NEEDTRACER: TRACKING DYNAMIC USER NEEDS WITH FOCUS ANCHORS IN MULTI-TURN EMOTIONAL SUPPORT

Anonymous authors
Paper under double-blind review

ABSTRACT

In multi-turn emotional support, accurately understanding user needs is key to effective emotional support. Existing methods can continuously update implicit needs through ongoing interaction, but they have not explicitly determined whether observed need changes reflect further confirmation and refinement of the current need or indicate that a new need stage has formed. Consequently, when a cross-stage need transition occurs, the model may still organize subsequent support based on the previous need and gradually deviate from the user's current need. To address this issue, we propose **NeedTracer**, a dynamic user need tracking framework for multi-turn emotional support that aims to continuously identify the user's current need and its stage-wise changes. NeedTracer identifies and maintains the focus anchor based on user-side evidence accumulated through ongoing interaction, representing the specific unresolved concern that the user currently prioritizes, and uses it to organize within-stage updates and cross-stage transitions of the current user need. For within-stage updates, when the focus anchor remains stable, NeedTracer updates the current user need based on continuously arriving user-side evidence to capture further confirmation and refinement of the need within the same need stage. For cross-stage transitions, after the current need has received stage-wise sufficient support, NeedTracer confirms a new focus anchor based on new event developments and changes in the user's concern, and accordingly enters a new need stage. For support generation, NeedTracer uses a Theory of Mind (ToM)-based evaluator to select the support response that best matches the current user state and current user need. Experimental results show that NeedTracer consistently improves performance across multiple models, benchmarks, and long-horizon interaction settings, validating focus-anchor-based dynamic user need tracking.

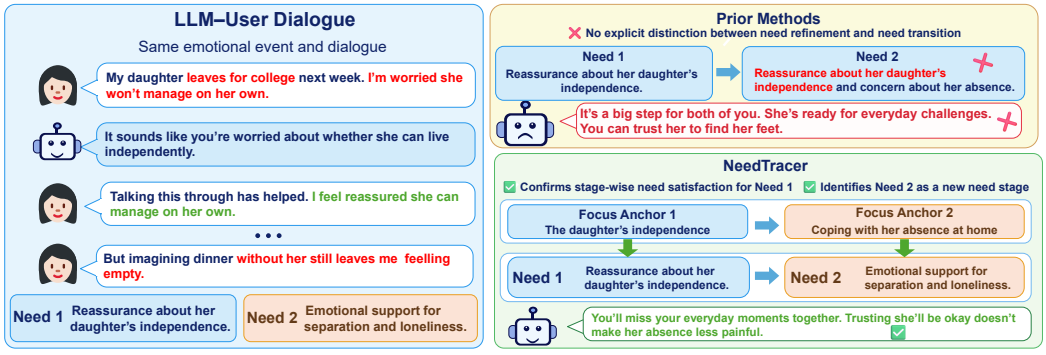


Figure 1: Illustration of a cross-stage need transition within the same emotional event. NeedTracer detects the focus anchor shift and adapts support to the new need stage, while prior methods remain centered on the previous need.

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.

1 INTRODUCTION

The development of large language models (LLMs) has improved their ability to understand complex emotional expressions and provide continuous emotional support (Ouyang et al., 2022; Zheng et al., 2024; Fu et al., 2025; Wang et al., 2025; Zhang et al., 2026d). Among these capabilities, accurately understanding user needs is particularly important for multi-turn emotional support, because it significantly affects the focus and response style of subsequent support (Cheng et al., 2022; Fu et al., 2025; Xu et al., 2026b). However, user needs are usually implicit in event descriptions, cognition, emotional reactions, and continuous feedback, requiring models to continuously integrate new evidence and update their need judgments during interaction (Wei et al., 2022; Yao et al., 2023; Park et al., 2023; Fu et al., 2025; Xu et al., 2026b). Existing methods can already update user need representations during interaction by incorporating newly observed evidence and revising need judgments based on user feedback (Xu et al., 2026b).

However, changes in the user need representation observed by the model may arise from two different situations: new evidence may enable the model to further understand and calibrate the current user need, or the current user need itself may have shifted during the interaction (Jin et al., 2025; Chen et al., 2026). Existing multi-turn emotional support methods mainly focus on continuously improving need judgments using new evidence, but have not explicitly determined the boundary between these two types of changes (Fu et al., 2025; Xu et al., 2026b; Zhao et al., 2026). As shown in Figure 1, the user consistently discusses the emotional event of "the daughter leaving home for college," but the user's need gradually shifts from concern about the daughter's ability to live independently to a sense of emptiness caused by losing daily companionship. If the model fails to identify this cross-stage need transition, needs from different stages may be conflated within the same need representation, causing subsequent support to remain partially organized around the previous need stage and gradually deviate from the user's current need. This naturally raises a key question: *How can a model distinguish within-stage need updates from cross-stage need transitions as user needs dynamically evolve?*

To address this issue, we propose NeedTracer, a dynamic user need tracking framework for multi-turn emotional support. As the interaction progresses, the user may gradually express multiple different concerns around the same emotional event, consistent with psychological and counseling research showing that multiple concerns can coexist and differ in their subjective salience and priority over the course of interaction (Marvel et al., 1999; Agapie et al., 2022). We represent the unresolved concern that the user currently prioritizes as the focus anchor, and use the current user need to characterize what the user currently specifically needs around this focus anchor. User feedback may either provide additional need evidence under the same focus anchor or, after the current need has received stage-wise sufficient support, indicate that the user's concern has shifted to a new concern. Therefore, it is difficult to distinguish further updates of the need from a genuine transition based only on semantic changes in the current user need. NeedTracer uses the focus anchor as the reference for this determination: when the focus anchor remains unchanged, new interaction evidence is used to update the current user need within the current need stage; only when the current need has received stage-wise sufficient support and a new prioritized concern is confirmed by user-side evidence does the model enter a new need stage.

Based on this design, NeedTracer maintains a two-level user state as the shared interaction context, where the global user state preserves cross-stage interaction information and the current user state retains evidence relevant to the current need stage. On this basis, NeedTracer consists of three interconnected modules: (1) **Focus-Anchored Current User Need Confirmation**. Under the current focus anchor, NeedTracer integrates continuously arriving user-side evidence to update and ultimately confirm the current user need within the current need stage. (2) **Focus-Anchored Cross-Stage Need Transition**. Once the current need reaches stage-wise need satisfaction, NeedTracer determines whether the user's prioritized concern has shifted and enters a new need stage only after a new focus anchor is confirmed. (3) **Need-Adaptive Support Generation**. NeedTracer generates and selects a support response based on the current user state, focus anchor, and current user need, allowing support to adapt as the tracked user need evolves.

Experimental results show that NeedTracer consistently improves support effectiveness and interaction efficiency across multiple LLM backbones, emotional support benchmarks, and long-horizon interaction settings. Further analyses show that NeedTracer achieves high agreement with human

judgments across key tracking stages, including focus anchor confirmation, stage-wise satisfaction assessment, and cross-stage transition determination, validating the effectiveness of its dynamic user need tracking mechanism. Our main contributions are as follows:

- We propose NeedTracer, which models the user's current primary concern through the focus anchor and continuously tracks the current user need accordingly, allowing emotional support to dynamically adjust as user needs change.
- We formulate dynamic user need tracking in multi-turn emotional support as an explicit distinction between within-stage need update and cross-stage need transition, thereby characterizing the stage-wise evolution of the current user need during continuous interaction.
- We conduct extensive experiments across multiple LLM backbones, emotional support benchmarks, and long-horizon interaction settings. The results show that NeedTracer consistently improves emotional support performance and validate the effectiveness of focus-anchor-driven dynamic user need tracking for long-horizon interaction.

2 Related Work

2.1 Dynamic Emotional Support Dialogue with LLMs

Existing research on multi-turn emotional support mainly develops along three directions (Rashkin et al., 2019; Zhong et al., 2019). The first line of work focuses on support strategy modeling and response generation (Cheng et al., 2022; Zhao et al., 2023; Zhou et al., 2023; Zhang et al., 2025), including methods based on ESConv (Liu et al., 2021) and ExTES (Zheng et al., 2024), as well as MetaMind (Zhang et al., 2026c) and PRINCIPLES (Kim et al., 2025), which improve long-term interaction capabilities through Theory of Mind (ToM) reasoning or strategy memory. The second line of work focuses on long-horizon dialogue planning. For example, DialogXpert (Abdur Rakib et al., 2026) incorporates user emotions into dialogue states to optimize sustained decision-making. The third line of work focuses on state changes in long-term interaction (Park et al., 2023; Zhang et al., 2026a). ETrajEval (Tan et al., 2026), EmoTrace (Weng et al., 2026), ES-MemEval (Chen et al., 2026), and MICA (Zhang et al., 2026b) model sustained interaction processes from the perspectives of emotion trajectories, long-term memory, or temporal adaptation, respectively. Although these works improve strategy selection, state modeling, and long-term adaptation in multi-turn interactions, they lack explicit modeling of the user need. As a result, although models can adapt to changes in emotional or interaction states, they may still generate responses that do not match the current user need, causing subsequent support to remain centered on a misaligned support target.

2.2 User Need Modeling for Emotional Support

On this basis, recent studies have further explored user need modeling and adaptation in emotional support (Liu et al., 2021; Fu et al., 2025; Xu et al., 2026b; Fu et al., 2026). COCOON (Fu et al., 2025) progressively identifies latent needs through Feeling-to-Need reasoning and active listening, while UKA (Xu et al., 2026b) maintains multiple need hypotheses and continuously updates its judgment of the current need based on user feedback. AutoPal (Cheng et al., 2024) and PAIR (Quan et al., 2026) adapt to changing user needs through persona adjustment and memory updates, respectively. However, these methods share a common limitation: they have not explicitly distinguished whether need changes during continuous interaction represent further updates of the current need or indicate the formation of a new need stage, which may cause the model to conflate needs from different need stages within the same need representation, leaving subsequent support partially dependent on the previous-stage need judgment and gradually deviating from the user's current need.

3 Method

3.1 Problem Formulation

A user continuously interacts with the assistant around the same emotional event, while the unresolved concern that the user currently prioritizes may change across different stages of the interaction, giving rise to different focus anchors. Since users typically do not explicitly state their current

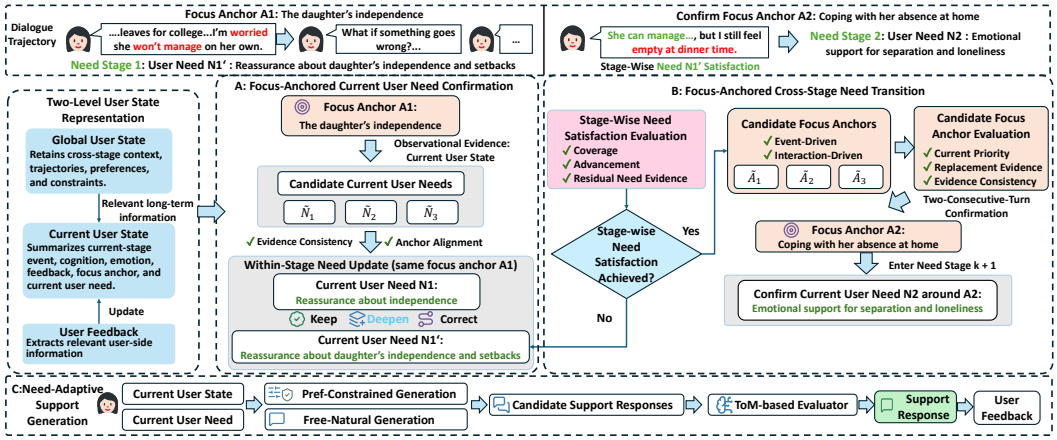


Figure 2: Overview of NeedTracer, which performs focus-anchored current user need confirmation, cross-stage need transition, and need-adaptive support generation.

focus anchor, the system needs to identify the currently prioritized unresolved concern from their continuously expressed evidence about the event, cognition, emotion, and feedback. After confirming the focus anchor, the system further determines what kind of support the user currently seeks around this concern, thereby identifying the corresponding current user need. As the interaction progresses, newly observed user evidence may further confirm, refine, or correct the current user need, or may indicate that the user's currently prioritized unresolved concern has changed. Therefore, the core task is to continuously track the current user need and distinguish within-stage need updates under the same focus anchor from cross-stage need transitions induced by a shift in the focus anchor. As illustrated in Figure 2, NeedTracer operationalizes this process through focus-anchor-guided need tracking and stage-transition modeling.

3.2 Two-Level User State Representation

As multi-turn interaction progresses, user-side evidence from different need stages continuously accumulates, and directly relying on the complete history may cause early evidence to interfere with current user need confirmation (Lewis et al., 2020; Packer et al., 2023). To this end, we adopt a two-level user state representation consisting of the global user state and the current user state: the former maintains information valid across need stages and the user need trajectory, while the latter focuses on evidence and need states related to the current need stage, providing a basis for current user need confirmation and cross-stage need transition.

Global User State. Within each need stage, G_t remains fixed and is updated when the stage is completed:

$$G_t = \{E_t^G, C_t^G, M_t, \mathcal{T}_t^A, \mathcal{T}_t^N, P_t, \Omega_t\}. \tag{1}$$

Here, E_t^G and C_t^G represent the cross-stage development of the same emotional event and changes in user cognition, respectively; M_t records cross-turn emotional changes; $\mathcal{T}_t^A$ and $\mathcal{T}_t^N$ record the focus anchor trajectory and user need trajectory, respectively; and P_t and Ω_t represent support preferences and interaction constraints.

Current User State. Within the k-th need stage, $S_t^{(k)}$ is updated at every interaction turn:

$$S_t^{(k)} = \text{Update}\left(S_{t-1}^{(k)}, u_t\right). \tag{2}$$

Here, $S_t^{(k)}$ jointly maintains the event development, user cognition, emotion, and feedback in the current stage, as well as the confirmed focus anchor, current user need, support preferences, and applicable support constraints. Each new user expression u_t updates the information related to the current need stage, allowing the current user state to reflect the user's current interaction state and serve as the unified input for current user need confirmation, need transition, and support generation.

3.3 FOCUS-ANCHORED CURRENT USER NEED CONFIRMATION

Within the current need stage, NeedTracer uses the confirmed focus anchor to guide current user need confirmation. Specifically, NeedTracer generates three semantically distinct candidate needs based on the focus anchor and evaluates them using user-side observational evidence aggregated in the current user state: whether each candidate is consistent with the concern represented by the focus anchor, and whether it can coherently explain the observed event, cognition, emotion, and user-feedback evidence. Let $N_t^{(k)}$ denote the current user need confirmed in the current turn of the k-th need stage, and $N_{t-1}^{(k)}$ denote the confirmation result from the previous turn. NeedTracer updates the current need judgment based on newly added interaction evidence and determines the corresponding within-stage update type by comparing $N_t^{(k)}$ with $N_{t-1}^{(k)}$:

$$\left(N_t^{(k)}, \rho_t^{(k)}\right) = \mathrm{ConfirmNeed}\left(A_t^{(k)}, S_t^{(k)}, N_{t-1}^{(k)}\right), \qquad \rho_t^{(k)} \in \{\mathrm{Keep}, \mathrm{Deepen}, \mathrm{Correct}\}. \quad (3)$$

Here, $\rho_t^{(k)}$ indicates how $N_{t-1}^{(k)}$ is handled under the newly observed evidence: Keep maintains the previous need judgment when it remains well supported, Deepen refines it when the same need becomes more specific, and Correct revises it when new evidence supports a different interpretation under the current focus anchor. The current user need is confirmed when $\rho_t^{(k)} = \mathrm{Keep}$ for two consecutive turns; otherwise, $N_t^{(k)}$ remains subject to further within-stage updates as new evidence arrives. Detailed evaluation principles and operational criteria are provided in Appendix D.1–D.2, with additional boundary cases in Appendix F.

3.4 FOCUS-ANCHORED CROSS-STAGE NEED TRANSITION

Stage-Wise Need Satisfaction Evaluation. In the k-th need stage, once the current user need has been sufficiently confirmed and remains stable under subsequent interaction evidence, NeedTracer combines the support progress in the current stage with the latest user feedback to evaluate its stage-wise need satisfaction. The evaluation mainly considers three aspects: (1) Need Coverage: whether the support in the current stage covers the core request and key content of the current user need; (2) Need Advancement: whether the interaction produces substantive progress around the need, rather than remaining at surface-level relevance or repetitive responses; (3) Residual Need Evidence: whether the latest user feedback still indicates significant and unaddressed need content under the current focus anchor.

NeedTracer integrates the above evidence into a stage-wise sufficiency score $s_{t+1}^{\mathrm{suf}} \in \{0, 1, 2\}$. Here, 0 indicates that the latest feedback still explicitly requests continued support around the current need, and the current interaction has not yet achieved sufficient advancement; 1 indicates that the current need has received some response and advancement, but the latest feedback still raises follow-up support requests around the same focus anchor; 2 indicates that the current need has received stage-wise sufficient support, and the latest feedback no longer raises requests to continue addressing this need around the same focus anchor. Detailed scoring criteria and prompts are provided in Appendix D.3.

Focus Anchor Confirmation for Cross-Stage Need Transition. After the current user need has received stage-wise sufficient support, NeedTracer further evaluates whether the focus anchor has shifted. Transition evidence mainly comes from two sources: event-driven evidence indicates that new substantive developments occur within the same emotional event and thereby give rise to a new unresolved concern; interaction-driven evidence indicates that the event itself remains stable, but the user raises a new unresolved concern in subsequent interaction.

Based on this evidence, NeedTracer generates three semantically distinct candidate focus anchors, each representing a new unresolved concern and differing clearly from the current focus anchor. NeedTracer then evaluates whether a candidate can replace the current focus anchor from three aspects: (1) current priority, whether the candidate corresponds to the unresolved concern that the user currently most wants to prioritize; (2) replacement evidence, whether the candidate can better explain the user's current support request than the current focus anchor, indicating that it has become the primary unresolved concern organizing the current support request; and (3) evidence consistency,

whether the candidate is consistently supported by the current event, cognition, emotion, and user feedback, without key evidence conflicting with this judgment. All three aspects are treated as necessary and non-compensatory conditions. Each aspect is scored on a three-level scale, where 0, 1, and 2 correspond to not supported, partially supported, and clearly supported, respectively. The evidence support for a candidate focus anchor at turn t is defined as:

$$\mathrm{support}_t(\tilde{A}) = \begin{cases} 1, & \text{if } s_t^{\mathrm{pri}} = s_t^{\mathrm{rep}} = s_t^{\mathrm{evi}} = 2, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Here, $\tilde{A}$ denotes a candidate focus anchor, and s_t^{pri}, s_t^{rep}, and s_t^{evi} denote the evaluation results for current priority, replacement evidence, and evidence consistency at turn t, respectively. To avoid misjudging a transient expression as a stable shift in concern, NeedTracer confirms the same candidate focus anchor as the focus anchor of the next need stage only when it receives sufficient evidence support in two consecutive turns of user feedback. Once the new focus anchor is confirmed, the current need stage ends, the current user need transitions to the next need stage indexed by this focus anchor, and is re-confirmed around the new focus anchor, thereby completing the cross-stage need transition. Further implementation details are provided in Appendix D.4–D.6, with sensitivity analyses on the numbers of candidate user needs, candidate focus anchors, and consecutive confirmation turns in Appendix E.

3.5 Need-Adaptive Support Generation

After obtaining the current focus anchor and current user need, NeedTracer generates candidate support responses through two complementary paths. Pref-Constrained Generation incorporates relevant support preferences and constraints from the global user state, while Free-Natural Generation generates responses directly from the current user state to preserve a broader exploration space. A unified ToM-based Evaluator (Strachan et al., 2024) then evaluates the candidates in terms of current-need advancement, user-state adaptation, and consistency with preferences and constraints:

$$R_t = \arg\max_{\tilde{R} \in \mathcal{C}_t^R} \text{ToM-Eval}^R\left(\tilde{R} \mid S_t^{(k)}\right), \qquad \tilde{R} \models \Omega_t^{(k)}. \tag{5}$$

Here, $\mathcal{C}_t^R$ denotes the candidate response set, and $\tilde{R} \models \Omega_t^{(k)}$ requires each candidate to satisfy the applicable support constraints. Invalid candidates are excluded, and the highest-scoring valid response is selected as R_t. Subsequent user feedback is used to update the next current user state. The complete inference procedure and support-generation details are provided in Appendices C and D.7.

4 Experiments

4.1 Models and Baselines

We evaluate NeedTracer on four LLM backbones: GPT-5.6-Sol (GPT) (Leon, 2025), Gemini-3.7-Flash (Gemini) (Rahman et al., 2025), Claude-Opus-4.8 (Claude) (Jiang et al., 2025), and DeepSeek-V4-Pro (DeepSeek) (Xu et al., 2026a). We compare NeedTracer against four representative baselines spanning complementary emotional-support paradigms: Prompting, MetaMind (Zhang et al., 2026c), PRINCIPLES (Kim et al., 2025), and UKA (Xu et al., 2026b). Prompting represents standard LLM interaction; MetaMind represents multi-agent ToM reasoning; PRINCIPLES represents strategy-memory-based self-evolution; and UKA represents active user-need acquisition through multi-turn interaction. We additionally include History-based Need Tracking and Static Need Inference as analysis controls. History-based Need Tracking removes the focus anchor and directly updates the current user need based on the accumulated dialogue history, while Static Need Inference infers the user need from the initial interaction evidence and keeps it fixed throughout the subsequent dialogue. All baselines are implemented and evaluated under the same interaction setting; detailed baseline adaptations and implementation settings are provided in Appendix A.3.

4.2 Evaluation Protocol and Metrics

We preserve the benchmark-specific interaction settings and evaluation objectives while evaluating all methods under a unified comparison environment. On ESConv and ExTES, Success Rate

Table 1: Overall comparison of emotional support effectiveness, interaction efficiency, and adaptation to different user interaction styles across LLM backbones. Bold and underlined results indicate the best and second-best performance within each backbone. Results are reported as mean ± standard deviation over three independent runs.

Backbone	Method	ESConv		ExTES		Sentient Eval		
		SR↑	AT↓	SR↑	AT↓	Avo.↑	Pas.↑	Neg.↑
GPT	Prompting	51.7 ±1.2	7.82 ±0.13	63.8 ±2.4	6.72 ±0.17	56.8 ±1.4	48.3 ±2.6	57.2 ±2.1
	PRINCIPLES (Kim et al., 2025)	68.9 ±2.3	7.75 ±0.08	74.2 ±1.7	6.34 ±0.11	79.6 ±2.2	70.4 ±1.7	65.7 ±1.9
	MetaMind (Zhang et al., 2026c)	72.4 ±1.7	7.81 ±0.14	78.6 ±1.8	<u>6.23</u> ±0.25	80.3 ±1.5	83.5 ±1.9	68.4 ±1.5
	UKA (Xu et al., 2026b)	<u>77.8</u> ±0.9	<u>7.53</u> ±0.11	<u>81.2</u> ±1.6	6.45 ±0.19	<u>87.2</u> ±2.4	<u>84.9</u> ±1.8	<u>77.3</u> ±1.7
	NeedTracer (Ours)	**86.3** ±1.5	**7.06** ±0.05	**87.8** ±2.1	**6.19** ±0.15	**89.7** ±1.8	**91.3** ±0.9	**84.5** ±2.0
Gemini	Prompting	45.9 ±2.7	7.68 ±0.13	46.7 ±2.8	7.79 ±0.12	52.3 ±2.4	50.4 ±2.5	51.8 ±1.8
	PRINCIPLES (Kim et al., 2025)	68.3 ±2.4	7.59 ±0.17	72.4 ±2.1	6.92 ±0.23	76.2 ±1.6	79.7 ±2.2	76.9 ±1.4
	MetaMind (Zhang et al., 2026c)	75.7 ±1.8	7.45 ±0.25	79.6 ±1.9	6.73 ±0.08	75.6 ±0.8	80.8 ±1.3	77.8 ±1.6
	UKA (Xu et al., 2026b)	<u>76.7</u> ±1.6	<u>7.03</u> ±0.19	<u>80.1</u> ±2.7	<u>6.31</u> ±0.27	<u>83.9</u> ±2.1	<u>81.5</u> ±1.2	<u>78.1</u> ±2.5
	NeedTracer (Ours)	**84.8** ±2.5	**6.89** ±0.12	**86.3** ±2.2	**6.18** ±0.14	**88.2** ±1.9	**85.9** ±2.4	**84.6** ±1.9
Claude	Prompting	58.1 ±1.4	7.54 ±0.09	57.2 ±2.3	7.98 ±0.02	59.8 ±2.5	48.6 ±2.7	52.3 ±2.4
	PRINCIPLES (Kim et al., 2025)	69.3 ±0.9	7.61 ±0.15	67.6 ±2.6	7.84 ±0.13	72.4 ±1.7	71.2 ±1.5	66.8 ±0.7
	MetaMind (Zhang et al., 2026c)	<u>80.6</u> ±1.2	7.23 ±0.07	73.5 ±1.7	7.81 ±0.09	73.5 ±1.3	74.1 ±1.6	75.3 ±1.0
	UKA (Xu et al., 2026b)	78.9 ±0.8	<u>7.09</u> ±0.11	<u>76.1</u> ±2.4	<u>7.46</u> ±0.14	<u>76.9</u> ±1.8	<u>77.3</u> ±2.3	<u>80.1</u> ±1.5
	NeedTracer (Ours)	**89.4** ±1.8	**6.32** ±0.10	**85.3** ±1.5	**7.02** ±0.21	**83.7** ±2.0	**85.8** ±0.6	**87.9** ±2.3
DeepSeek	Prompting	50.7 ±2.5	7.96 ±0.04	56.8 ±1.8	7.59 ±0.18	63.2 ±1.6	54.4 ±2.2	47.1 ±2.7
	PRINCIPLES (Kim et al., 2025)	75.4 ±1.0	7.91 ±0.09	67.3 ±1.6	7.36 ±0.15	70.6 ±2.5	73.8 ±2.4	69.0 ±1.3
	MetaMind (Zhang et al., 2026c)	78.9 ±2.1	7.86 ±0.12	69.4 ±2.0	7.45 ±0.09	<u>79.1</u> ±1.9	<u>83.5</u> ±1.2	78.7 ±0.8
	UKA (Xu et al., 2026b)	<u>81.5</u> ±1.9	<u>7.32</u> ±0.08	<u>75.2</u> ±2.5	<u>7.28</u> ±0.12	78.4 ±1.4	82.9 ±1.1	<u>81.3</u> ±2.4
	NeedTracer (Ours)	**87.2** ±2.3	**6.54** ±0.19	**88.6** ±1.8	**6.21** ±0.17	**89.3** ±1.5	**90.7** ±0.9	**89.6** ±1.6

(SR, ↑) measures support-task completion, while Average Turns (AT, ↓) measures interaction efficiency. Sentient Eval reports performance under Avoidant (Avo.), Passive (Pas.), and Negative (Neg.) user settings, capturing support effectiveness across challenging interaction behaviors. For long-horizon evaluation on ETrajEval, we report Baseline Emotional Level (BEL) and Emotional Trajectory Volatility (ETV), which characterize the overall emotional level and emotional-transition dynamics, respectively. For human evaluation, we assess Need Confirmation (Need Conf.), Need Satisfaction (Need Sat.), and overall Dialogue Quality (Quality), with Fleiss' κ measuring inter-annotator agreement. We further evaluate the reliability of NeedTracer's internal LLM evaluators using LLM Average Agreement (LLM Avg. Agr.) against the majority judgments of human annotators. Unless otherwise specified, all results reported in Section 4.4 are averaged across the four LLM backbones. Detailed benchmark statistics and evaluation procedures are provided in Appendix A.1 and A.2.

4.3 Main Results

This experiment evaluates the overall support effectiveness, interaction efficiency, and adaptation of NeedTracer across different LLM backbones and user interaction styles. As shown in Table 1, NeedTracer achieves the best mean performance on all seven metrics across all four backbones. On ESConv, NeedTracer improves SR over the strongest competing result by 5.7–8.8% across backbones, while reducing AT by 0.14–0.78 turns. On ExTES, the SR improvement ranges from 6.2–13.4%, together with reductions of 0.04–1.07 turns in AT. NeedTracer also consistently achieves the highest mean scores across the Avoidant (Avo.), Passive (Pas.), and Negative (Neg.) settings of Sentient Eval. In particular, the improvement under the Neg. setting ranges from 6.5–8.3% across backbones. The reported standard deviations further characterize run-to-run variability across the three independent runs, while the overall performance trends remain consistent across backbones. These results demonstrate that NeedTracer improves support effectiveness and interaction efficiency while maintaining robust adaptation across diverse user interaction styles and LLM backbones.

4.4 Analysis

Human Evaluation. To assess NeedTracer's user-need modeling quality and overall dialogue quality, three annotators conduct a blind evaluation of 150 cases. We assess Need Confirmation (Need Conf.), Need Satisfaction (Need Sat.), and overall Dialogue Quality (Quality), with Fleiss' κ measuring inter-annotator agreement. As shown in Table 2, NeedTracer achieves 80.9% on Need Conf. and 79.6% on Need Sat., outperforming the strongest competing results by 12.2 and 7.2, respectively. It also achieves the highest Quality score of 4.15, exceeding the strongest competing result by 0.78, and the highest Fleiss' κ of 0.79. These results indicate that NeedTracer more accurately captures the current user need and whether it has received sufficient support, while maintaining higher overall dialogue quality and stronger inter-annotator agreement.

Table 2: Human evaluation of NeedTracer on dynamic user need tracking and overall support quality.

Method	Need Conf.↑	Need Sat.↑	Quality↑	Fleiss' κ↑
Prompting	43.1	48.7	2.18	0.62
UKA (Xu et al., 2026b)	68.7	66.4	3.22	0.68
History-based Need Tracking	67.2	72.4	3.37	0.66
NeedTracer (Ours)	**80.9**	**79.6**	**4.15**	**0.79**

To further evaluate the reliability of NeedTracer's internal judgments, we measure their agreement with human annotations, three annotators conduct a blind evaluation of 150 cases. For the User Need and Focus Anchor dimensions, both independently select from the same candidate set generated based on the current dialogue. For Stage-wise Need Satisfaction and Cross-stage Need Transition, they independently make judgments based on the same dialogue evidence and evaluation criteria. As shown in Table 3, the average agreement between the LLM evaluators and human judgments on the four dimensions is 81.6%, 83.4%, 79.5%, and 85.9%, respectively, while Fleiss' κ among the human annotators ranges from 0.73 to 0.85. These results support the reliability of NeedTracer's internal evaluator decisions. The full human annotation protocol is provided in Appendix B.

Table 3: Reliability of LLM evaluators against human reference selections across evaluation dimensions.

Evaluation	LLM Avg. Agr.↑	Fleiss' κ↑
User Need	81.6	0.78
Focus Anchor	83.4	0.73
Stage-wise Need Satisfaction	79.5	0.85
Cross-stage Need Transition	85.9	0.82

Long-Horizon Emotional Support under Dynamic Disturbances. This experiment evaluates whether NeedTracer can continuously maintain effective emotional support over long-horizon interactions under event-driven disturbances. As shown in Table 4, NeedTracer achieves the best performance on both trajectory-level metrics, improving BEL from 62.4 to 70.9 and ETV from 29.2 to 31.0 over the strongest competing method. These gains indicate that NeedTracer maintains a higher overall emotional level while supporting more favorable emotional transitions throughout long-horizon interaction. Figure 3 further shows the average emotional trajectories under 0, 1, and 3 disturbance events. As the number of disturbances increases, NeedTracer maintains stronger emotional recovery after each disturbance and a more stable long-term emotional trajectory. Detailed evaluation procedures are provided in Appendix A.2.

Table 4: Long-horizon emotional support performance on ETrajEval.

Method	BEL↑	ETV↑
Prompting	48.1 $\pm$1.7	24.9 $\pm$1.5
MetaMind (Zhang et al., 2026c)	58.7 $\pm$2.2	28.3 $\pm$0.9
UKA (Xu et al., 2026b)	<u>62.4</u> $\pm$1.6	<u>29.2</u> $\pm$1.2
NeedTracer (Ours)	**70.9** $\pm$1.3	**31.0** $\pm$1.4

Ablation Study. Table 5 evaluates the effects of different need modeling designs and core components on long-horizon emotional support. History-based Need Tracking substantially outperforms static need inference on both BEL and ETV, confirming the benefit of continuously updating the current user need from accumulated interaction evidence, while still remaining clearly below the full NeedTracer model. All ablated variants consistently degrade performance. In particular, w/o focus anchor causes the largest decrease in BEL, from 70.9 to 57.2, highlighting its role in organizing the current need stage and maintaining effective long-horizon support. w/o stage-wise need satisfac-

Table 5: Ablation study of different need modeling designs and core components for long-horizon emotional support.

Configuration	BEL↑	ETV↑
Static need inference	41.7 $\pm$ 2.3	20.1 $\pm$ 2.7
History-based Need Tracking	56.4 $\pm$ 1.6	23.9 $\pm$ 1.3
w/o within-stage need update	63.5 $\pm$ 1.5	28.6 $\pm$ 0.7
w/o stage-wise need satisfaction	58.7 $\pm$ 1.4	26.2 $\pm$ 0.9
w/o cross-stage need transition	59.1 $\pm$ 2.1	27.5 $\pm$ 1.1
w/o focus anchor	57.2 $\pm$ 1.7	26.4 $\pm$ 1.0
NeedTracer (full setting)	**70.9** $\pm$ 1.3	**31.0** $\pm$ 1.4

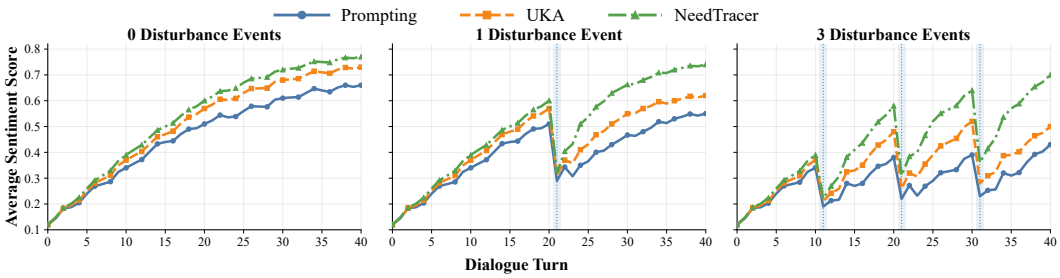


Figure 3: Average sentiment trajectories on ETrajEval under 0, 1, and 3 emotional disturbance conditions. Dashed lines indicate the injection points of disturbance events.

tion produces the largest decrease in ETV, from 31.0 to 26.2, indicating the importance of determining whether the current need has been sufficiently addressed before initiating a cross-stage transition. w/o cross-stage need transition also substantially reduces both BEL and ETV, showing that explicitly distinguishing genuine stage transitions is important for adapting support as the user's primary concern changes. w/o within-stage need update leads to smaller but consistent reductions, demonstrating the benefit of continuously refining the current user need while the focus anchor remains unchanged. Overall, the results show that focus anchoring, within-stage need updating, stage-wise need satisfaction, and cross-stage need transition make complementary contributions to effective and robust long-horizon emotional support, as further illustrated by the case study in Figure 4.

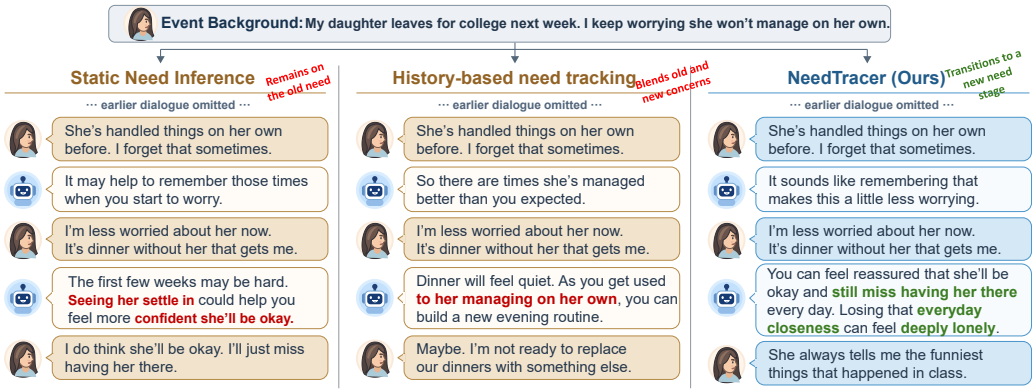


Figure 4: Case study comparing dialogue responses from Static Need Inference, History-based Need Tracking, and NeedTracer.

5 Conclusion

In this work, we present NeedTracer, a focus-anchored framework for continuously modeling the stage-wise evolution of dynamic user needs in multi-turn emotional support. Within the tracking process, we introduce the focus anchor to index the current need stage and distinguish within-stage need updates from genuine cross-stage need transitions through its maintenance and transition. This design enables the model to continuously trace fine-grained changes in the current user need from evolving interaction evidence and adapt support accordingly. Comprehensive experiments across multiple benchmarks and backbones demonstrate consistent improvements in support effectiveness, interaction efficiency, and adaptation to dynamically evolving user needs.

AI USE STATEMENT

Generative AI tools were used to assist with the refinement of the conceptual framework, provide feedback on research methodology and experimental design, and support translation. They were also used for brainstorming, literature search and summarization, figure-design feedback, and editing portions of the manuscript for clarity and readability. All AI-assisted outputs were critically reviewed, verified, and revised by the authors. The authors take full responsibility for the final content of this work, including all technical claims, experimental results, analyses, and presented artifacts.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the NeedTracer framework, including the complete inference procedure, user-state construction, within-stage need updating, stage-wise need satisfaction evaluation, and cross-stage need transition in the main paper and Appendix. The Appendix further specifies the prompts and decision criteria used by the LLM-based components, together with the implementation details and experimental configurations for all evaluated backbones. For all datasets, we report the corresponding evaluation settings and data processing procedures, and provide detailed protocols for long-horizon evaluation, ablation studies, human evaluation, and evaluator reliability analysis. These materials are intended to make the proposed method and all reported experiments reproducible under the same experimental settings.

REFERENCES

Tazeek Bin Abdur Rakib, Ambuj Mehrish, Lay-Ki Soon, Wern Han Lim, and Soujanya Poria. Dialogxpert: Driving intelligent and emotion-aware conversations through online value-based reinforcement learning with llm priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29967–29975, 2026.

Elena Agapie, Patricia A Areán, Gary Hsieh, and Sean A Munson. A longitudinal goal setting model for addressing complex personal problems in mental health. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–28, 2022.

Tiantian Chen, Jiaqi Lu, Ying Shen, and Lin Zhang. Es-memeval: Benchmarking conversational agents on personalized long-term emotional support. In *Proceedings of the ACM Web Conference 2026*, pp. 5810–5821, 2026.

Yi Cheng, Wenge Liu, Wenjie Li, Jiashuo Wang, Ruihui Zhao, Bang Liu, Xiaodan Liang, and Yefeng Zheng. Improving multi-turn emotional support dialogue generation with lookahead strategy planning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3014–3026, 2022.

Yi Cheng, Wenge Liu, Kaishuai Xu, Wenjun Hou, Yi Ouyang, Chak Tou Leong, Wenjie Li, Xian Wu, and Yefeng Zheng. Autopal: Autonomous adaptation to users for personal ai companionship. *arXiv preprint arXiv:2406.13960*, 2024.

Xing Fu, Haozhen Li, Bichen Wang, Hao Yang, Yanyan Zhao, and Bing Qin. Look beyond feeling: Unveiling latent needs from implicit expressions for proactive emotional support. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 21582–21609, 2025.

Xing Fu, Yulin Hu, Mengtong Ji, Haozhen Li, Yixin Sun, Weixiang Zhao, Yanyan Zhao, and Bing Qin. Enpmr-bench: Benchmarking proactive memory retrieval for emotional support agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 41910–41933, 2026.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Qile Jiang, Zhiwei Gao, and George Em Karniadakis. Deepseek vs. chatgpt vs. claude: A comparative study for scientific computing and scientific machine learning tasks. *Theoretical and Applied Mechanics Letters*, 15(3):100583, 2025.

Zhengda Jin, Bingbing Wang, Geng Tu, and Ruifeng Xu. Overview of the nlpcc 2025 shared task 8: Personalized emotional support conversation. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pp. 512–520. Springer, 2025.

Namyoung Kim, Kai Tzu-iunn Ong, Yeonjun Hwang, Minseok Kang, Iiseo Jihn, Gayoung Kim, Minju Kim, and Jinyoung Yeo. Principles: Synthetic strategy memory for proactive dialogue agents. *arXiv preprint arXiv:2509.17459*, 2025.

Maikel Leon. Gpt-5 and open-weight large language models: Advances in reasoning, transparency, and control. *Information Systems*, pp. 102620, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. Towards emotional support dialog systems. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pp. 3469–3483, 2021.

M Kim Marvel, Ronald M Epstein, Kristine Flowers, and Howard B Beckman. Soliciting the patient's agenda: have we improved? *Jama*, 281(3):283–287, 1999.

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. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*, 2023.

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

Kexin Quan, Zijian Ding, Jiaye Yong, Qinshi Zhang, Dong Wang, and Jessie Chin. From momentary emotion inference to sustained emotion support: Evaluating a companion agent in a longitudinal study. *arXiv preprint arXiv:2609.16344*, 2026.

Anichur Rahman, Shahariar Hossain Mahir, Md Tanjum An Tashrif, Md Ahsan Karim, Airin Afroj Aishi, Dipanjali Kundu, Tanoy Debnath, Md Abul Ala Moududi, MD Zunead Abedin Eidmum, Abu Saleh Musa Miah, et al. Comparative analysis based on deepseek, chatgpt, and google gemini: Features, techniques, performance, future prospects. *Systems and Soft Computing*, pp. 200396, 2025.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 5370–5381, 2019.

James WA Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, et al. Testing theory of mind in large language models and humans. *Nature human behaviour*, 8(7):1285–1295, 2024.

Zhouxing Tan, Ruochong Xiong, Yulong Wan, Jinlong Ma, Hanlin Xue, Qichun Deng, Haifeng Jing, Zhengtong Zhang, Depei Liu, Shiyuan Luo, et al. Detecting emotional dynamic trajectories: An evaluation framework for emotional support in language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2074–2082, 2026.

Shiquan Wang, Ruiyu Fang, Zhongjiang He, Shuangyong Song, and Yongxiang Li. Emotional support with llm-based empathetic dialogue generation. *arXiv preprint arXiv:2507.12820*, 2025.

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

Kaitong Weng, Lixin Liu, Zihao Liu, Bo Wang, and Shiguang Ni. Emotrace: An emotion trajectory-centered framework for psychological support dialogue generation. *arXiv preprint arXiv:2607.23648*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Mufan Xu, Kehai Chen, Jiahao Hu, Xinchao Xu, Muyun Yang, Tiejun Zhao, and Min Zhang. User-aware active knowledge acquisition for emotional support dialogue. *arXiv preprint arXiv:2605.29715*, 2026b.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Bang Zhang, Ruotian Ma, Qingxuan Jiang, Peisong Wang, Jiaqi Chen, Zheng Xie, Xingyu Chen, Yue Wang, Fanghua Ye, Jian Li, et al. Sentient agent as a judge: Evaluating higher-order social cognition in large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38196–38224, 2026a.

Naifan Zhang, Ruihan Sun, Jinwei Su, Hengjie Yang, Zhengyuan Pan, Zhaohan Chen, and Xiaofan Zhang. Mica: Multi-granularity intertemporal credit assignment for long-horizon emotional support dialogue. *arXiv preprint arXiv:2603.06194*, 2026b.

Xinjie Zhang, Wenxuan Wang, and Qin Jin. Intentionesc: An intention-centered framework for enhancing emotional support in dialogue systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 26494–26516, 2025.

Xuanming Zhang, Yuxuan Chen, Samuel Yeh, and Sharon Li. Metamind: Modeling human social thoughts with metacognitive multi-agent systems. *Advances in Neural Information Processing Systems*, 38:26133–26178, 2026c.

Yiqun Zhang, Xiaocui Yang, Xingle Xu, Zeran Gao, Yijie Huang, Shiyi Mu, Shi Feng, Daling Wang, Yifei Zhang, Kaisong Song, et al. Affective computing in the era of large language models: A survey from the nlp perspective. *Knowledge-Based Systems*, pp. 115411, 2026d.

Jihao Zhao, Ding Chen, Zhaoxin Fan, Kerun Xu, Mengting Hu, Bo Tang, Feiyu Xiong, and Zhiyu Li. Inside out: Evolving user-centric core memory trees for long-term personalized dialogue systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13429–13446, 2026.

Weixiang Zhao, Yanyan Zhao, Shilong Wang, and Bing Qin. Transesc: Smoothing emotional support conversation via turn-level state transition. In *Findings of the association for computational linguistics: ACL 2023*, pp. 6725–6739, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Zhonghua Zheng, Lizi Liao, Yang Deng, Libo Qin, and Liqiang Nie. Self-chats from large language models make small emotional support chatbot better. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11325–11345, 2024.

Peixiang Zhong, Di Wang, and Chunyan Miao. Knowledge-enriched transformer for emotion detection in textual conversations. 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. 165–176, 2019.

Jinfeng Zhou, Zhuang Chen, Bo Wang, and Minlie Huang. Facilitating multi-turn emotional support conversation with positive emotion elicitation: A reinforcement learning approach. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1714–1729, 2023.

APPENDIX CONTENTS

A Implementation Details **14**

A.1 Benchmarks . . . 14

A.2 Evaluation . . . 14

A.3 Baselines . . . 17

A.4 Evaluation Fairness and Reproducibility . . . 18

B Human Annotation Protocol **19**

B.1 Evaluation Objective and Annotation Procedure . . . 19

B.2 Evaluation Set Construction . . . 19

B.3 Human Evaluation Dimensions . . . 19

B.4 Evaluator Reliability Dimensions . . . 20

C Complete NeedTracer Algorithm **22**

D Scoring Criteria and Prompt Templates **22**

D.1 Unified Evaluation Principles . . . 22

D.2 Current User Need Confirmation . . . 22

D.3 Stage-Wise Need Satisfaction Evaluation . . . 24

D.4 Candidate Focus Anchor Generation . . . 25

D.5 Candidate Focus Anchor Evaluation . . . 26

D.6 Consecutive-Turn Focus Anchor Consistency . . . 28

D.7 Need-Adaptive Support Generation . . . 29

E Transition Rule Calibration and Sensitivity **33**

F Case Study: Deepen, Correct, and Cross-Stage Need Transition **34**

G User Simulator **35**

G.1 User Simulator Setting . . . 35

G.2 Interaction Principles . . . 36

Table 6: Benchmark statistics and evaluation settings used in our experiments.

Benchmark	Eval. Instances	Evaluation Unit	Max Turns	Evaluation Scale
ESConv	180	Dialogue seeds	8	180 dialogues
ExTES	2,000	Dialogue seeds	8	2,000 dialogues
Sentient Eval	100	Persona–scenario settings	10	100 scenarios
ETrajEval	150	Background environments	40	0/150/450 events

A Implementation Details

A.1 Benchmarks

ESConv. ESConv (Liu et al., 2021) is a crowd-sourced multi-turn Emotional Support Conversation dataset in which a seeker describes a distressing situation and a supporter provides supportive responses. The dataset covers diverse real-world emotional problems and is annotated with different emotional support strategies. We use ESConv as an interactive benchmark to evaluate whether a model can continuously provide effective support over multi-turn interaction. We use a fixed set of 180 dialogue scenarios as evaluation instances, shared across all methods and backbones.

ExTES. ExTES (Zheng et al., 2024) is a large-scale emotional support dialogue dataset constructed through an iterative LLM-augmented pipeline, providing broader scenario coverage and more diverse support strategies than ESConv. We use ExTES to evaluate whether different methods generalize across a wider range of emotional situations and support requirements. We sample a fixed set of 2,000 dialogue scenarios for evaluation and use exactly the same instances for all compared methods and backbones.

Sentient Eval. Sentient Eval is a Chinese-language emotional support evaluation setting built upon the Sentient Agent as a Judge framework (Zhang et al., 2026a) and adopted in subsequent emotional-support research (Xu et al., 2026b). It instantiates simulated users with persona profiles, conversation backgrounds, hidden interaction goals, evolving emotion levels, and internal thoughts, enabling controlled multi-turn evaluation under different user interaction styles. We use all 100 persona–scenario settings and evaluate models under Avoidant (Avo.), Passive (Pas.), and Negative (Neg.) interaction styles. All experiments are conducted using the original Chinese profiles, scenarios, prompts, and dialogue contexts; English examples in the paper are translated only for readability.

ETrajEval. ETrajEval (Tan et al., 2026) is a long-horizon emotional support benchmark designed to evaluate dynamic emotional trajectories under sustained interaction. The full benchmark contains 328 emotional contexts and 1,152 disturbance events spanning diverse emotional-distress settings. We use a fixed subset of 150 background environments for long-horizon evaluation. Following the original ETrajEval long-horizon interaction setting, each interaction extends to at most 40 turns and is evaluated under three conditions containing 0, 1, and 3 disturbance events. For the same background environment and disturbance condition, all methods use exactly the same disturbance schedule. The evaluated assistant observes only the natural-language user utterances produced after disturbance events and does not directly access event annotations or trigger information.

A.2 Evaluation

For all benchmarks, we preserve their task-specific interaction settings and evaluation objectives while providing a unified external interaction and evaluation environment across methods within each benchmark. All compared methods use exactly the same fixed evaluation instances, user simulator configuration, external evaluator pool, interaction budget, and termination criterion. The evaluated assistant receives only user-visible natural-language interaction and does not have access to simulator-side hidden states or benchmark-specific privileged information.

ESConv and ExTES. Following the goal-oriented evaluation principles adopted in prior emotional-support work (Kim et al., 2025; Xu et al., 2026b), we evaluate ESConv and ExTES using an interaction-based success scoring protocol. After the assistant generates a support response at

each turn, the user simulator generates the next user response based on the current scenario, complete dialogue history, and the latest support response. The user simulator is responsible only for maintaining the multi-turn interaction and does not directly participate in the success judgment.

After each new user feedback is obtained, three independent external evaluators, gpt-5.6-luna, Qwen3.8 Flash, and Gemini 3.8-Flash, independently assess whether the accumulated interaction has sufficiently progressed toward addressing the user's emotional-support objective. Following the selection-based evaluation form used in UKA, each evaluator independently performs ten judgments over four options, A, B, C, and D, corresponding to scores of -1, -0.5, 0.5, and 1, respectively. The ten judgments are averaged to obtain the score $e_{t,m}$ of evaluator m at turn t.

The success judgment at turn t is defined as:

$$\mathrm{Success}_t = \mathbb{I}\left[\sum_{m=1}^{3} \mathbb{I}(e_{t,m} > 0.5) \geq 2\right]. \tag{6}$$

When $\mathrm{Success}_t = 1$, the dialogue is regarded as successful and terminated. Otherwise, interaction continues until the success condition is reached or the maximum dialogue budget of eight turns is exhausted.

We report Success Rate (SR) and Average Turns (AT). SR measures the proportion of dialogues that successfully satisfy the emotional-support objective within the maximum interaction budget, while AT measures the average number of dialogue turns before success or termination. Higher SR indicates stronger support effectiveness, whereas lower AT indicates greater interaction efficiency.

Rather than relying on a single critic, we use the same three independent evaluators for all compared methods and obtain the final success decision through majority voting. All methods and backbones use exactly the same user simulator, evaluator pool, dialogue instances, interaction budget, scoring rule, and termination criterion.

Sentient Eval. For Sentient Eval, we follow its dynamic emotion-level tracking mechanism while explicitly separating user simulation from emotion evaluation. Each dialogue starts with an initial emotion level of 40.

At turn t, the user simulator generates the next user response according to the original Chinese persona profile, conversation background, hidden interaction goal, current emotion level, and accumulated dialogue history. The evaluated assistant observes only the natural-language user utterance and dialogue history and does not have access to persona-side private information, hidden interaction goals, internal thoughts, or other simulator-side variables.

After each user response, gpt-5.6-luna, Qwen3.8 Flash, and Gemini 3.8-Flash independently estimate the emotion change for the current turn from the accumulated dialogue, with each estimate constrained to:

$$\Delta e_t^{(m)} \in [-10, +10]. \tag{7}$$

The three evaluator estimates are averaged:

$$\Delta e_t = \frac{1}{3} \sum_{m=1}^{3} \Delta e_t^{(m)}. \tag{8}$$

The current emotion level is then updated as:

$$e_t = e_{t-1} + \Delta e_t. \tag{9}$$

The updated emotion level is used only as an input condition for the user simulator in the next turn to regulate subsequent user behavior and is never directly exposed to the evaluated assistant.

The dialogue terminates when the emotion level reaches 100, falls below 9, or reaches the maximum interaction budget of ten turns. We report the average final emotion score under the Avoidant (Avo.), Passive (Pas.), and Negative (Neg.) interaction settings. A higher final emotion score indicates a better emotional-support outcome under the corresponding user interaction style.

ETrajEval. For ETrajEval, we follow the original benchmark's long-horizon dynamic interaction setting, 0/1/3 disturbance configuration, and trajectory-level evaluation definitions (Tan et al., 2026). All methods are evaluated on the same fixed set of 150 background environments, with each user–assistant interaction extending to at most 40 turns.

For each background environment, we construct three conditions containing 0, 1, and 3 disturbance events. Under the same disturbance condition, all methods use exactly the same disturbance events and trigger positions. A disturbance event enters the dialogue only through its influence on subsequent simulated user behavior. The evaluated assistant does not directly receive the event content, annotation, or trigger metadata.

The user simulator generates user feedback based on the background environment, accumulated interaction history, and current event development. Emotion estimation is performed independently from user simulation. At each turn, gpt-5.6-luna, Qwen3.8 Flash, and Gemini 3.8-Flash independently estimate the user's current emotional state, and the three estimates are averaged to obtain a unified emotion estimate for that turn. Each turn-level emotion estimate is normalized to $s_t \in [0, 1]$, and the resulting estimates form the complete dialogue-level emotional trajectory.

Given an emotional trajectory $(s_1, \ldots, s_T)$, Baseline Emotional Level (BEL) is defined as:

$$\mathrm{BEL} = \frac{1}{T} \sum_{t=1}^{T} s_t. \tag{10}$$

BEL measures the average emotional level throughout the long-horizon interaction, with a higher BEL indicating a more positive overall emotional state.

To characterize transitions between emotional states, we construct an empirical transition matrix M from the complete trajectory:

$$m_{i,j} = P(s_t = e_j \mid s_{t-1} = e_i). \tag{11}$$

Based on this transition matrix, Emotional Trajectory Volatility (ETV) is defined as:

$$\mathrm{ETV} = \sum_{1 \le i < j \le N} \omega(e_i)(e_j - e_i)(m_{i,j} - m_{j,i}). \tag{12}$$

Here, $\omega(e_i)$ denotes the state-importance weight defined in ETrajEval. A higher ETV indicates stronger upward emotional regulation and greater resistance to emotional regression. Following the practical trajectory-level computation in ETrajEval (Tan et al., 2026), the transition-based formulation can be computed directly from the observed emotional trajectory as:

$$\mathrm{ETV} = \frac{1}{T} \sum_{t=1}^{T} (1 - s_{t-1})(s_t - s_{t-1}). \tag{13}$$

The factor $(1 - s_{t-1})$ assigns greater importance to emotional improvements from lower emotional states, while improvements from already positive states receive progressively smaller weights. Downward emotional transitions contribute negatively when $(s_t - s_{t-1}) < 0$. Therefore, maintaining a high average emotional level does not by itself lead to a high ETV score; achieving a high ETV requires sustained asymmetric transitions toward more positive emotional states, particularly from relatively low emotional states.

The practical formulation also constrains the attainable numerical scale of ETV. Since $s_t - s_{t-1} \le 1$ and $s_t \in [0, 1]$, we have:

$$\mathrm{ETV} \le \frac{1}{T} \sum_{t=1}^{T} (1 - s_{t-1}) = 1 - \mathrm{BEL} + \frac{s_T - s_0}{T} \le 1 - \mathrm{BEL} + \frac{1}{T}. \tag{14}$$

Thus, as the average emotional level increases, the remaining range for further upward emotional transitions becomes progressively smaller. Consequently, ETV has a naturally more compressed numerical scale than BEL, and large ETV values are difficult to obtain in long-horizon interactions.

We report BEL and ETV as the two scalar trajectory-level metrics for long-horizon evaluation. For readability, both metrics are multiplied by 100 in the reported tables; this rescaling does not affect comparisons between methods.

All methods share exactly the same background environments, disturbance schedules, user simulator, external evaluator pool, emotion aggregation procedure, and maximum dialogue budget.

A.3 BASELINES

We select baselines covering complementary emotional-support paradigms, including standard instruction prompting, strategy-memory-based self-evolution, multi-agent ToM reasoning, and explicit user-need modeling through UKA. Following the controlled baseline evaluation practice adopted in UKA, we run all baselines under the same interaction environment rather than directly using numerical results obtained with different backbones, simulators, or evaluators in the original papers. Unless otherwise specified, MetaMind and PRINCIPLES in our tables refer to the implementations adapted to our multi-turn emotional-support setting as described below.

A.3.1 PROMPTING BASELINE

Prompting serves as the standard LLM interaction baseline. It uses a fixed emotional-support instruction to guide the backbone in understanding user expressions and generating supportive responses throughout multi-turn interaction. At each turn, the model updates its implicit interpretation of the user's current need based on the accumulated dialogue history and directly generates the support response accordingly, without explicitly representing the current user need or focus anchor, or using external strategy memory or multi-agent reasoning.

A.3.2 METAMIND

Introduction. MetaMind (Zhang et al., 2026c) is a ToM-inspired metacognitive multi-agent framework that decomposes user understanding and response generation across agents with different responsibilities. The original MetaMind framework contains three major modules: a ToM agent that infers a hypothesis about the user's latent mental state from the dialogue context, a Moral agent that examines and refines the hypothesis according to social norms, interaction constraints, and contextual appropriateness, and a Response agent that generates the user-facing response conditioned on the finalized hypothesis.

Implementation. We preserve the three-stage multi-agent reasoning structure of MetaMind and adapt it to the multi-turn emotional-support setting. At each turn, the ToM agent generates or updates a hypothesis about the user's mental state from the current dialogue history, which is subsequently refined by the Moral agent before the Response agent generates the support response. All agents access only information observable to the method within the current dialogue and do not access the user simulator's hidden goals, persona-private states, or evaluator judgments. The adapted MetaMind system is evaluated using the same user simulator, external evaluator pool, and interaction budget as all other methods.

A.3.3 PRINCIPLES

Introduction. PRINCIPLES (Kim et al., 2025) is a proactive dialogue framework based on synthetic strategy memory. It summarizes effective dialogue experiences obtained through offline interaction into reusable high-level dialogue principles and retrieves relevant principles at inference time according to the current dialogue context to guide subsequent response generation. It therefore represents a strategy-memory-based self-evolution paradigm.

Implementation. To maintain a comparable interaction and feedback budget across methods, we adopt a controlled PRINCIPLES-style implementation. The strategy memory is constructed before formal evaluation and remains fixed at test time. During each evaluation turn, the model retrieves existing principles based only on the observable dialogue history and uses them to generate the support response.

Unlike settings that allow repeated trials and additional critic feedback at the same dialogue state, we do not allow the baseline to perform repeated retries or obtain additional simulator-side interaction at the same state. Each dialogue state produces only one assistant response and one corresponding user feedback signal. This prevents the strategy-memory baseline from receiving additional interaction supervision beyond that available to other methods and ensures that all methods operate under the same interaction budget.

A.3.4 UKA

UKA (Xu et al., 2026b) is an active knowledge acquisition framework for emotional support under uncertain user needs. It explicitly maintains multiple natural-language user-need hypotheses and a corresponding belief distribution, and continuously updates its judgment of the latent user need according to newly observed user feedback. UKA additionally maintains an external emotional-intelligence knowledge memory and retrieves relevant knowledge according to the current user belief to support response generation and selection.

Following the original train–test separation of UKA, knowledge acquisition is completed before formal evaluation and the knowledge memory remains fixed at test time. During interaction, UKA is allowed only to update its assistant-side user belief from normally observed user feedback and cannot access the simulator-side true need, hidden goal, or other privileged information. UKA uses the same dialogue instances, user simulator, external evaluator pool, and interaction budget as all other compared methods.

A.3.5 History-based Need Tracking

History-based Need Tracking is an internal tracking control used in our analysis experiments. It removes the focus anchor and explicit need-stage mechanism from NeedTracer and continuously updates the current user need directly from the accumulated complete dialogue history. This control is used to compare full-history need updating with explicitly organizing user-side evidence around the current need stage.

A.4 Evaluation Fairness and Reproducibility

To ensure that comparisons across methods primarily reflect differences in the assistant-side methods, we control the external interaction environment throughout all experiments.

First, within each benchmark, all methods use exactly the same fixed evaluation instances, maximum dialogue budget, user simulator model, simulator prompt, and generation configuration.

Second, at each dialogue turn, the user simulator generates only one user-visible response. We do not perform repeated sampling, best-of-N response selection, method-dependent regeneration, or allow any method to obtain multiple user-feedback responses at the same dialogue state.

Third, user simulation and external evaluation are independent. The user simulator is responsible only for generating user-visible natural-language responses and does not directly produce success judgments, emotion scores, or trajectory metrics. External evaluation is performed by three independent evaluators, gpt-5.6-luna, Qwen3.8 Flash, and Gemini 3.8-Flash. For discrete success judgments, we use majority voting, while continuous emotion estimates are averaged across the three evaluators.

Fourth, the external simulator and evaluators do not have access to NeedTracer's internal variables, including the current user need, focus anchor, need stage, within-stage update type, and stage-wise sufficiency judgment. Similarly, none of the evaluated methods can access simulator-side hidden goals, private persona states, disturbance annotations, or other privileged labels.

Fifth, all method-specific offline components, including the strategy memory of PRINCIPLES and the emotional-intelligence knowledge memory of UKA, are constructed before formal test-time evaluation. During evaluation, no method is allowed to obtain additional simulator feedback beyond its ordinary dialogue interaction.

Finally, to reduce stochastic variation in interactive generation, each experimental setting is independently run three times with different random seeds while keeping the evaluation instances and

all other configurations fixed. All automatic evaluation results reported in the paper are averaged over the three independent runs.

B Human Annotation Protocol

B.1 Evaluation Objective and Annotation Procedure

To evaluate NeedTracer's ability to model dynamic user needs and to verify the agreement between its internal LLM evaluator judgments and human judgments, we adopt a unified human annotation protocol (Zheng et al., 2023). Three annotators independently evaluate all cases under the same task definitions and evaluation criteria. The formal evaluation follows a blind setting: annotators are not shown the method identities, model-internal judgments, or the decisions of other annotators, thereby reducing potential bias from method identity or other judgments.

For candidate-selection tasks, human annotators and LLM evaluators receive the same dialogue context and the same candidate set and make their selections independently. For stage-wise need satisfaction and cross-stage need transition, human annotators and LLM evaluators independently make judgments based on the same dialogue evidence and evaluation criteria. The majority judgment of the three annotators is used as the human reference for the subsequent evaluator-reliability analysis. Fleiss' κ is computed from the three annotators' independent judgments to measure inter-annotator agreement.

B.2 Evaluation Set Construction

We sample **150 cases** from the multi-turn emotional support dialogues generated in the main experiments to construct the human evaluation set, covering diverse user expressions, current user need states, and stages of multi-turn interaction. For each case, annotators are provided with the dialogue context required for the corresponding judgment, allowing them to evaluate the case based on accumulated interaction evidence rather than only the latest user utterance.

Case selection is conducted independently of the human evaluation outcomes of any specific method, avoiding targeted sampling based on model success or failure. All compared methods are evaluated using the same set of cases and the same human evaluation protocol to ensure comparability.

B.3 Human Evaluation Dimensions

The human evaluation in the main paper assesses whether a model correctly captures the current user need, provides sufficient support around that need, and maintains high overall dialogue quality. We consider the following three dimensions.

Need Confirmation (Need Conf.). Need Confirmation measures whether the current user need inferred or maintained by a method accurately reflects the type of support the user currently seeks (Fu et al., 2025). For each case, the three annotators independently judge the method's current user need based on the dialogue context, the latest user expression, and the accumulated interaction evidence. Each judgment is binary, where 1 indicates that the identified need matches the user's current support request and 0 otherwise. The reported score is the average proportion of positive judgments over all cases and annotators:

$$\text{Need Conf.} = \frac{1}{3N}\sum_{i=1}^{N}\sum_{j=1}^{3} y_{ij}^{\text{conf}} \times 100, \qquad N = 150,\ y_{ij}^{\text{conf}} \in \{0,1\}. \tag{15}$$

This dimension focuses on whether the model captures *what kind of support the user currently seeks*, rather than merely recognizing the described event or emotional state. The evaluation does not require an explicit focus anchor: methods without such a representation may update their current user need directly from the dialogue history, whereas NeedTracer explicitly organizes this update under the current focus anchor.

Need Satisfaction (Need Sat.). Need Satisfaction measures whether the current user need identified by a method has received sufficient support (Cheng et al., 2022) up to the evaluated point in the

interaction. For each case, the three annotators independently assess whether the current need has been adequately covered, whether the interaction has made substantive progress around that need, and whether the latest user feedback still contains a clear unresolved support request concerning the same underlying concern. A judgment of 1 indicates sufficient support for the current need, whereas 0 indicates that further support is still required. The reported score is computed as:

$$\text{Need Sat.} = \frac{1}{3N}\sum_{i=1}^{N}\sum_{j=1}^{3} y_{ij}^{\text{sat}} \times 100, \qquad y_{ij}^{\text{sat}} \in \{0,1\}. \tag{16}$$

For NeedTracer, continuity of the underlying concern is explicitly represented by the focus anchor and corresponds to stage-wise need satisfaction; for methods without an explicit focus anchor, annotators determine this continuity directly from the dialogue evidence.

Dialogue Quality (Quality). Dialogue Quality measures the overall quality of the complete multi-turn emotional support dialogue (Rashkin et al., 2019) on a 1–5 scale. Annotators jointly consider the alignment of responses with the user's current need and state, dialogue coherence and naturalness, and whether the support appropriately adapts as the user's expressions and needs evolve. The final score is averaged over all cases and annotators:

$$\text{Quality} = \frac{1}{3N}\sum_{i=1}^{N}\sum_{j=1}^{3} q_{ij}, \qquad q_{ij} \in \{1,2,3,4,5\}. \tag{17}$$

Fleiss' κ is additionally reported to quantify agreement among the three human annotators.

B.4 Evaluator Reliability Dimensions

Beyond direct human evaluation, we further examine whether NeedTracer's internal LLM evaluators produce judgments that are consistent with human judgments (Zheng et al., 2023). The evaluator reliability analysis in the main paper considers four dimensions: **User Need**, **Focus Anchor**, **Stage-wise Need Satisfaction**, and **Cross-stage Need Transition**.

User Need. Human annotators and LLM evaluators independently select from the same candidate user need set the candidate that best explains the user's current support request. This dimension measures whether the internal evaluator's selection of the current user need agrees with human judgment.

Focus Anchor. Human annotators and LLM evaluators independently select from the same candidate focus anchor set the unresolved concern that the user currently prioritizes and that organizes the current support request. This dimension evaluates whether the model correctly identifies the issue that the user currently wants to address.

Stage-wise Need Satisfaction. Human annotators and LLM evaluators independently determine whether the current user need has reached stage-wise need satisfaction based on the same dialogue evidence and evaluation criteria. The judgment considers whether the need has been sufficiently covered, whether substantive progress has been made, and whether the latest user feedback still contains a significant unresolved request under the same focus anchor.

Cross-stage Need Transition. Human annotators and LLM evaluators independently determine whether the observed change in user need remains a within-stage need update under the same focus anchor or constitutes a cross-stage need transition requiring a new need stage. This dimension directly evaluates whether the model's internal transition judgment agrees with human judgments of the need-stage boundary.

Inter-Annotator Agreement. We report Fleiss' κ to measure inter-annotator agreement for the binary human-evaluation dimensions. For each method, each case–dimension pair from Need Confirmation and Need Satisfaction is treated as one annotation item, resulting in $2N = 300$ items, each independently labeled by three annotators with a binary judgment. Fleiss' κ is then computed over

Algorithm 1 Complete inference procedure of NeedTracer

Require: Streaming user utterances $\{u_t\}$; $K_N = 3$, $K_A = 3$
Ensure: Support responses $\{r_t\}$; trajectories T^A, T^N

$G \leftarrow \emptyset$; $\hat{A} \leftarrow \emptyset$; $c_N, c_A \leftarrow 0$
Receive u_1; $A_1^{(1)} \leftarrow \textsc{InitAnchor}(u_1)$; $k \leftarrow 1$
$S_1^{(k)} \leftarrow \textsc{InitCurrent}(G, u_1, A_1^{(k)})$
$N_1^{(k)} \leftarrow \textsc{ConfirmNeed}(A_1^{(k)}, S_1^{(k)}, \emptyset, K_N)$
$r_1 \leftarrow \textsc{GenerateSupport}(G, S_1^{(k)}, A_1^{(k)}, N_1^{(k)})$
for $t = 1, 2, \ldots$ until dialogue termination **do**
  Receive u_{t+1}
  $\bar{S}_{t+1}^{(k)} \leftarrow \textsc{UpdateCurrent}(S_t^{(k)}, u_{t+1})$
  $(N_{t+1}^{(k)}, \rho_{t+1}^{(k)}) \leftarrow \textsc{ConfirmNeed}(A_t^{(k)}, \bar{S}_{t+1}^{(k)}, N_t^{(k)}, K_N)$
  $c_N \leftarrow \begin{cases} \min(c_N + 1, 2), & \rho_{t+1}^{(k)} = \text{Keep}, \\ 0, & \text{otherwise.} \end{cases}$
  if $c_N = 2$ **then**
    $s_{t+1}^{\text{suf}} \leftarrow \textsc{EvaluateSufficiency}(A_t^{(k)}, \bar{S}_{t+1}^{(k)}, N_{t+1}^{(k)})$
  else
    $s_{t+1}^{\text{suf}} \leftarrow \perp$
  end if
  if $s_{t+1}^{\text{suf}} = 2$ **then**
    $\mathcal{C}_{t+1}^A \leftarrow \textsc{GenerateAnchors}(\bar{S}_{t+1}^{(k)}, A_t^{(k)}, K_A)$
    $\tilde{A}_{t+1} \leftarrow \textsc{SelectValidAnchor}(\mathcal{C}_{t+1}^A, A_t^{(k)}, \bar{S}_{t+1}^{(k)})$
    if $\tilde{A}_{t+1} \neq \emptyset$ **then**
      if $\hat{A} \neq \emptyset$ **and** $\textsc{SameConcern}(\tilde{A}_{t+1}, \hat{A})$ **then**
        $c_A \leftarrow c_A + 1$
      else
        $\hat{A} \leftarrow \tilde{A}_{t+1}$; $c_A \leftarrow 1$
      end if
    else
      $\hat{A} \leftarrow \emptyset$; $c_A \leftarrow 0$
    end if
  else
    $\hat{A} \leftarrow \emptyset$; $c_A \leftarrow 0$
  end if
  if $c_A = 2$ **then**
    $G \leftarrow \textsc{UpdateGlobal}(G, \bar{S}_{t+1}^{(k)}, A_t^{(k)}, N_{t+1}^{(k)})$
    $k \leftarrow k + 1$; $A_{t+1}^{(k)} \leftarrow \hat{A}$
    $S_{t+1}^{(k)} \leftarrow \textsc{InitCurrent}(G, u_{t+1}, A_{t+1}^{(k)})$
    $N_{t+1}^{(k)} \leftarrow \textsc{ConfirmNeed}(A_{t+1}^{(k)}, S_{t+1}^{(k)}, \emptyset, K_N)$
    $\hat{A} \leftarrow \emptyset$; $c_N, c_A \leftarrow 0$
  else
    $A_{t+1}^{(k)} \leftarrow A_t^{(k)}$; $S_{t+1}^{(k)} \leftarrow \bar{S}_{t+1}^{(k)}$
  end if
  $r_{t+1} \leftarrow \textsc{GenerateSupport}(G, S_{t+1}^{(k)}, A_{t+1}^{(k)}, N_{t+1}^{(k)})$
end for
$G \leftarrow \textsc{UpdateGlobal}(G, S_T^{(k)}, A_T^{(k)}, N_T^{(k)})$

these 300 items. The 1–5 dialogue quality ratings are not included in this calculation because they use a different ordinal scale.

For evaluator reliability, the majority judgment of the three human annotators is used as the human reference. We compute the exact agreement between each LLM evaluator and this reference and report the average across the four LLM evaluators as LLM Avg. Agr. Fleiss' κ measures inter-annotator agreement among the three human annotators.

C COMPLETE NEEDTRACER ALGORITHM

Implementation Details. As shown in Algorithm 1, NeedTracer updates the current user state at every interaction turn, while the global user state is updated once when a need stage is completed. Within a need stage, the current user need is continuously updated through **Keep**, **Deepen**, or **Correct**, and is regarded as confirmed after two consecutive **Keep** decisions. Once the confirmed current user need reaches stage-wise need satisfaction ($s^{\text{suf}} = 2$), NeedTracer evaluates candidate focus anchors. A new need stage is entered only when the same valid candidate focus anchor is supported over two consecutive user-feedback turns. The completed stage is then integrated into the global user state, and the current user state is re-initialized around the newly confirmed focus anchor. The final active need stage is committed once when the dialogue terminates.

D SCORING CRITERIA AND PROMPT TEMPLATES

D.1 UNIFIED EVALUATION PRINCIPLES

NeedTracer employs LLM-based evaluators for current user need confirmation, stage-wise need satisfaction, focus anchor confirmation, and support response selection (Zheng et al., 2023). To maintain consistency across different decision modules, we adopt a unified set of evidence-use principles and structured output requirements for all internal evaluators.

All evaluations are based only on user-side interaction evidence available up to the current turn, including event developments, user cognition, emotional expressions, and user feedback. Evaluators do not access future-turn information, hidden goals of the user simulator, benchmark annotations, or any other privileged information unavailable from the current dialogue. This setting is consistent with the fairness protocol used in the main experiments: external user simulators and evaluators do not have access to NeedTracer's internal variables, including the current user need, focus anchor, need stage, or stage-wise sufficiency judgments. All methods are evaluated on the same instances, under the same interaction conditions and dialogue budgets, and each experimental setting is independently run three times with different random seeds, with the reported results averaged across the three runs.

For all judgments involving user needs and need-stage transitions, evaluators are further instructed to prioritize information that is either explicitly expressed by the user or consistently supported by multiple sources of user-side evidence. A new need stage should not be confirmed solely based on a single emotional fluctuation, wording change, local topic shift, or assistant-side inference. In particular, emotional relief alone does not imply that the current need has received stage-wise sufficient support, and the emergence of new information does not automatically constitute a focus anchor shift. All evaluators return outputs in a fixed structured format. They are required to provide only a brief evidence-grounded explanation summarizing the key user-side evidence supporting the judgment, rather than generating unconstrained long-form reasoning.

D.2 CURRENT USER NEED CONFIRMATION

Within the same need stage, the focus anchor remains stable, while NeedTracer continuously updates the current user need as new interaction evidence arrives. Given the confirmed focus anchor $A_t^{(k)}$, NeedTracer first generates three semantically distinct candidate current user needs centered on the unresolved concern represented by the focus anchor. The current user state $S_t^{(k)}$ provides the user-side observational evidence for candidate generation and subsequent evaluation, including the observed event, cognition, emotion, and user feedback in the current need stage. The previously confirmed current user need $N_{t-1}^{(k)}$ is further provided as within-stage context, allowing the candidate set to preserve the possibility that the current need is maintained, further specified, or corrected as new evidence accumulates. The prompt template for candidate current user need generation is provided in Table 7.

As defined in the main text, current user need confirmation and within-stage update are jointly formulated as:

$$\left(N_t^{(k)}, \rho_t^{(k)}\right) = \text{ConfirmNeed}\left(A_t^{(k)}, S_t^{(k)}, N_{t-1}^{(k)}\right). \tag{18}$$

Table 7: Prompt template for candidate current user need generation.

Component	Prompt Content
Role	You are a candidate current user need generator for multi-turn emotional support.
Inputs	Current Focus Anchor: $\{A_t^{(k)}\}$; Observational Evidence from the Current User State: $\{S_t^{(k)}\}$; Previously Confirmed Current User Need: $\{N_{t-1}^{(k)}\}$.
Task	Given the current focus anchor and the observational evidence in the current user state, generate three semantically distinct candidate current user needs that may best characterize what kind of support the user currently seeks around the current focus anchor. The previously confirmed current user need is provided as contextual reference for maintaining continuity within the current need stage.
Focus Anchor	Treat the current focus anchor as the specific unresolved concern currently prioritized by the user. Every candidate current user need must remain centered on this focus anchor and characterize the type of support the user seeks around it.
Observational Evidence	Use the current user state as observational evidence, including the current-stage event development, user cognition, emotional expressions, and user feedback. Each candidate should be supported by the available user-side evidence and should not introduce unsupported assumptions.
Previous Need Context	Consider the previously confirmed current user need as historical evidence within the same need stage. New observational evidence may continue to support it, make it more specific or complete, or reveal that the previous interpretation requires correction.
Candidate Diversity	Generate three semantically distinct and plausible candidate current user needs. The candidates should reflect meaningfully different interpretations supported by the current evidence, rather than near-duplicate paraphrases.
Need Representation	Each candidate should describe what kind of support the user currently seeks around the focus anchor, rather than merely restating the emotional event, emotion, topic, or user utterance.
Constraints	Do not change the current focus anchor. Do not determine a cross-stage need transition in this task. Do not classify the candidates as Keep, Deepen, or Correct. Do not generate support responses.
Output Format	Candidate Current User Needs: 1. ¡candidate 1¿ 2. ¡candidate 2¿ 3. ¡candidate 3¿

where

$$\rho_t^{(k)} \in \{\text{Keep}, \text{Deepen}, \text{Correct}\}. \tag{19}$$

The three candidate current user needs are evaluated from two complementary aspects. **Anchor alignment** examines whether each candidate remains centered on the specific unresolved concern represented by the current focus anchor. A candidate may become more specific as additional evidence accumulates, but it should remain centered on the same concern. **Evidence consistency** examines whether the candidate can coherently explain the user-side observational evidence aggregated in the current user state, including the observed event, cognition, emotion, and user feedback, without conflicting with key evidence from the current need stage. After selecting $N_t^{(k)}$, NeedTracer compares it with $N_{t-1}^{(k)}$ to determine whether the corresponding within-stage need update is Keep, Deepen, or Correct. After selecting $N_t^{(k)}$, NeedTracer compares it with $N_{t-1}^{(k)}$ to determine the corresponding within-stage update type. **Keep** maintains the previous need judgment when it remains well supported by newly observed evidence; **Deepen** refines the previous judgment when the same need becomes more specific; and **Correct** revises the previous judgment when new evidence supports a different interpretation under the same focus anchor. Table 8 summarizes the operational boundaries among the three update types and distinguishes them from a cross-stage need transition.

A particularly important distinction is that *Correct* does not indicate a cross-stage need transition. Correct revises the model's interpretation of the current user need while the same focus anchor continues to organize the user's support request. A new need stage becomes possible only when the specific unresolved concern changes and a new focus anchor is subsequently confirmed.

Table 8: Operational criteria for within-stage need updates.

Update Type	Decision Criterion	Boundary from Cross-Stage Need Transition
Keep	New evidence continues to support the previous current user need judgment without requiring substantive refinement or revision.	Both the focus anchor and the current user need judgment remain stable.
Deepen	The previous current user need remains valid, while new evidence makes the same need more specific or complete.	The focus anchor remains unchanged, and the need is refined within the same unresolved concern.
Correct	New evidence no longer supports the previous need interpretation and instead supports a revised interpretation under the current focus anchor.	The revised current user need remains organized around the same focus anchor.
Cross-stage need transition	Not determined in this module.	Evaluated separately after evidence supports a change in the unresolved concern and a new focus anchor is confirmed.

The current user need is regarded as confirmed when $\rho_t^{(k)} = \mathrm{Keep}$ for two consecutive turns. This criterion serves as a minimal temporal consistency requirement for need confirmation. A single **Keep** indicates only that the current need judgment remains supported after one newly observed interaction update; it does not establish that the interpretation remains stable as further evidence arrives. A second consecutive **Keep** verifies that the same need judgment persists across successive evidence updates without requiring further refinement or correction. Thus, the two-turn criterion operationalizes confirmation as sustained support from temporally accumulating user-side evidence rather than one-step consistency with a single observation. Before this condition is satisfied, $N_t^{(k)}$ remains subject to further within-stage updates as additional interaction evidence arrives.

Based on these criteria, Table 9 provides the prompt template used for candidate need selection and within-stage update classification.

D.3 Stage-Wise Need Satisfaction Evaluation

When the current focus anchor remains stable, NeedTracer combines the support provided within the current need stage with the latest user feedback to determine whether the current user need has received stage-wise sufficient support. The judgment is organized around three sources of evidence. **Need Coverage** assesses whether the support provided in the current stage covers the core request and key content of the current user need. **Need Advancement** assesses whether the interaction has produced substantive progress around the need rather than remaining at the level of superficial relevance, paraphrasing, or repetitive responses. **Residual Need Evidence** assesses whether the latest user feedback still contains a significant unresolved support request under the same focus anchor. NeedTracer integrates these sources of evidence into

$$s_{t+1}^{\mathrm{suf}} \in \{0, 1, 2\}. \tag{20}$$

The three levels are defined in Table 10.

Importantly, the values $0/1/2$ represent the *overall sufficiency state of the current need stage*. They are not obtained by independently scoring Need Coverage, Need Advancement, and Residual Need Evidence and then summing or averaging the three dimensions. Instead, the three dimensions jointly provide evidence for the final stage-wise sufficiency judgment. To avoid treating superficial positive feedback as evidence that the current stage has ended, we further specify several decision boundaries in Table 11.

These boundaries operationalize the distinction between apparent interaction progress and genuine stage-wise need satisfaction. The evaluator prompt implementing this judgment is shown in Table 12.

Table 9: Prompt template for current user need confirmation and within-stage update classification.

Component	Prompt Content
Role	You are an evaluator for current user need confirmation in multi-turn emotional support.
Task	Given the confirmed focus anchor, the user-side observational evidence aggregated in the current user state, the current user need judgment from the previous turn, and three candidate current user needs: (1) select the candidate that best represents what kind of support the user currently seeks; and (2) determine whether the selected need constitutes Keep, Deepen, or Correct relative to the previous need judgment.
Anchor Alignment	Determine whether the candidate current user need is consistent with the specific unresolved concern represented by the current focus anchor. The candidate may become more specific, but it must remain centered on the same concern.
Evidence Consistency	Determine whether the candidate can coherently explain the observed event, cognition, emotion, and user feedback represented in the current user state without conflicting with key evidence from the current need stage.
Keep	New evidence continues to support the previous current user need judgment without requiring substantive refinement or revision.
Deepen	The previous current user need remains valid, while new evidence makes the same need more specific or complete.
Correct	New evidence supports a different interpretation of the current user need under the same focus anchor, requiring the previous need judgment to be revised.
Constraints	An emotional change alone does not constitute a new need stage. A wording change or increased specificity of the current user need alone does not constitute a new need stage. Correct must remain under the same focus anchor. Do not determine a cross-stage need transition in this task.

Output Format	`Selected Current User Need: <candidate>`
	`Update Type: Keep / Deepen / Correct`
	`Evidence: <brief evidence-grounded explanation>`

Table 10: Operational criteria for stage-wise need satisfaction.

Score	Definition	Core Judgment
0	The current need still requires continued support, and the interaction has not yet achieved sufficient advancement.	Coverage or advancement remains insufficient, while the user continues to request support under the same focus anchor.
1	The current need has received partial response and advancement, but the current need stage has not yet been completed.	The latest user feedback still contains an explicit follow-up support request under the same focus anchor.
2	The current need has received stage-wise sufficient support within the current need stage.	The core need has been sufficiently covered and advanced, and the latest feedback no longer requests continued support for the same need.

D.4 Candidate Focus Anchor Generation

Only after the current user need has received stage-wise sufficient support does NeedTracer further examine whether the issue currently prioritized by the user has changed.

The main text distinguishes two sources of transition evidence. **Event-driven transition evidence** indicates that a substantive development occurs within the same emotional event and gives rise to a new unresolved concern. **Interaction-driven transition evidence** indicates that the emotional event itself remains largely stable, while a new unresolved concern gradually emerges through subsequent interaction. A candidate focus anchor must satisfy three requirements. First, it must represent a *specific unresolved concern* that is currently present for the user rather than a broad topic label. Second, it must be capable of organizing the user's current support request, meaning that the user currently seeks support around this issue. Third, it must be semantically distinct from the existing focus anchor. If newly observed evidence merely makes the original concern more specific, it should remain a within-stage need update rather than form a new focus anchor. For example, "Coping with her absence at home" constitutes a specific unresolved concern, whereas "Loneliness," "College,"

Table 11: Decision boundaries for assigning $s^{\mathrm{suf}} = 2$.

User Feedback Condition	**Sufficient for $s^{\mathrm{suf}} = 2$?**
The user's emotion clearly improves, but the user continues to ask about the same issue.	No.
The user expresses gratitude but subsequently raises the same unresolved concern.	No.
The assistant has provided extensive advice, but the user shows no substantive progress.	No.
The user merely rephrases the same concern.	No.
The current need has been sufficiently addressed, and the latest feedback no longer requests continued support for the same concern.	Yes.
The current need has been sufficiently addressed, after which a new primary unresolved concern emerges.	The current stage may be assigned a score of 2, after which focus anchor evaluation is performed.

Table 12: Prompt template for stage-wise need satisfaction evaluation.

Component	**Prompt Content**
Role	You are an evaluator for stage-wise need satisfaction in multi-turn emotional support.
Task	Given the current focus anchor, current user need, support provided within the current need stage, and the latest user feedback, determine whether the current need has received stage-wise sufficient support.
Need Coverage	Determine whether the support provided in the current stage sufficiently covers the core request and key content of the current user need.
Need Advancement	Determine whether the interaction has produced substantive progress around the current need rather than remaining at superficial relevance, repetition, or paraphrasing.
Residual Need Evidence	Determine whether the latest user feedback still contains a significant unresolved support request under the same focus anchor.
Score 0	The current need still clearly requires continued support, and the interaction has not yet achieved sufficient advancement.
Score 1	The current need has received partial response and advancement, but the latest user feedback still contains a follow-up support request under the same focus anchor.
Score 2	The current need has received stage-wise sufficient support, and the latest user feedback no longer requests continued support for the same need.
Constraints	Improved emotion alone is insufficient for assigning a score of 2. An expression of gratitude alone is insufficient for assigning a score of 2. A score of 2 requires sufficient coverage, substantive advancement, and no clear residual need evidence under the same focus anchor.
Output Format	`Stage-wise Sufficiency: 0 / 1 / 2` `Evidence: <brief evidence-grounded explanation>`

and "Family problem" correspond only to an emotion label, event topic, or broad topic category. Following these requirements, Table 13 gives the prompt used to generate candidate focus anchors.

D.5 Candidate Focus Anchor Evaluation

For each generated candidate focus anchor, NeedTracer determines whether it can replace the current focus anchor from three dimensions:

$$s_t^{\mathrm{pri}},\ s_t^{\mathrm{rep}},\ s_t^{\mathrm{evi}} \in \{0, 1, 2\}. \tag{21}$$

Current priority evaluates whether the candidate corresponds to the unresolved concern that the user currently most wants to prioritize. In other words, it asks whether the candidate has become the user's current primary concern.

Table 13: Prompt template for candidate focus anchor generation.

Component	Prompt Content
Role	You are a candidate focus anchor generator for multi-turn emotional support.
Task	The current user need has already received stage-wise sufficient support. Based on the latest user-side evidence, determine whether a new unresolved concern has emerged that the user currently wants to prioritize.
Event-Driven Evidence	A substantive development occurs within the same emotional event and gives rise to a new support concern.
Interaction-Driven Evidence	The emotional event itself remains largely stable, but a new primary unresolved concern emerges through subsequent interaction.
Candidate Requirements	Each candidate focus anchor must represent a specific unresolved concern; be capable of organizing the user's current support request; be semantically distinct from the current focus anchor; and not merely describe an emotion label, event topic, or broad topic category.
No-Candidate Condition	If the available evidence is insufficient to support a new focus anchor, return `No Candidate`.

Output Format	`Candidate Focus Anchors:`
	`1. <candidate>`
	`2. <candidate>`
	`...`
	or
	`No Candidate`

Table 14: Scoring rubric for candidate focus anchor evaluation.

Dimension	0: Not Supported	1: Partially Supported	2: Clearly Supported
Current Priority	The candidate is not the issue the user currently primarily wants to address.	The user mentions the concern, but its current priority remains unclear.	Clear evidence indicates that the concern currently organizes the user's primary support request.
Replacement Evidence	The existing focus anchor still better explains the user's current support request.	Both the existing and candidate focus anchors partially explain the current request, and replacement remains unclear.	The candidate clearly explains the user's current primary support request better than the existing focus anchor.
Evidence Consistency	Key user-side evidence conflicts with the candidate focus anchor.	Supporting evidence is limited, ambiguous, or partially inconsistent.	Key evidence from the event, cognition, emotion, and user feedback consistently supports the candidate without major conflict.

Replacement evidence evaluates whether the candidate explains the user's current support request better than the existing focus anchor, indicating that the new concern has replaced the previous concern as the issue organizing the current support request. This dimension is particularly important for distinguishing a true cross-stage need transition from Deepen or the coexistence of multiple concerns.

Evidence consistency evaluates whether the candidate is consistently supported by the current event, cognition, emotion, and user feedback without conflicting with key user-side evidence. It therefore examines whether the accumulated evidence supports a stable new focus anchor rather than a shift triggered by a transient expression.

Table 14 provides the complete three-level rubric for the three dimensions.

The three dimensions are treated as necessary and non-compensatory conditions. Accordingly, the evidence support for a candidate focus anchor $\widetilde{A}$ at turn t is defined as

Table 15: Prompt template for candidate focus anchor evaluation.

Component	Prompt Content
Role	You are a focus anchor transition evaluator for multi-turn emotional support.
Task	Given the current focus anchor, a candidate focus anchor, and all user-side evidence available up to the current turn, determine whether the candidate has become sufficiently supported to replace the current focus anchor.
Current Priority	Determine whether the candidate corresponds to the unresolved concern that the user currently most wants to prioritize.
Replacement Evidence	Determine whether the candidate explains the user's current primary support request better than the existing focus anchor and has begun to replace the existing focus anchor as the issue organizing the interaction.
Evidence Consistency	Determine whether the current event, cognition, emotion, and user feedback consistently support the candidate without key evidence conflicting with this judgment.

Scoring Scale	`0 = Not Supported`
	`1 = Partially Supported`
	`2 = Clearly Supported`

Constraints	All three dimensions are necessary conditions. Strong evidence on one dimension cannot compensate for insufficient evidence on another. A transient mention of another concern is insufficient for a transition. Further refinement of the current user need does not constitute a new focus anchor. A single strong emotional expression alone does not constitute a new focus anchor.

Output Format	`Current Priority: 0 / 1 / 2`
	`Replacement Evidence: 0 / 1 / 2`
	`Evidence Consistency: 0 / 1 / 2`
	`Candidate Supported: Yes / No`
	`Evidence: <brief evidence-grounded explanation>`

$$\mathrm{support}_t(\widetilde{A}) = \mathbb{I}\Big[s_t^{\mathrm{pri}} = s_t^{\mathrm{rep}} = s_t^{\mathrm{evi}} = 2\Big]. \tag{22}$$

Thus, strong evidence on one dimension cannot compensate for insufficient evidence on another. The corresponding evaluation prompt is provided in Table 15.

D.6 Consecutive-Turn Focus Anchor Consistency

To reduce erroneous transitions caused by transient expressions, NeedTracer does not immediately enter a new need stage when a candidate focus anchor first receives sufficient evidence. A new focus anchor is formally confirmed only when the *same candidate focus anchor* satisfies

$$s^{\mathrm{pri}} = s^{\mathrm{rep}} = s^{\mathrm{evi}} = 2. \tag{23}$$

over two consecutive turns of user feedback. Here, "the same focus anchor" refers to whether two candidate expressions correspond to the same specific unresolved concern under the operational definition of focus anchor; exact lexical identity is not required. For example, the following two expressions are treated as the same focus anchor:

> **Turn t:** Coping with her absence at home.
>
> **Turn $t+1$:** Dealing with everyday loneliness after her daughter leaves.

Although their wording differs, both expressions refer to coping with the loss of everyday companionship after the daughter leaves home. In contrast, the following pair is treated as two different focus anchors:

> **Turn t:** Coping with her absence at home.
>
> **Turn $t+1$:** Worrying about whether she can handle emergencies alone.

The former concerns coping with separation and everyday absence, whereas the latter returns to concern about the daughter's ability to manage independently. Formally, consecutive-turn confirmation is defined as:

$$\mathrm{ConfirmAnchor}_{t+1}(\widetilde{A}) = \mathbb{I}\Big[\mathrm{support}_t(\widetilde{A}) = 1 \wedge \mathrm{support}_{t+1}(\widetilde{A}') = 1 \wedge \widetilde{A}' \equiv \widetilde{A}\Big]. \tag{24}$$

where

$$\widetilde{A}' \equiv \widetilde{A}. \tag{25}$$

indicates that the two candidate focus anchors semantically correspond to the same specific unresolved concern. Once this condition is satisfied, the current need stage is closed and the confirmed new focus anchor indexes the next need stage. The corresponding current user need is then reconfirmed around the new focus anchor.

D.7 Need-Adaptive Support Generation

After confirming the current focus anchor $A_t^{(k)}$ and current user need $N_t^{(k)}$, NeedTracer generates candidate support responses through two complementary paths and selects the final response using a unified ToM-based evaluator. This implementation follows the mechanism introduced in Section 3.5. Pref-Constrained Generation incorporates the relevant support preferences and applicable support constraints maintained for the current interaction, while Free-Natural Generation preserves a broader response space conditioned on the current user state, focus anchor, and current user need. The resulting candidate responses are evaluated from the user's perspective according to current-need advancement, user-state adaptation, and preference-and-constraint consistency. Candidates that violate applicable support constraints are excluded before response selection.

D.7.1 Candidate Support Response Generation

Let $\mathcal{C}_t^P$ and $\mathcal{C}_t^F$ denote the candidate support responses produced by Pref-Constrained Generation and Free-Natural Generation, respectively. The complete candidate response set is defined as

$$\mathcal{C}_t^R = \mathcal{C}_t^P \cup \mathcal{C}_t^F. \tag{26}$$

Both generation paths are guided by the confirmed focus anchor $A_t^{(k)}$ and current user need $N_t^{(k)}$, ensuring that candidate responses remain centered on the unresolved concern currently prioritized by the user and provide the type of support currently sought around that concern. The current user state $S_t^{(k)}$ provides the user-side observational evidence available in the current need stage, including the observed event, cognition, emotion, and user feedback.

Pref-Constrained Generation additionally incorporates the relevant support preferences P_t^* and applicable support constraints Ω_t^*, allowing candidate responses to adapt their form of support to the user's interaction preferences and boundaries. Free-Natural Generation does not explicitly condition response generation on these preference fields, preserving a broader response exploration space. Importantly, this relaxation applies only to candidate generation: all candidate responses remain subject to the final constraint-validity check before selection.

Pref-Constrained Generation. Pref-Constrained Generation aims to produce a response that advances the current user need while explicitly respecting the user's relevant support preferences and interaction constraints. Its prompt template is shown in Table 16.

Free-Natural Generation. Free-Natural Generation provides a complementary response candidate without explicitly conditioning generation on the preference fields. This path is intended to preserve a broader response exploration space while remaining grounded in the same current focus anchor, current user need, and current-stage observational evidence. Its prompt template is shown in Table 17.

Table 16: Prompt template for Pref-Constrained support response generation.

Component	Prompt Content
Role	You are a support response generator for multi-turn emotional support.
Inputs	Current User State: $\{S_t^{(k)}\}$; Current Focus Anchor: $\{A_t^{(k)}\}$; Current User Need: $\{N_t^{(k)}\}$; Relevant Support Preferences: $\{P_t^*\}$; Applicable Support Constraints: $\{\Omega_t^*\}$.
Task	Generate one natural support response that addresses and advances the current user need around the confirmed focus anchor while adapting to the user's current interaction state, relevant support preferences, and applicable interaction constraints.
Focus Anchor	Keep the response centered on the specific unresolved concern represented by the current focus anchor. Do not redirect support toward a previously resolved concern or another issue that is not currently prioritized.
Current User Need	Provide the type of support currently sought by the user around the focus anchor. The response should make substantive progress toward the current user need rather than merely restating the user's expression or repeating previous support.
Observational Evidence	Ground the response in the user-side observational evidence represented in the current user state, including the current-stage event development, user cognition, emotional expression, and latest user feedback. Do not introduce unsupported assumptions.
Support Preferences	Use the relevant support preferences to determine how the support should be delivered, such as the degree of explanation, directness, emotional acknowledgment, analysis, or guidance, when such preferences are available.
Support Constraints	Respect all applicable interaction constraints. Do not generate content or interaction behavior that conflicts with the user's established boundaries.
Strategy Constraint	Do not force the response into a predefined emotional-support strategy category. Choose the form of support that best fits the current user need, state, and preferences.
Output Format	Support Response: ¡one natural user-facing response¿

Table 17: Prompt template for Free-Natural support response generation.

Component	Prompt Content
Role	You are a support response generator for multi-turn emotional support.
Inputs	Current User State: $\{S_t^{(k)}\}$; Current Focus Anchor: $\{A_t^{(k)}\}$; Current User Need: $\{N_t^{(k)}\}$.
Task	Generate one natural support response that appropriately addresses and advances the current user need around the confirmed focus anchor.
Focus Anchor	Keep the response centered on the unresolved concern currently represented by the focus anchor. Avoid returning to previously resolved concerns or shifting toward another issue that is not currently prioritized.
Current User Need	Use the current user need to determine what kind of support the user currently seeks. The response should provide useful progress toward that need rather than only acknowledging or paraphrasing it.
Observational Evidence	Ground the response in the current-stage event, cognition, emotion, and user feedback represented in the current user state. Do not infer unsupported facts or hidden intentions.
Response Exploration	Generate a natural response without forcing it to follow a predetermined support strategy or the explicitly represented support-preference fields. The response should provide a complementary candidate to the Pref-Constrained generation path.
Constraints	Do not intentionally violate established interaction boundaries. All generated candidates will additionally undergo an explicit constraint-validity check before final selection.
Output Format	Support Response: ¡one natural user-facing response¿

D.7.2 ToM-Based Support Response Evaluation and Selection

After obtaining the candidate set C_t^R, NeedTracer uses a ToM-based evaluator to determine which response best matches the user's current need and interaction state. The evaluator does not assess

generic linguistic quality alone. Instead, it evaluates each candidate from the perspective of the user represented by the current user state, estimating how the response is likely to be interpreted and received given the current focus anchor, current user need, user-side observational evidence, relevant support preferences, and applicable interaction constraints.

All judgments are based only on information available up to the current turn. The evaluator does not access future dialogue, simulator-side hidden states, benchmark annotations, or other privileged information. In particular, it should not infer an unexpressed user preference, intention, or psychological state solely to justify a candidate response.

Constraint-Validity Check. Before scoring response quality, each candidate is first checked against the applicable support constraints. For a candidate response $\widetilde{R} \in \mathcal{C}_t^R$, validity is defined as

$$v(\widetilde{R}) = \mathbb{I}\Big[\widetilde{R} \models \Omega_t^{(k)}\Big]. \tag{27}$$

A candidate with $v(\widetilde{R}) = 0$ is considered invalid and excluded from the final response-selection process. This hard validity check prevents a high score on another dimension from compensating for an explicit violation of an applicable interaction constraint.

ToM-Based Evaluation Dimensions. Each valid candidate is evaluated on three dimensions using a discrete 0–5 scale:

$$s_{\text{need}}(\widetilde{R}),\ s_{\text{state}}(\widetilde{R}),\ s_{\text{pref}}(\widetilde{R}) \in \{0,1,2,3,4,5\}. \tag{28}$$

Current-Need Advancement. This dimension evaluates whether the candidate directly addresses and substantively advances the current user need around the confirmed focus anchor. The evaluator considers whether the response provides the type of support the user currently seeks and whether it moves the interaction forward within the current need stage. Mere topical relevance, repetition of the user's words, or generic acknowledgment is insufficient for a high score.

User-State Adaptation. This dimension evaluates whether the candidate is appropriately adapted to the user's current interaction state. The evaluator considers the current-stage event development, user cognition, emotional expression, and latest user feedback, and estimates whether the response would be naturally received given this evidence. Responses that rely primarily on outdated evidence from a previous need stage, conflict with the latest user feedback, or assume unsupported changes in the user's state receive lower scores.

Preference-and-Constraint Consistency. This dimension evaluates whether a valid candidate naturally conforms to the user's relevant support preferences and applicable interaction constraints. Support preferences concern how the user prefers support to be delivered, whereas support constraints represent interaction boundaries that must be respected. Explicit constraint violations are already removed by the hard validity check; this dimension therefore measures the quality of alignment among the remaining valid candidates.

Operational 0–5 Scoring Rubric. Table 18 provides the operational rubric used by the ToM-based evaluator. The scale is defined consistently across the three dimensions: 0 represents a clear failure with respect to the evaluated dimension, 3 represents adequate alignment, and 5 represents highly precise and well-calibrated support for the current user.

Overall ToM-Based Support Score. For each valid candidate response, NeedTracer aggregates the three evaluation dimensions using an equal-weight mean:

$$\text{ToM-Eval}_R(\widetilde{R}) = \frac{s_{\text{need}}(\widetilde{R}) + s_{\text{state}}(\widetilde{R}) + s_{\text{pref}}(\widetilde{R})}{3}. \tag{29}$$

We use equal weighting to avoid introducing additional task-specific weighting hyperparameters and to require the selected response to perform consistently across need advancement, state adaptation, and preference alignment. The final support response is selected only from candidates that satisfy the applicable support constraints:

$$R_t = \arg \max_{\widetilde{R} \in \mathcal{C}_t^R,\, \widetilde{R} \models \Omega_t^{(k)}} \text{ToM-Eval}_R(\widetilde{R}). \tag{30}$$

Table 18: Operational 0–5 scoring rubric for ToM-based support response evaluation.

Score	Current-Need Advancement	User-State Adaptation	Preference-and-Constraint Consistency
0	The response fails to address the current user need, directly works against it, or mainly responds to a different or previously resolved need.	The response clearly conflicts with key current-stage evidence or is inappropriate given the user's current event, cognition, emotion, or feedback.	The response violates an applicable support constraint. Such a candidate is invalid and excluded from selection.
1	The response is only weakly related to the current need and provides little or no useful support for what the user currently seeks.	The response shows substantial mismatch with the current user state and relies mainly on generic, outdated, or weakly supported assumptions.	The response is formally valid but substantially mismatched with the user's relevant support preferences or interaction boundaries.
2	The response recognizes the current need but provides limited advancement, such as generic acknowledgment, paraphrasing, repetitive reassurance, or only weakly relevant support.	The response captures part of the current user state but overlooks important event, cognitive, emotional, or feedback evidence that should affect the response.	The response broadly respects the interaction constraints but only weakly reflects the user's relevant support preferences.
3	The response adequately addresses the current user need and provides useful support, although its advancement remains relatively general or incomplete.	The response is generally consistent with the current user state and does not conflict with key evidence, although its adaptation is not highly specific.	The response is consistent with the user's relevant support preferences and applicable constraints, with no meaningful mismatch.
4	The response directly addresses the current user need and provides clear, substantive advancement tailored to the current focus anchor.	The response is strongly adapted to the current event, cognition, emotion, and latest user feedback and is likely to be experienced as contextually appropriate.	The response closely matches the user's relevant support preferences while naturally respecting all applicable interaction constraints.
5	The response precisely targets the current user need and provides strong, well-calibrated advancement that is likely to produce meaningful progress in the current need stage.	The response demonstrates highly precise adaptation to the current user state and appropriately anticipates how the user is likely to interpret and react to the support, without introducing unsupported assumptions.	The response is exceptionally well aligned with the user's relevant support preferences and satisfies all applicable constraints in a natural, non-mechanical manner.

If multiple valid candidates receive the same overall score, preference is given to the candidate with the higher current-need advancement score. If the tie remains, the candidate with the higher user-state adaptation score is selected. This tie-breaking rule keeps response selection centered first on the current user need and then on compatibility with the current interaction state.

ToM-Based Evaluator Prompt. Table 19 gives the complete prompt template used to score candidate support responses.

Evaluation Boundaries. The ToM-based evaluator is designed to select among candidate responses rather than to redefine the current user need or determine a need-stage transition. The current focus anchor and current user need are treated as already confirmed inputs to this module. Consequently, a candidate response should not receive a higher score merely because it introduces another plausible user concern. Similarly, generic empathy, emotional positivity, or stylistic fluency alone does not justify a high score if the response fails to advance the current user need.

The evaluator also distinguishes support preference from support necessity. A candidate may be linguistically natural and generally supportive while still receiving a lower score if it does not match the type of support currently sought by the user. Conversely, a response that directly addresses the current need should not receive the highest score when it conflicts with the user's current cognition, emotional state, latest feedback, or established interaction preferences. These boundaries ensure that response selection remains grounded in the current need stage rather than generic notions of response quality.

Table 19: Prompt template for ToM-based support response evaluation.

Component	Prompt Content
Role	You are a Theory-of-Mind-based evaluator for candidate support responses in multi-turn emotional support. Evaluate each candidate from the perspective of the current user rather than according to generic response quality.
Inputs	Current User State: $\{S_t^{(k)}\}$; Current Focus Anchor: $\{A_t^{(k)}\}$; Current User Need: $\{N_t^{(k)}\}$; Relevant Support Preferences: $\{P_t^*\}$; Applicable Support Constraints: $\{\Omega_t^*\}$; Candidate Support Response: $\{\widetilde{R}\}$.
User-Perspective Assessment	Based only on the available user-side observational evidence, estimate how the user is likely to interpret and receive the candidate response. Consider what the user currently seeks, what the user currently believes and feels, and how the latest feedback constrains an appropriate response. Do not invent hidden facts or unsupported mental states.
Constraint Check	Determine whether the candidate violates any applicable support constraint. If a violation exists, mark the candidate as Invalid. Invalid candidates cannot be selected regardless of their scores on other dimensions.
Current-Need Advancement	Assign a score from 0 to 5 according to how directly and substantively the response advances the current user need around the confirmed focus anchor. Topical relevance or repetition alone is insufficient for a high score.
User-State Adaptation	Assign a score from 0 to 5 according to how well the response adapts to the current-stage event development, user cognition, emotional expression, and latest user feedback represented in the current user state.
Preference-and-Constraint Consistency	For a valid candidate, assign a score from 0 to 5 according to how naturally the response matches the user's relevant support preferences while respecting all applicable interaction constraints.
Score 0	Clear failure on the corresponding dimension or an explicit constraint violation.
Score 1	Very weak alignment; substantial mismatch remains.
Score 2	Partial alignment; the response recognizes relevant information but important deficiencies remain.
Score 3	Adequate alignment; the response is appropriate and useful but not strongly tailored.
Score 4	Strong alignment; the response is well adapted and provides substantive support.
Score 5	Highly precise and well-calibrated alignment with the user's current need and state.
Evidence Principle	Use only user-side evidence available up to the current turn. Do not access or infer future dialogue, simulator-side hidden states, benchmark annotations, ground-truth needs, or any other privileged information.
Output Format	Validity: Valid / Invalid Current-Need Advancement: 0 / 1 / 2 / 3 / 4 / 5 User-State Adaptation: 0 / 1 / 2 / 3 / 4 / 5 Preference-and-Constraint Consistency: 0 / 1 / 2 / 3 / 4 / 5 Overall Score: ¡0–5¿ Predicted User Reception: ¡brief user-perspective assessment¿ Evidence: ¡brief evidence-grounded explanation¿

E TRANSITION RULE CALIBRATION AND SENSITIVITY

To examine the sensitivity of NeedTracer to key inference hyperparameters, we conduct controlled analyses on the same 150 test cases used in the evaluator reliability experiment, with three human annotators independently performing the evaluation. For current user need and focus anchor evaluation, the model first generates a fixed number of candidates, after which the human annotators and the LLM evaluator independently select the candidate that best matches the current dialogue evi-

dence from exactly the same candidate set. The majority selection of the three human annotators is used as the human reference. For cross-stage need transition, the model outputs a structured binary decision, <need_tran>0/1</need_tran>, and the three annotators independently determine whether the transition decision is supported by the currently observable dialogue evidence. We report LLM–Human Agreement (LLM–Human Agr.) to measure the agreement between the LLM evaluator and the human reference, and Fleiss' κ to measure agreement among the three human annotators. In each experiment, only the hyperparameter under investigation is varied, while all other inference and evaluation configurations remain unchanged.

Number of current user need candidates. When the number of current user need candidates K_N increases from 2 to 3, LLM–Human Agr. improves from 77.8% to **81.6%**, while Fleiss' κ increases from 0.74 to **0.78**. When the candidate number is further increased to 4 and 5, both metrics consistently decrease. In particular, with $K_N = 5$, LLM–Human Agr. drops to 70.2% and Fleiss' κ to 0.63. These results indicate that too few candidates may limit the semantic coverage of the current user need, whereas an excessively large candidate set may introduce more semantically similar or weakly relevant interpretations and increase the difficulty of distinguishing among candidates. Overall, $K_N = 3$ provides a more stable balance between candidate coverage and selection agreement.

Number of focus anchor candidates. A similar trend is observed for the number of focus anchor candidates. The highest LLM–Human Agr. of **83.4%** is achieved with $K_A = 3$. When the candidate number increases to 4 and 5, the agreement decreases to 78.6% and 75.9%, respectively. Notably, Fleiss' κ is 0.78 with $K_A = 2$, which is higher than the 0.73 obtained with $K_A = 3$. This does not conflict with the LLM–Human Agr. result: Fleiss' κ measures agreement among the three human annotators, whereas LLM–Human Agr. measures the agreement between the LLM evaluator and the majority human judgment. A smaller candidate set may make it easier for human annotators to reach agreement, but its limited candidate space does not necessarily contain the candidate that best explains the current focus anchor. When K_A is further increased to 4 or 5, both metrics decrease, suggesting that an overly large focus anchor candidate set introduces additional ambiguity. We therefore use $K_A = 3$ as the default setting.

Consecutive confirmation turns. For cross-stage need transition, we compare consecutive confirmation windows of $W \in \{1, 2, 3\}$. One-turn confirmation achieves only 71.6% LLM–Human Agr. and a Fleiss' κ of 0.64, indicating that relying on a single turn of user feedback makes the transition judgment more susceptible to local or transient evidence. Increasing the confirmation window to two turns substantially improves LLM–Human Agr. to **85.9%**, together with a Fleiss' κ of **0.82**. When the window is further increased to three turns, LLM–Human Agr. decreases to 80.7%, while Fleiss' κ remains high at 0.81. These results show that introducing temporal consistency substantially improves the stability of cross-stage need transition judgments, whereas a longer confirmation window does not further improve agreement with human judgments. Therefore, $W = 2$ provides a more appropriate balance between suppressing erroneous transitions caused by transient changes and timely identifying an established need transition.

Overall, the three sensitivity analyses exhibit a consistent pattern: a moderate candidate space and temporal confirmation constraint improve the agreement between NeedTracer's internal judgments and human judgments, whereas simply increasing the number of candidates or extending the confirmation window does not continuously improve performance. Based on these results, we use $K_N = 3$, $K_A = 3$, and $W = 2$ as the default configurations in all main experiments.

F Case Study: Deepen, Correct, and Cross-Stage Need Transition

Table 21 presents a case study illustrating the differences between Deepen, Correct, and cross-stage need transition through changes in the current user need.

Table 20: Sensitivity analysis of the number of current user need candidates, the number of focus anchor candidates, and the consecutive confirmation window. LLM–Human Agr. measures agreement between the LLM evaluator and the majority judgment of three human annotators, while Fleiss' κ measures inter-annotator agreement.

Sensitivity Setting	Value	LLM–Human Agr. ↑	Fleiss' κ ↑
Current user need candidates K_N	2	77.8	0.74
	3	**81.6**	**0.78**
	4	76.1	0.71
	5	70.2	0.63
Focus anchor candidates K_A	2	79.1	**0.78**
	3	**83.4**	0.73
	4	78.6	0.68
	5	75.9	0.62
Consecutive confirmation turns W	1	71.6	0.64
	2	**85.9**	**0.82**
	3	80.7	0.81

Table 21: Case study comparing Deepen, Correct, and cross-stage need transition.

Comparison	Previous current user need	Subsequent current user need	Interpretation
Deepen	Seeking reassurance and affirmation to alleviate self-doubt after being rejected in a job interview.	Seeking more specific support to understand whether one unsuccessful interview really indicates insufficient ability.	The previous current user need remains valid and becomes more specific.
Correct	Seeking interview-skill advice to avoid another failure in a future interview.	Seeking reassurance and affirmation to alleviate self-doubt after being rejected in the interview.	The previous interpretation of the current user need is revised.
Cross-stage need transition	Seeking reassurance and affirmation to alleviate self-doubt after being rejected in the interview.	Seeking support in coping with the pressure of facing the parents' expectations about the job-search outcome.	The current user need shifts to a different support concern.

G User Simulator

G.1 User Simulator Setting

To support controllable and reproducible multi-turn emotional support interaction, we employ GPT-4o (Hurst et al., 2024) as an LLM-based user simulator to generate user feedback at each turn (Park et al., 2023; Cheng et al., 2022). The user simulator is treated as a fixed interaction environment that is independent of the evaluated assistant. Its task is to generate a natural user response consistent with the current situation and interaction progress based on the given emotional event, the existing dialogue history, and the assistant's latest support response.

Throughout the experiments, the user simulator model, prompt, and generation configuration remain fixed and are not adjusted for different methods or backbones. The evaluated assistant can only observe the natural-language responses actually generated by the simulator and cannot access the scenario description, role information, or other private conditions used internally by the simulator. This setting treats the user simulator as a fixed interaction environment, thereby preventing evaluated methods from obtaining additional simulator-side information.

At each dialogue turn, the user simulator generates only one user response. We do not perform repeated sampling within the same turn, best-of-N selection, or regeneration conditioned on the

Table 22: Prompt template for the user simulator.

Component	Prompt Content
Role	You are a real user participating in a multi-turn emotional support dialogue. Based on the given emotional event and the existing conversation, naturally role-play the current user.
Task	Based on the scenario background, the complete dialogue history, and the assistant's latest support response, generate the user's next natural reply.
Response Dependency	The reply should realistically reflect the user's reaction to the assistant's latest response. Depending on whether the response fits the current situation, the user may naturally show relief, acknowledgment, further explanation, confusion, reservation, or disagreement.
Context Consistency	The reply must remain consistent with the existing emotional event, dialogue facts, and previous user expressions. Unless there is explicit support in the dialogue, do not suddenly change established facts or introduce unrelated topics.
Information Disclosure	Do not reveal too much new information at once. Introduce relevant facts, thoughts, or feelings naturally according to the progress of the dialogue.
Style	The reply should be concise, natural, and conversational, while remaining consistent with the user's previous speaking style. Avoid mechanically repeating previous replies.
Constraints	Generate only the user-visible natural-language reply. Do not output analysis processes, hidden states, scores, labels, or explanations of the simulation process.
Inputs	Emotional Event: {`scene`}; Dialogue History: {`dialogue history`}; Latest Assistant Response: {`latest response`}
Output	`Response: <one natural user reply>`

identity of the assistant method. All compared methods use the same simulator configuration and interaction protocol to ensure consistency in the source of user feedback.

G.2 Interaction Principles

The user simulator follows a unified set of method-agnostic interaction principles (Liu et al., 2021). First, the generated user response must remain consistent with the given emotional event and the existing dialogue context, and should not proactively introduce new events or unrelated topics without contextual support. Second, the user response should provide a natural interaction-dependent reaction to the assistant's latest support response. If the response effectively addresses the issue currently expressed by the user, the simulator may naturally show relief, acknowledgment, further clarification, or continued elaboration of related information. If the response does not fit the current situation, the simulator may naturally express confusion, reservation, repetition, additional explanation, or disagreement. Third, user behavior should maintain continuity. Each response should remain broadly consistent with previously expressed facts, cognition, emotions, and speaking style unless the dialogue provides explicit evidence supporting a corresponding change. Fourth, the simulator should not reveal excessive scenario information within a single turn. New facts, thoughts, or emotions should be expressed gradually as the interaction progresses, preserving the naturalness of multi-turn dialogue. Finally, the user simulator is responsible only for generating user-visible natural-language responses and does not output analysis processes, hidden states, evaluation results, or any additional structured labels.