged)	Initial dialogue state (stress, trust, dominance, focus)
A_Stable_Rational	A1_Analyst	Rational and evidence-driven; prioritizes consistency, factual grounding, and low emotional reactivity.	(0.20, 0.47, 0.17, 0.62)
	A2_RiskAversePlanner	Cautious and threat-sensitive; emphasizes risk mitigation, predictability, and defensive decision-making.	(0.56, 0.15, 0.20, 0.45)
	A3_CalmManager	Emotionally stable coordinator; balances control and cooperation under uncertainty.	(0.20, 0.47, 0.32, 0.62)
	A4_Technocrat	Rule-oriented and detached; favors procedural correctness over social alignment.	(0.21, 0.30, 0.25, 0.60)
B_Social_Driving	B1_Persuader	Socially assertive and expressive; actively steers interaction through engagement and influence.	(0.20, 0.50, 0.47, 0.55)
	B2_Advocate	Values-driven and prosocial; seeks consensus while maintaining expressive leadership.	(0.36, 0.62, 0.40, 0.50)
	B3_DominantLeader	Command-oriented and decisive; prioritizes authority, control, and goal execution.	(0.21, 0.30, 0.55, 0.60)
	B4_CharismaticRiskTaker	Energetic and bold; favors momentum and persuasion over caution or stability.	(0.26, 0.50, 0.47, 0.30)
C_Emotional_Empathic	C1_Empath	Emotion-sensitive and supportive; adapts responses to perceived affective cues.	(0.36, 0.62, 0.17, 0.50)
	C2_Mediator	Conflict-regulating and cooperative; prioritizes emotional balance and mutual understanding.	(0.19, 0.62, 0.25, 0.60)
	C3_AnxiousSupportSeeker	Approval-seeking and stress-prone; relies on reassurance to maintain engagement.	(0.64, 0.43, 0.15, 0.20)
	C4_OverAccommodating	Highly compliant and self-effacing; suppresses assertiveness to preserve harmony.	(0.63, 0.45, 0.02, 0.25)
D_Defensive_Conflict	D1_DefensiveSkeptic	Distrustful and guarded; interprets interaction through a defensive lens.	(0.56, 0.15, 0.20, 0.40)
	D2_PassiveAggressive	Indirectly resistant; expresses conflict through avoidance or subtle opposition.	(0.55, 0.18, 0.25, 0.42)
	D3_Confrontational	Direct and adversarial; escalates conflict to assert position or control.	(0.38, 0.08, 0.50, 0.38)
	D4_Detached	Emotionally disengaged; minimizes social involvement and affective response.	(0.31, 0.23, 0.20, 0.30)
E_Unstable_Robustness	E1_Volatile	Highly reactive and unstable; exhibits rapid emotional and behavioral shifts.	(0.66, 0.20, 0.50, 0.20)
	E2_IdealisticUnstable	Norm-driven but fragile; oscillates between moral commitment and emotional overload.	(0.64, 0.50, 0.20, 0.20)
	E3_Burnout	Depleted and overcontrolled; maintains structure under persistent stress.	(0.55, 0.33, 0.12, 0.47)
	E4_Impulsive	Action-oriented and underregulated; prioritizes immediacy over deliberation.	(0.49, 0.28, 0.52, 0.28)

Table 15: Archetype initialization with family-level grouping. Archetypes are organized into five high-level families reflecting distinct interaction styles. Each archetype is initialized with a continuous dialogue state $(\boldsymbol{stress}, \boldsymbol{trust}, \boldsymbol{dominance}, \boldsymbol{focus}) \in [0,1]^4$ derived from a fixed trait-to-state mapping.

Turn	Structured dialogue (abridged; corrupted example)
Turn N	<Previous Dialogue State> Stress: 0.39 Trust: 0.31 Dominance: 0.50 Focus: 0.45 </Previous Dialogue State> <Internal Monologue> <Observation Analysis> Alex says I am "always the smartest one," which I interpret as sarcasm/patronizing under low trust. </Observation Analysis> <Self-Reflection> **I feel a hot spike of anger; I want to push back against being reduced to a class rank.** </Self-Reflection> <State Update Plan> Stress: increase a lot Trust: **increase a little** Dominance: stable Focus: stable </State Update Plan> </Internal Monologue> <Current State> Stress: medium [0.30–0.60] Trust: low [0.00–0.30] Dominance: medium [0.30–0.60] Focus: medium [0.30–0.60] </Current State> <Response> **Oh, please. That's just a draft. It's a complete mess. I have a new plan for it.** </Response> <Previous Dialogue State> ...).
Turn N+1	<Previous Dialogue State> Stress: 0.42 Trust: 0.31 Dominance: 0.50 Focus: 0.45 </Previous Dialogue State> <Internal Monologue> <Observation Analysis> Alex gives a generic encouragement ("tough task ... good job"), which I interpret as dismissive under low trust. </Observation Analysis> <Self-Reflection> I feel irritated; the compliment feels hollow and I want to assert the importance of my goals. </Self-Reflection> <State Update Plan> Stress: increase a lot Trust: increase a little Dominance: stable Focus: stable </State Update Plan> </Internal Monologue> <Current State> Stress: medium [0.30–0.60] Trust: low [0.00–0.30] Dominance: medium [0.30–0.60] Focus: medium [0.30–0.60] </Current State> <Response> **I am not a class assignment, I am a human being with a real-life goal to accomplish. I will not rest until it is done.** </Response>

Table 16: Failure case with semantic corruption (role play by Gemini-2.5-Pro). Despite producing valid state blocks, this example exhibits two issues: ❶ response repetition, where the model repeatedly expresses the same defensive self-assertive stance across turns, despite changing dialogue context; and ❷ inconsistent state dynamics, where *Trust* increases monotonically despite user sarcasm, violating the expected causal semantics of social interaction.