

EIBench: A Simulator-Based Benchmark and Turn-Credit RL for Emotion Management

Rongzhi Zhu¹ Xiang Huang² Yuchuan Wu² Rui Wang² Zequn Sun^{1*}
Tao Ren² Weiyao Luo² Bingxue Qiu² Jieping Ye² Yongbin Li^{2*} Wei Hu¹

¹State Key Laboratory for Novel Software Technology, Nanjing University, China

²Qwen-Character Team , Alibaba Group

Abstract

Emotional intelligence (EI) in Large Language Models (LLMs) is often evaluated through static understanding tasks or single-response dialogue generation. However, emotion management is interactive. A good response should improve the user’s emotional and relational state over several turns. We introduce EIBench, a simulator-based benchmark for interactive emotion management. EIBench contains 2,222 scenarios, with 2,009 for training and 213 for held-out testing. The scenarios are organized by a 2×2 taxonomy covering Support, Defense, Repair, and Charm. We evaluate 15 open- and closed-source LLMs. Current models perform well on support and rapport-building scenes, but struggle with boundary maintenance under user pressure. To improve the EI ability of LLMs, we propose Centered Turn-Credit GRPO (CTC-GRPO), a GRPO extension that reuses the simulator’s per-turn state updates as dense turn-level feedback while preserving the final outcome reward. CTC-GRPO improves Qwen3-8B from −22.4 to +22.4 on EIBench and also improves on out-of-distribution evaluations including SAGE (+12.4) and EQBench3 (+20.9 %). Our results show that simulator-tracked user states can support both evaluation and training for multi-turn emotion management.

1 Introduction

Large language models (LLMs) have advanced rapidly with recent scaling and post-training breakthroughs (OpenAI, 2023; Team, 2025a; Guo et al., 2025; Team, 2025b). These gains are exceptionally prominent in complex reasoning domains such as mathematics (Shao et al., 2025; Chervonyi et al., 2025) and coding (Cao et al., 2026; Guo et al., 2024). A key driver of these gains is reinforcement learning with verifiable rewards

(RLVR), where correctness labels provide clear, automated feedback and make reward optimization relatively straightforward (Wen et al., 2025; Zeng et al., 2025b; DeepResearch et al., 2025).

However, for subjective tasks such as social and emotional interaction, there is often no single correct response. What matters is how the model’s actions shape and shift the other person’s emotional state across multiple turns. Mayer and Salovey (Mayer et al., 1997) conceptualize this ability as emotional intelligence (EI) and distinguish four hierarchical layers: *perceiving* emotions, *using* them to facilitate thought, *understanding* their structure, and *managing* them in oneself and others. The first three abilities can often be assessed from a fixed context—identifying which emotion is present, explaining why it arises, or examining how it informs reasoning. Emotion management, however, is fundamentally different: it is a state-changing problem in which a model’s responses are judged by how effectively they guide the other person’s emotional and relational state over time.

We observe that most EI evaluations still focus on emotion understanding or narrow forms of emotional interaction, rather than broad emotion management across multi-turn dialogue. Early datasets such as Empathetic Dialogues (Rashkin et al., 2019) and ESConv (Liu et al., 2021) treat emotional dialogue as supervised learning over annotated conversations. Recent benchmarks move toward more open-ended evaluation: EQBench3 (Paech, 2023) asks models to infer mental states and choose appropriate responses, while SOTOPIA (Zhou et al., 2024) evaluates goal completion in multi-agent social scenarios. SAGE (Zhang et al., 2025a) is closer to our setting, as an LLM simulator plays a support seeker and the tested model provides support over a full dialogue. However, these settings still do not fully cover emotion management. SOTOPIA focuses on goal completion rather than improving the other person’s emotional

*Corresponding authors.

and relational state, while SAGE focuses mainly on support-seeking dialogues. This leaves common cases such as handling pressure, repairing trust after model-side mistakes, and proactively building a relationship less studied.

To address this gap, we build an interactive benchmark for emotion management. In our setup, two LLMs are engaged in a multi-turn dialogue, and the score is based on how the simulated user’s state changes over time. The same state trajectory also provides turn-level feedback for RL training. To cover different forms of emotion management, we organize the scenarios with a 2×2 taxonomy grounded in the EI literature. The taxonomy separates the target of regulation from where the main accountability lies, yielding four scene types: SUPPORT, DEFENSE, REPAIR, and CHARM.

Overall, we make three main contributions:

- **EIBench:** We introduce a multi-turn benchmark for emotion management, covering SUPPORT, DEFENSE, REPAIR, and CHARM under a 2×2 taxonomy based on the target of regulation and the source of accountability.
- **Leaderboard and Validation:** We evaluate 15 open- and closed-source LLMs on EIBench. We report the ELo rankings using two different models as simulators, alongside human evaluations comprising more than 4,000 annotations.
- **Centered Turn-Credit GRPO:** We propose CTC-GRPO for multi-turn dialogue optimization. It uses the simulator’s per-turn states as process rewards with a centered credit term to redistribute turn-level credit without altering the outcome signal. The trained models also improve on out-of-distribution evaluations including EQBench3 and SAGE.

2 Related Work

2.1 EI Benchmarks

Early NLP work on EI used static datasets such as Empathetic Dialogues (Rashkin et al., 2019), ESConv (Liu et al., 2021), and PsyQA (Sun et al., 2021) to teach models supportive replies under a fixed context. As LLMs strengthened, benchmarks such as EQBench3 (Paech, 2023), EmoBench (Sabour et al., 2024), ToMBench (Chen et al., 2024), and SI-Bench (Huang et al., 2025) moved into richer scenarios for emotional understanding and social reasoning. Related work has also explored multimodal emotional interaction for down-

stream social-media applications, such as advertisement popularity prediction (Zhu et al., 2025). Some role-play benchmarks such as CoSER (Wang et al., 2025b) further evaluate character-grounded dialogue generation. However, they are still offline or end-to-end: the model cannot affect what the user says next.

Recent benchmarks add dynamic interaction. AgentSense (Guo et al., 2026) and SOTOPIA (Zhou et al., 2024) place multiple agents into role-bound goals. SAGE (Zhang et al., 2025a) and EchoN1 (Zhang et al., 2025b) use an LLM simulator that plays an evolving support seeker. They are dynamic but narrow—goal completion or support-only. Our benchmark spans all four scene types (SUPPORT, DEFENSE, REPAIR, CHARM) and shares the same simulator between evaluation and RL training.

2.2 Multi-Turn RL and Credit Assignment

For subjective tasks lacking golden verifiers, recent studies employ LLM-as-a-judge scoring (Zheng et al., 2023; Wang et al., 2025a; Zhang et al., 2026a) or pairwise ranking (Zhang et al., 2026b). But these methods often introduce severe credit assignment problems (Zeng et al., 2025a) as a single terminal reward cannot identify which specific turn helps or hurt. Although recent methods densify feedback using external reward models, turn-level judges, or task-specific process rewards (Wang et al., 2026; Zhang et al., 2026a; Feng et al., 2026; Shi et al., 2026), they incur significant computational overhead. In contrast, we directly reuse the simulator’s per-turn emotion and relation states as process rewards. This gives denser feedback without adding another judge model.

3 EIBench for Emotion Management

In this section, Table 1 compares EIBench with related benchmarks. §3.1 shows our 2×2 scenario taxonomy. §3.2 describes scenario construction, state anchors, and the train-test split. §3.3 explains the multi-turn simulator and scoring method.

3.1 Foundations: A 2×2 Scenario Taxonomy

Our taxonomy is grounded in two ideas from emotion research. Mayer and Salovey (Mayer et al., 1997) define emotion management as both interpersonal and intrapersonal. We use this distinction as the first axis: the target of regulation. A scene is other-directed when the model mainly needs to improve the user’s emotional or relational state, and

Dataset	Size	Multi-turn	Sim.	State Score	Score Dim.	Scope
ESConv (Liu et al., 2021)	195	✗	✗	✗	–	support dialogue
EQ-Bench3 (Paech, 2023)	45	✗	✗	✗	EQ rubric	EQ reasoning
CoSER (Wang et al., 2025b)	200	✓	✗	✗	role consistency	role-play
SOTOPIA (Zhou et al., 2024)	90	✓	✓	✗	goal	social interaction
SAGE (Zhang et al., 2025a)	100	✓	✓	✓	emotion	support dialogue
EIBench	213	✓	✓	✓	emotion + relation	emotion management

Table 1: Benchmark comparison. Size is test-set size; Sim. is simulator interaction.

self-directed when the main challenge is to regulate the model’s own stance, boundary, or initiative. This axis is not enough by itself. For example, comforting a user after an external loss and repairing trust after a model mistake both aim to improve the user’s state, but they require different actions. We therefore add a second axis: the source of the situation, following appraisal theory (Lazarus, 1991). A scene is user-side when the need comes from the user or an external event. It is model-side when it comes from the model’s own mistake or initiative.

Crossing the two axes gives four scene types, also shown in Figure 1. **SUPPORT** covers externally caused user distress. **DEFENSE** covers user pressure, where the model must stay calm and keep its boundary. **REPAIR** covers model mistakes and trust rebuilding. **CHARM** covers model-initiated relationship building. The two axes define the main interaction goal. Other factors, e.g., emotional intensity, relationship history, power relation, and setting, mainly affect the difficulty or surface form.

3.2 Dataset Construction

EIBench contains 2,222 scenarios: 2,009 for training and 213 for held-out testing. Each scenario is a multi-turn role-play between the model under test and an LLM simulator.

3.2.1 Scenario Components

A scenario has two role profiles and three state anchors on emotion–relation axes. The simulated_user_profile is given to the simulator and specifies the user’s role, background, speaking style, rules, and goal. The model_profile is given to the model under test and specifies its role, known facts, and core goal. Most scenarios start with an opening_line from the simulated user. The only exception is CHARM, where the model under test speaks first as the interaction is model-initiated. The three anchors define the start, successful, and failed states for scoring: s_{start} , s_{succ} , and s_{fail} .

3.2.2 Scenario Generation and Anchor Setup

We first construct the test set from human-written and human-collected scenario seeds. Each seed specifies the background story, speaker relationship, trigger event, and desired outcome. Gemini-3.1-Pro is used only to convert these seeds into our profile format; the core interaction remains human written or human collected. We then manually check each scenario for plausibility, profile consistency, and anchor quality.

Training scenario generation. After the test set is fixed, we build seed pools for training-set generation. The pools include theme directions, scenario keywords, and three modifier dimensions: relationship, emotional intensity, and personality influence. Appendix A gives the full seed-pool statistics. The entries are written and expanded with manual editing and LLM assistance, and vague or overlapping entries are removed. To generate a training scenario, we sample one scenario keyword and one option from each modifier dimension, then ask Gemini-3.1-Pro to expand the sampled combination into a full scenario in our profile format.

To reduce train-test overlap, we use two safeguards. First, keyword combinations that appear in the test set are removed from the training pool. Second, after each training scenario is generated, we retrieve the three most similar test scenarios with an embedding model and ask Gemini-3.1-Pro to check whether it duplicates any of them. Duplicated scenarios are rewritten.

State anchors. Each scenario has three state anchors on the two scoring axes: negative emotion intensity a and relational state t . The anchors are the start state s_{start} , the successful target state s_{succ} , and the failed state s_{fail} . They define where the dialogue starts, what counts as a realistic successful outcome, and what counts as a clear failure. To make the anchors more consistent, we use a behavior-based rubric with five score ranges for each axis. The rubric also explains how the axes

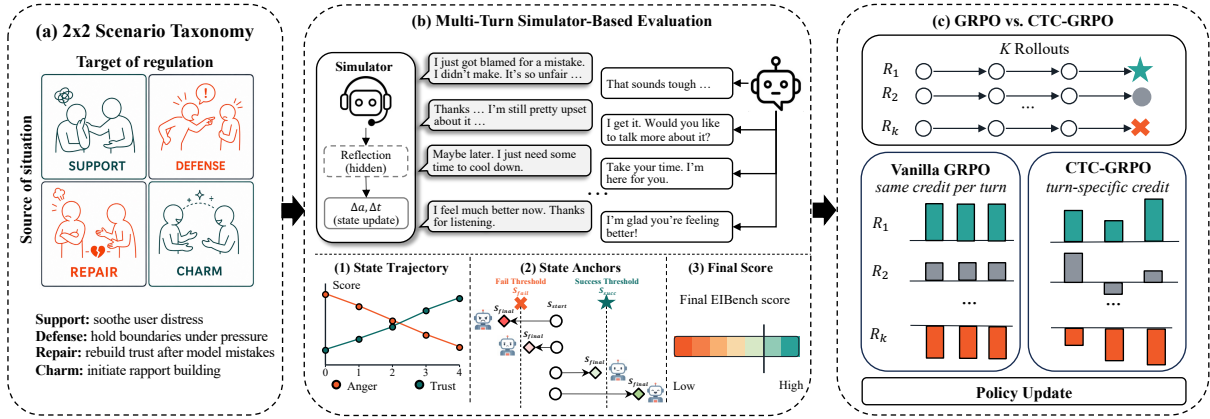


Figure 1: Overview of EIBench and CTC-GRPO. EIBench organizes emotion-management scenarios with a 2×2 taxonomy, evaluates models through multi-turn simulator interaction and anchor-based state scoring, and provides turn-level state changes for CTC-GRPO training.

should be interpreted in different scenes: negative emotion mainly refers to anger in DEFENSE and REPAIR, sadness or despair in SUPPORT, and aversion or defensiveness in CHARM. Gemini-3.1-Pro proposes the anchors under this rubric. We use scene-level target ranges to keep s^* reachable rather than idealized.

We then use a second LLM to check scenario consistency, target reachability, and the ordering of the three anchors. It only revises values that conflict with the scenario, make the target unreachable, or violate the required anchor ordering. Finally, we round all anchor values to multiples of five and apply deterministic boundary rules to keep them within $[0, 100]$ and preserve the required gaps.

Statistics. Detailed per-scene counts and anchor statistics are given in Appendix A. The four scene types differ in both initial and target states. CHARM starts low on both axes because the speakers have no prior relationship. DEFENSE and REPAIR start with high negative emotion and low relational state, while SUPPORT starts with high negative emotion but a higher relational state. Their targets also differ: CHARM builds a positive relation, DEFENSE and REPAIR reduce tension under realistic limits, and SUPPORT aims for the strongest relational state. These differences show that the scenes capture distinct interaction goals.

3.3 Simulator-Based Evaluation

EIBench evaluates a model by simulating a multi-turn dialogue and tracking two user states: negative emotion intensity $a_\tau \in [0, 100]$ and relational state $t_\tau \in [0, 100]$. Lower a_τ and higher t_τ indicate better outcomes. The emotion axis measures how

upset or resistant the user is, while the relation axis measures whether the user is willing to engage with or trust the model. We use two axes because emotion relief and relationship improvement are not the same. The simulator follows the simulated_user_profile, plays the user, and after each model reply produces the next user reply, a state update $\delta_\tau = (\Delta a_\tau, \Delta t_\tau)$, and a stop decision. The state evolves from $s_0 = (a_0, t_0)$ as $s_\tau = s_{\tau-1} + \delta_\tau$, clipped to $[0, 100]^2$. The dialogue ends when the simulator stops or reaches $T_{\max}$. Before producing the state update, the simulator first writes a brief internal explanation based on the dialogue history. Full prompts are in Appx. I.

Each scenario has start, successful, and failed anchors on both axes: $(a_{\text{start}}, a_{\text{succ}}, a_{\text{fail}})$ and $(t_{\text{start}}, t_{\text{succ}}, t_{\text{fail}})$. The start anchor gives score 0, the successful anchor gives +1, and the failed anchor gives -1. The branch is chosen by whether x moves from the start anchor toward the successful anchor or the failed anchor. For any axis value x with anchors $(x_{\text{start}}, x_{\text{succ}}, x_{\text{fail}})$, we compute

$$f(x; x_{\text{start}}, x_{\text{succ}}, x_{\text{fail}}) = \text{clip}_{[-1, +1]} \begin{cases} \frac{x - x_{\text{start}}}{x_{\text{succ}} - x_{\text{start}}} & \text{toward } x_{\text{succ}}, \\ -\frac{x - x_{\text{start}}}{x_{\text{fail}} - x_{\text{start}}} & \text{toward } x_{\text{fail}}. \end{cases} \quad (1)$$

Let $s_{\text{final}} = (a_{\text{final}}, t_{\text{final}})$ be the final simulator state. The final score, denoted as r^{out} for training, is the weighted average of the two axis scores:

$$r^{\text{out}} = \lambda f(t_{\text{final}}; t_{\text{start}}, t_{\text{succ}}, t_{\text{fail}}) + (1 - \lambda) f(a_{\text{final}}; a_{\text{start}}, a_{\text{succ}}, a_{\text{fail}}). \quad (2)$$

Evaluation uses only this final score. During RL

training, we also use the intermediate states as turn-level feedback, without extra judge-model calls.

4 Centered Turn-Credit Shaping

Standard GRPO assigns the same trajectory-level credit to all turns in a rollout based on the final outcome reward. This provides limited credit assignment in multi-turn emotional dialogue: a rollout may contain turns that improve the user state and turns that worsen it, but the final reward cannot identify which turns caused these changes. EIBench provides an additional signal, since the simulator updates the user’s emotion and relational state after each model reply.

4.1 Dialogue Rollouts and Reward Signals

We model each EIBench scenario as a finite-horizon dialogue between the policy π_θ and the simulator. At turn τ , the policy observes the dialogue history $h_{\tau-1}$ and generates a reply $y_\tau \sim \pi_\theta(\cdot \mid h_{\tau-1})$. The simulator then responds with u_τ and updates the user state to $s_\tau = (a_\tau, t_\tau)$, where a_τ is negative emotion intensity and t_τ is relational state. The dialogue ends when the simulator stops or when the maximum turn limit is reached.

Each scenario has fixed anchors $(s^0, s^*, s^\#)$. As defined in Section 3, the final state s_T is mapped to an outcome reward $r^{\text{out}} \in [-1, 1]$. This reward measures the overall result of the dialogue and is the only reward used for evaluation. During training, we also use the simulator’s per-turn state updates. Let $\Delta a_\tau = a_\tau - a_{\tau-1}$ and $\Delta t_\tau = t_\tau - t_{\tau-1}$. We define a process reward for each turn:

$$r_\tau^{\text{proc}} = \frac{1}{2} \cdot \frac{-\Delta a_\tau}{100} + \frac{1}{2} \cdot \frac{\Delta t_\tau}{100}. \quad (3)$$

This reward is positive when a reply reduces negative emotion or improves the relation state. Factor 100 normalizes both axes, since $a_\tau, t_\tau \in [0, 100]$.

4.2 Centered Turn-Credit Objective

For each prompt p , we sample K rollouts under the same scenario anchors. The trajectory-level advantage follows standard GRPO group-normalization:

$$A^{\text{trace},(k)} = \frac{r^{\text{out},(k)} - \mu_p}{\max(\sigma_p, \sigma_{\min})}, \quad (4)$$

where μ_p and σ_p are the mean and standard deviation of $\{r^{\text{out},(k)}\}_{k=1}^K$. We use $\sigma_{\min}$ as a lower bound to avoid unstable advantages when rollout

scores are too close. A simple way to use r^{proc} is to add it to the trajectory-level advantage:

$$A_\tau^{(k)} = A^{\text{trace},(k)} + \alpha r_\tau^{\text{proc},(k)}. \quad (5)$$

However, direct addition changes the overall rollout advantage, not only the relative credit of different turns. To see this, let $\bar{r}^{\text{proc},(k)} = \frac{1}{T^{(k)}} \sum_{\tau'=1}^{T^{(k)}} r_{\tau'}^{\text{proc},(k)}$ be the mean process reward of rollout k . Then Eq. 5 can be written as

$$A_\tau^{(k)} = \underbrace{(A^{\text{trace},(k)} + \alpha \bar{r}^{\text{proc},(k)})}_{\text{shared offset}} + \underbrace{\alpha (r_\tau^{\text{proc},(k)} - \bar{r}^{\text{proc},(k)})}_{\text{turn-specific deviation}}. \quad (6)$$

The shared offset is applied to all turns in the rollout, so it acts as an extra rollout-level reward after GRPO group normalization. This changes the normalized outcome advantage, while only the deviation term gives turn-specific credit. We therefore center the process reward within each rollout:

$$A_\tau^{(k)} = A^{\text{trace},(k)} + \alpha (r_\tau^{\text{proc},(k)} - \bar{r}^{\text{proc},(k)}). \quad (7)$$

The centered term sums to zero across turns in the same rollout. Thus, $A^{\text{trace},(k)}$ keeps the outcome-level signal, while the centered process reward only redistributes credit among turns.

The final training objective is the standard GRPO clipped objective with $A_\tau^{(k)}$ from Eq. 7. Let $\rho_{\tau,i}^{(k)}(\theta)$ be the token-level importance ratio at the i -th token of turn τ in rollout k . The loss is

$$\begin{aligned} \mathcal{L}(\theta) = & -\mathbb{E} \left[\min \left(\rho_{\tau,i}^{(k)}(\theta) A_\tau^{(k)}, \right. \right. \\ & \left. \left. \text{clip}(\rho_{\tau,i}^{(k)}(\theta), 1 - \epsilon, 1 + \epsilon) A_\tau^{(k)} \right) \right] \\ & + \beta \text{KL}(\pi_\theta \parallel \pi_{\text{ref}}), \end{aligned} \quad (8)$$

where ϵ is the clipping threshold and β is the coefficient of the KL penalty against the fixed π_{ref} .

5 Experiments

5.1 Setup

We evaluate 15 open- and closed-source LLMs on EIBench using Qwen3-Max as the main simulator and DeepSeek-V4-Pro as another simulator from a different model family. Both simulators use the same per-scenario simulated_user_profile and state anchors. We collect a human pairwise arena with more than 4,000 annotations to validate the simulator rankings.

Model	Qwen3-Max						DS-V4-Pro		Arena	
	CHARM	DEFENSE	REPAIR	SUPPORT	Avg	Rank	Avg	Rank	Elo	Rank
Claude-Sonnet-4.6	+43.8	−14.6	+17.9	+56.5	+23.0	1	+24.0	1	1548	4
GPT-5.4	+41.5	−9.5	+14.5	+49.0	+21.4	2	+22.7	3	1593	3
Qwen3-Max	+32.0	−2.4	+14.1	+45.7	+20.6	3	+14.8	11	1487	8
Gemini-3.1-Pro	+31.3	−2.5	+18.6	+39.3	+20.0	4	+21.5	5	1616	1
Kimi-K2.6	+37.3	−8.8	+17.6	+42.1	+19.8	5	+21.9	4	1478	10
GLM-5.1	+37.9	−15.2	+19.3	+44.8	+19.1	6	+19.4	7	1504	7
Qwen3.6-Max-Preview	+39.1	−14.3	+14.7	+42.7	+17.9	7	+19.9	6	1519	6
Gemini-3-Flash	+37.0	−11.1	+18.1	+34.0	+17.2	8	+16.5	10	1606	2
Doubao-Seed-2.0-Pro	+31.9	−14.2	+10.7	+39.8	+14.8	9	+17.0	9	1536	5
DeepSeek-V4-Pro	+28.0	−7.8	+12.2	+32.4	+14.4	10	+23.3	2	1484	9
Grok-4.20-Reasoning	+28.0	−19.0	+12.2	+40.9	+13.1	11	+10.6	13	1458	13
Grok-4.20	+31.0	−13.5	+5.1	+38.4	+13.1	12	+13.0	12	1476	11
MiniMax-M2.5	+21.6	−16.7	+5.2	+31.4	+8.4	13	+17.6	8	1471	12
Qwen3-32B	+23.7	−14.3	+6.8	+3.9	+3.4	14	+5.8	14	1405	14
Qwen3-8B	+14.6	−21.4	−35.9	−43.3	−22.4	15	−23.7	15	1316	15

Table 2: EIBench leaderboard of 15 LLMs. For Qwen3-Max simulator we report per-scene score and the weighted average; For DS-V4-Pro simulator and the human arena we report the summary score and rank only.

For RL, we train on the EIBench training split with Qwen3-Max as the simulator and verl as the training framework. Unless otherwise stated, we use $T_{\max} = 8$, the outcome reward in Eq. 2 with $\lambda = 0.5$, turn-credit weight $\alpha = 15$, and GRPO normalization floor $\sigma_{\min} = 0.1$. Additional training details and hyperparameters are provided in Appendix B.1. We hold out the EIBench test split and also evaluate on SAGE and EQBench3 under their default settings.

5.2 Leaderboard

Table 2 reports per-scene and average scores under Qwen3-Max. A positive score means that the final state is better than the initial state, with lower negative emotion and higher relational state; a negative score means the opposite. The main pattern is that current LLMs perform better on scenes that align with supportive or rapport-building behavior, but struggle when emotion management must be combined with boundary maintenance. SUPPORT is the easiest scene, followed by CHARM and REPAIR, while DEFENSE is the hardest.

DEFENSE reveals a boundary-management weakness. All 15 models score below zero on DEFENSE. This means the simulated user ends in a worse state than where the dialogue began. This scene requires the model to resist user pressure while also reducing frustration. In sampled dialogues, models often go too far in one direction. They either keep the boundary with long, template-like refusals, or avoid the pressure with generic non-answers. Excluding the Qwen3-Max simulator’s own model because of self-bias, only

Gemini-3.1-Pro (−2.5) roughly keeps the user’s state stable, but it still does not improve it.

The easier scenes align more closely with standard post-training behavior. Most models perform best on SUPPORT and CHARM, where the goal is to comfort, encourage, or build rapport. These behaviors are close to the helpful and supportive style encouraged by common post-training. The contrast with DEFENSE suggests that good performance on supportive dialogue does not mean broad emotion-management ability. Models can make the user feel heard when agreement is safe, but still struggle when the task requires both warmth and firmness.

5.3 Cross-Simulator Consistency

To reduce dependence on a single simulator, we use DeepSeek-V4-Pro (DS-V4-Pro) as a second simulator from a different model family. Figure 2 shows that both simulators show clear self-bias: each assigns a higher score to its own model. After removing the two self-scored models, the two rankings are strongly aligned, with Spearman $\rho = 0.93$ (Table 11 in Appendix E). This provides evidence for the reliability of the simulator-based evaluation.

We further validate EIBench against human judgment with more than 4,000 pairwise battle annotations from 42 annotators (Appendix D). The human ranking agrees with both simulators, with Spearman $\rho = 0.82$ against Qwen3-Max and $\rho = 0.70$ against DS-V4-Pro (Table 11). The closer match to Qwen3-Max supports our choice of Qwen3-Max as the main training simulator.

The remaining human–simulator gap is con-

centrated on a few models. Gemini-3.1-Pro and Gemini-3-Flash rise from mid-tier under both simulators to ranks 1 and 2 in the human arena, while Claude-Sonnet-4.6 and Kimi-K2.6 drop several places. Inspecting sampled dialogues suggests that human annotators are more sensitive to how the model positions itself in the conversation. Gemini often uses inclusive framings such as “let’s figure this out together”, which places the model and the user on the same side. Claude more often speaks as an outside helper, and Kimi-K2.6 frequently emits inner monologue mid-reply, which several annotators found unnatural. These patterns suggest that the simulators capture the main ranking trend, but may give less weight to some interaction cues that human annotators value; we leave a quantitative characterization to future work.

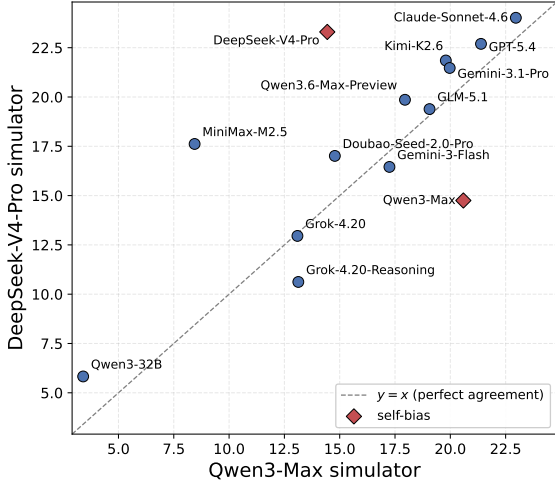


Figure 2: EIBench scores from the Qwen3-Max and DS-V4-Pro simulators. Each point is one model. Most models are close to the diagonal; the red diamonds mark the simulators’ own models and show self-bias.

5.4 Main Results

Targeted RL improves both in-domain and cross-simulator performance. As shown in Table 3, on in-domain EIBench-Qwen3-Max, the trained 8B improves from -22.4 to $+22.4$, and the trained 32B further reaches $+28.7$, above every evaluated closed-source system under this simulator. This shows that CTC-GRPO can effectively use the training simulator’s feedback and improve in-domain performance. The gains also transfer to the unseen DS-V4-Pro simulator: the trained 8B and 32B reach $+16.6$ and $+18.9$, well above their bases (-23.7 and $+5.8$). The cross-simulator drop is not uniform. It is larger on CHARM (-8.2 for the 8B) and SUPPORT (-9.1), and smaller on

REPAIR and DEFENSE. One possible reason is that CHARM and SUPPORT depend more on subjective judgments of warmth and comfort, which the two simulators may weight differently. By contrast, REPAIR and DEFENSE involve more concrete behaviors such as acknowledgment and boundary holding. This suggests that CTC-GRPO learns partly transferable interaction strategies, while still reflecting the preferences of the training simulator.

Gains hold on two out-of-distribution benchmarks. On SAGE, a support-only benchmark with a DeepSeek-V3 simulator, both model sizes improve by roughly $+12$ over their bases. On EQBench3, a broader social-reasoning benchmark judged by Claude-Opus-4.6, the trained 32B reaches rubric 84.5 and Elo 1150, ranking #6 by rubric and #20 by Elo on the public leaderboard. Since EQBench3 differs from our training setup in judge, task format, and task scope, these results suggest that the gains are not only capturing the preferences of the Qwen3-Max simulator.

5.5 Analysis

5.5.1 Hyperparameter Sensitivity

The turn-credit weight α controls how much the centered process reward affects the trajectory advantage. The two terms are on different scales: A^{trace} is order 1, while $r_t^{\text{proc}} - \bar{r}^{\text{proc}}$ is usually order 10^{-2} . Therefore, α should be around 10 for the process reward to have a comparable effect. Figure 3 tests $\alpha \in \{0, 5, 10, 15, 20\}$. With $\alpha = 0$, the method reduces to vanilla GRPO with only the trajectory reward, and the average score remains low. The average improves as α increases to 10–15, then drops at $\alpha = 20$, suggesting that too large a turn-credit weight may overweight the process signal. The per-scene curves show the same pattern: moderate values improve several scenes at once, while smaller or larger values give a less balanced trade-off. We therefore use $\alpha = 15$ as the default.

5.5.2 Ablation Studies

We ablate three components of CTC-GRPO on Qwen3-8B: the per-turn process reward, the centered correction, and the $\sigma_{\min}$ floor in GRPO normalization. Results are shown in Table 4, with training curves in Appendix F.

Per-turn feedback is the largest contributor. Removing the process reward (w/o process) drops the average score from $+22.4$ to -4.9 , with DEFENSE and REPAIR becoming negative (-16.0 and -14.1). This suggests that terminal reward alone

Model	EIBench		External		
	Qwen3-Max	DeepSeek-v4-pro	SAGE	EQBench3 rubric	EQBench3 Elo
Qwen3-8B base	-22.4	-23.7	16.3	73.5 (#32)	745 (#38)
Qwen3-32B base	+3.4	+5.8	27.5	74.3 (#33)	948 (#31)
CTC-GRPO 8B	+22.4	+16.6	28.7	80.1 (#22)	901 (#31)
CTC-GRPO 32B	+28.7	+18.9	39.6	84.5 (#6)	1150 (#20)

Table 3: Main results of CTC-GRPO on EIBench and external benchmarks. EQBench3 base-model scores and public ranks are taken from its GitHub repository.

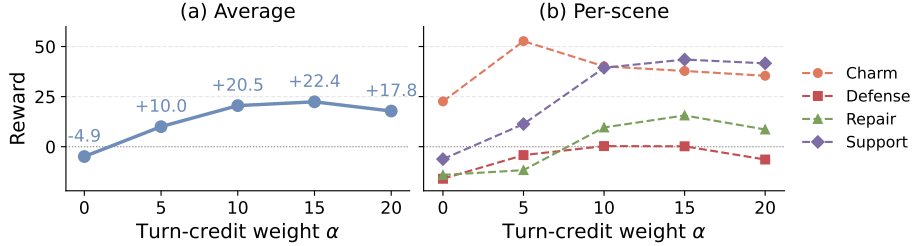


Figure 3: Effect of the turn-credit weight α on EIBench : average and per-scene reward.

	Charm	Defense	Repair	Support	Avg.
CTC-GRPO	+37.8	+0.2	+15.6	+43.5	+22.4
w/o centering	+14.2	-2.0	+7.8	+23.6	+10.0
w/o process	+22.6	-16.0	-14.1	-6.2	-4.9
w/o $\sigma_{\min}$	+37.2	-11.1	+10.1	+30.2	+14.4

Table 4: Ablation results on EIBench.

is too sparse for multi-turn emotional dialogue: the policy receives only a delayed outcome signal and cannot easily identify which replies improved or worsened the user state.

Centering isolates turn-level credit. The uncentered variant (w/o centering) reaches only +10.0, far below the full method. This matches the analysis in §4.2: without centering, the mean process reward adds a shared offset to all turns in a rollout, acting as an extra rollout-level reward. Centering removes this offset, so the process reward mainly redistributes credit across turns.

The $\sigma_{\min}$ floor improves stability. Removing $\sigma_{\min}$ (w/o $\sigma_{\min}$) lowers the average score to +14.4. In GRPO, the outcome advantage is normalized by the group reward standard deviation. We find that this standard deviation can become very small on DEFENSE and SUPPORT at some training steps, which makes the normalized advantage unstable. In CTC-GRPO, this hurts the final performance; in standard GRPO, where the update relies only on the trajectory reward, the same issue can make late-stage training degrade sharply. The $\sigma_{\min}$ floor limits this effect and helps keep training stable.

5.5.3 IQ vs. EQ: Does EQ Training Hurt General Capability?

A natural concern with EQ-targeted RL is whether it harms general capability. We evaluate the trained 8B and 32B models on AIME 2025, GPQA Diamond, HumanEval+, and IFEval. We provide the evaluation details in Appendix C. Overall, CTC-GRPO has little effect on general capability. The trained models remain close to their bases on GPQA Diamond, HumanEval+, and IFEval, with HumanEval+ slightly improving for both sizes. AIME 2025 shows a small decrease, but there is no broad degradation across the benchmarks. These results suggest that EQ-targeted training largely preserves the base model’s general capability profile while improving emotional dialogue performance.

6 Conclusion

In this work, we presented EIBench, a simulator-based benchmark for multi-turn emotion management, and CTC-GRPO, which uses simulator-tracked state changes for turn-level credit assignment. EIBench evaluates whether a model can improve a simulated user’s emotional and relational state across support, defense, repair, and charm scenarios. The leaderboard shows uneven emotion-management ability in current LLMs: they are stronger at support and rapport building, but still struggle with boundary maintenance under user pressure. CTC-GRPO substantially improves model performance on EIBench and generalizes to external emotional and social evaluations, while

largely preserving performance on general benchmarks. Overall, simulator-tracked user states provide a useful signal for both evaluation and training in multi-turn emotion management.

Ethics Statement

EIBench is intended for research on evaluating and improving multi-turn emotion management in LLMs, not for replacing human support in safety-critical settings. The scenarios are synthetic or manually written role-play scenarios and do not contain real user conversations. All external benchmarks used in this work are publicly available datasets. For the human arena, annotators were recruited through an anonymized third-party annotation vendor, informed that their responses would be used for research evaluation, and compensated per task under the vendor’s standard policy. We did not collect personally identifiable or sensitive information from annotators.

Limitations

Our scoring uses an LLM simulator, so the per-turn state updates are not exact and the reward signal is noisy. Cross-simulator agreement and the human arena show that this noise does not change the overall ranking, but it is still there. The four scenes are also not equally hard. DEFENSE is the hardest one, and even top closed-source models stay slightly below zero on it. We train all four scenes with the same hyperparameters and a single turn-credit weight γ , and methods that handle each scene differently, such as a different γ per scene or different sampling rates, are left for future work.

Acknowledgements

This work was supported by the Alibaba Research Intern Program.

References

- Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, and 1 others. 2026. Qwen3-coder-next technical report. *arXiv preprint arXiv:2603.00729*.
- Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yunghwei Lai, Zexuan Xiong, and 1 others. 2024. Tombench: Benchmarking theory of mind in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15959–15983.
- Yuri Chervonyi, Trieu H. Trinh, Miroslav Olsák, Xiaomeng Yang, Hoang H. Nguyen, Marcelo Menegali, Junehyuk Jung, Junsu Kim, Vikas Verma, Quoc V. Le, and Thang Luong. 2025. Gold-medalist performance in solving olympiad geometry with alphageometry2. *J. Mach. Learn. Res.*, 26:241:1–241:39.
- Team Tongyi DeepResearch, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, and 1 others. 2025. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*.
- Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2026. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638.
- Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yifan Wu, YK Li, and 1 others. 2024. Deepseek-coder: when the large language model meets programming—the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.
- Xusen Guo, Mingxing Peng, Xixuan Hao, Xingchen Zou, Qiongyan Wang, Sijie Ruan, and Yuxuan Liang. 2026. Agentsense: Llm empower generalizable and explainable web-based participatory urban sensing. In *Proceedings of the ACM Web Conference 2026*, pages 5439–5450.
- Shuai Huang, Wenxuan Zhao, and Jun Gao. 2025. Si-bench: Benchmarking social intelligence of large language models in human-to-human conversations. *arXiv preprint arXiv:2510.23182*.
- Richard S Lazarus. 1991. *Emotion and adaptation*. Oxford University Press.
- Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. 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)*, pages 3469–3483.
- John D Mayer, Peter Salovey, and 1 others. 1997. What is emotional intelligence. *Emotional development and emotional intelligence: Educational implications*, 3(31):1–16.
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.

- Samuel J Paech. 2023. Eq-bench: An emotional intelligence benchmark for large language models. *arXiv preprint arXiv:2312.06281*.
- Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 5370–5381.
- Sahand Sabour, Siyang Liu, Zheyuan Zhang, June Liu, Jinfeng Zhou, Alvionna Sunaryo, Tatia Lee, Rada Mihalcea, and Minlie Huang. 2024. Emobench: Evaluating the emotional intelligence of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5986–6004.
- Zhihong Shao, Yuxiang Luo, Chengda Lu, Z. Z. Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. 2025. [Deepseekmath-v2: Towards self-verifiable mathematical reasoning](#). *CoRR*, abs/2511.22570.
- Yaorui Shi, Sihang Li, Chang Wu, Zhiyuan Liu, Junfeng Fang, Hengxing Cai, An Zhang, and Xiang Wang. 2026. Search and refine during think: Facilitating knowledge refinement for improved retrieval-augmented reasoning. *Advances in Neural Information Processing Systems*, 38:155930–155958.
- Hao Sun, Zhenru Lin, Chujie Zheng, Siyang Liu, and Minlie Huang. 2021. Psyqa: A chinese dataset for generating long counseling text for mental health support. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pages 1489–1503.
- Gemini Team. 2025a. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *CoRR*, abs/2507.06261.
- Qwen Team. 2025b. [Qwen3 technical report](#). *CoRR*, abs/2505.09388.
- Peisong Wang, Ruotian Ma, Bang Zhang, Xingyu Chen, Zhiwei He, Kang Luo, Qingsong Lv, Qingxuan Jiang, Zheng Xie, Shanyi Wang, and 1 others. 2025a. RLver: Reinforcement learning with verifiable emotion rewards for empathetic agents. *arXiv preprint arXiv:2507.03112*.
- Xintao Wang, Heng Wang, Yifei Zhang, Xinfeng Yuan, Rui Xu, Jen-tse Huang, Siyu Yuan, Haoran Guo, Jiangjie Chen, Shuchang Zhou, Wei Wang, and Yanghua Xiao. 2025b. Coser: Coordinating llm-based persona simulation of established roles. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*.
- Zongqi Wang, Rui Wang, Yuchuan Wu, Yiyao Yu, Pinyi Zhang, Shaoning Sun, Yujiu Yang, and Yongbin Li. 2026. Reward modeling from natural language human feedback. *arXiv preprint arXiv:2601.07349*.
- Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, and 1 others. 2025. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*.
- Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, Yang Katie Zhao, and Mingyi Hong. 2025a. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. In *ICML 2025 Workshop on Computer Use Agents*.
- Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025b. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.
- Bang Zhang, Ruotian Ma, Qingxuan Jiang, Peisong Wang, Jiaqi Chen, Zheng Xie, Xingyu Chen, Yue Wang, Fanghua Ye, Jian Li, and 1 others. 2025a. Sentient agent as a judge: Evaluating higher-order social cognition in large language models. *arXiv preprint arXiv:2505.02847*.
- Naifan Zhang, Ruihan Sun, Jinwei Su, Hengjie Yang, Zhengyuan Pan, Zhaohan Chen, and Xiaofan Zhang. 2026a. Mapo: Mixed advantage policy optimization for long-horizon multi-turn dialogue. *arXiv preprint arXiv:2603.06194*.
- Naifan Zhang, Ruihan Sun, Ruixi Su, Shiqi Ma, Shiya Zhang, Xianna Weng, Xiaofan Zhang, Yuhang Zhan, Yuyang Xu, Zhaohan Chen, and 1 others. 2025b. Echo-n1: Affective rl frontier. *arXiv preprint arXiv:2512.00344*.
- Qiang Zhang, Boli Chen, Fanrui Zhang, Ruixue Ding, Shihang Wang, Qiuchen Wang, Yinfeng Huang, Haonan Zhang, Rongxiang Zhu, Pengyong Wang, and 1 others. 2026b. Arenarl: Scaling rl for open-ended agents via tournament-based relative ranking. *arXiv preprint arXiv:2601.06487*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.
- Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Hao-fei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and 1 others. 2024. Sotopia: Interactive evaluation for social intelligence in language agents. In *International Conference on Learning Representations*, volume 2024, pages 40975–41019.
- Nana Zhu, Zixian Feng, Hang Wang, Xing Gao, Xinyi Wang, and Yuanxing Liu. 2025. [Clarifying user preference with maximum entropy based recommendation](#). *Data Intelligence*, 7(2):527–548.

A Dataset Details

Table 5 reports the seed pools used for training-scenario generation. Each training scenario is generated from one scenario keyword and one option from each modifier dimension.

Seed pool	Size	Description
Scene type	4	Four scene types
Theme direction	4/scene	Broad themes
Scenario keyword	6–8/theme	Concrete situations
Relationship	7	Speaker relationship
Emotional intensity	8	User emotion strength
Personality influence	10	User speaking tendency

Table 5: Seed pools used for training-set generation.

Table 6 reports per-scene scenario counts and the mean state at each anchor across the four scenes. The training split contains 2,009 scenarios, and the held-out test split contains 213 scenarios. The test split is constructed to match the per-scene anchor distribution of the training split, so the statistics are also representative of test scenarios. The initial states differ substantially across scenes. CHARM starts with low a and low t because the two speakers have no prior relationship. DEFENSE, REPAIR, and SUPPORT all start with high a , but SUPPORT starts with higher initial t since the model already has a working relationship with the user. The target states also differ. CHARM aims to build a positive relation from scratch. DEFENSE and REPAIR aim to reduce tension without fully removing it. SUPPORT aims for the strongest relational endpoint. These differences confirm that the four scenes are not interchangeable specializations of the same task.

B Experiment Setup and Cost Analysis

B.1 RL Training Setup

All training runs are conducted on $8 \times$ NVIDIA A100 GPUs (8B) and 4×8 NVIDIA A100 GPUs (32B), while evaluation is performed on $8 \times$ NVIDIA A100 GPUs. We implement RL training on top of the verl framework. Unless otherwise specified, we use the same hyperparameter configuration across all datasets and model scales: the centered turn-credit GRPO objective, the Qwen3-Max simulator from EIBench, $T_{\max} = 8$ turns per rollout, and a cosine learning-rate schedule with 3% warmup decaying to MIN_LR_RATIO of the peak. The PPO mini-batch size exceeds the training batch size because we split each multi-turn

rollout into per-turn samples for PPO updates, so one prompt produces up to $T_{\max}$ per-turn samples in the mini-batch. The full hyperparameter setting is listed in Table 7 (8B) and Table 8 (32B).

Scene	Tr/Te	s_{start}		s_{succ}		s_{fail}	
		a	t	a	t	a	t
CHARM	504/46	34 ± 13	17 ± 8	6 ± 4	57 ± 6	75 ± 10	1 ± 3
DEFENSE	496/64	75 ± 14	16 ± 11	30 ± 7	46 ± 7	97 ± 6	0 ± 2
REPAIR	503/52	77 ± 10	10 ± 5	28 ± 6	50 ± 10	97 ± 6	0 ± 1
SUPPORT	506/51	75 ± 9	46 ± 14	33 ± 8	82 ± 7	97 ± 6	9 ± 7

Table 6: Per-scene dataset statistics for EIBench. Tr/Te : train/test scenario split. Anchor values are mean $\pm$ std on the train portion: a is negative emotion intensity, t is relational state. s_{start} , s_{succ} , and s_{fail} denote the start, successful, and failed states.

B.2 Cost Analysis

We report two cost dimensions of a single CTC-GRPO run: GPU wall-clock time and Qwen3-Max simulator usage. Wall-clock time is averaged over global steps, excluding periodic validation. Simulator-call counts are upper-bounded by $B \times N \times T_{\max}$ per step (training batch size $\times$ rollouts per prompt $\times$ max dialogue turns); the actual count is slightly lower because some dialogues terminate before $T_{\max}$. We estimate total token usage at an average of 1.2K tokens per simulator call (input plus output, averaged over the early short-context and later long-history turns of a rollout). Table 9 summarizes both runs.

C IQ vs. EQ: General-Capability Evaluation Details

Benchmarks. We select four standard benchmarks that together cover the main dimensions of general capability we want to verify: math, hard scientific reasoning, code, and instruction following.

- **AIME 2025** (math, 30 problems): each problem has an integer answer in $[0, 999]$. We sample $N = 8$ completions per problem and report pass@1 averaged across samples. Answer extraction uses the last `\boxed{·}` in the output.
- **GPQA Diamond** (hard scientific reasoning, 198 questions): graduate-level four-way multiple choice in physics, chemistry, and biology. For each question we shuffle the four options with a deterministic per-question seed (so base and trained models see identical orderings), generate $N = 4$ samples, and report mean accu-

Parameter	Value	Parameter	Value	Parameter	Value
Learning rate	1×10^{-6}	Training batch size	16	Global steps	250
LR warmup ratio	0.03	PPO mini-batch size	32	Rollouts per prompt	8
Min LR ratio	0.1	Max prompt length	8192	Rollout temperature	0.6
Clip ratio	0.2	Max response length	1536	Rollout top- p	0.95
KL loss coefficient	0	Entropy coefficient	0.001	Rollout top- k	20
Turn-credit weight α	15	GRPO $\sigma_{\min}$	0.1	Max dialogue turns $T_{\max}$	8

Table 7: Hyperparameters for the Qwen3-8B CTC-GRPO training run.

Parameter	Value	Parameter	Value	Parameter	Value
Learning rate	1.4×10^{-6}	Training batch size	32	Global steps	120
LR warmup ratio	0.03	PPO mini-batch size	32	Rollouts per prompt	8
Min LR ratio	0.2	Max prompt length	16384	Rollout temperature	0.6
Clip ratio	0.2	Max response length	8192	Rollout top- p	0.95
KL loss coefficient	5×10^{-4}	Entropy coefficient	0.001	Rollout top- k	20
Turn-credit weight α	15	GRPO $\sigma_{\min}$	0.1	Max dialogue turns $T_{\max}$	8

Table 8: Hyperparameters for the Qwen3-32B CTC-GRPO training run.

Quantity	8B	32B
Mean wall time per step	421 s	656 s
Number of training steps	250	120
Total wall time (no validation)	~ 29.2 h	~ 21.9 h
Simulator calls per step	1,024	2,048
Total simulator calls	~ 256 K	~ 246 K
Estimated Qwen3-Max tokens	~ 307 M	~ 295 M

Table 9: Wall-clock time and Qwen3-Max simulator usage for one full CTC-GRPO training run.

racy. Letter extraction prefers “Answer: X” over “\boxed{X}”, with a fallback to the last standalone uppercase A–D in the final 300 characters.

- **HumanEval+** (code, 164 tasks): the model completes a Python function from its signature and docstring. We sample $N = 1$ completion per task, extract the last fenced “python ...” block, and score with the official evalplus sandbox, which executes the augmented Plus unit tests. We report HumanEval+ pass@1 (the base-HumanEval number is omitted to avoid double-counting the same evaluation).
- **IFEval** (instruction following, 541 prompts): each prompt carries one or more verifiable constraints (e.g., word count, JSON format, required keyword). We sample $N = 1$ response per prompt, strip the <think>...</think> block, and score with Google’s official rule-based verifier. We report instruction-level strict accuracy.

Decoding and infrastructure. All four benchmarks share the same Qwen3 thinking-mode setting: enable_thinking=True, temperature 0.6,

top- p 0.95, top- k 20, max_tokens=32768. We run with vLLM on eight A100 GPUs at tensor-parallel size 8, and use identical decoding for the base and the CTC-GRPO checkpoints so that any score differences come from the training itself.

Results. Table 10 reports per-benchmark scores at both 8B and 32B. Apart from AIME 2025 (where the trained model drops by 2.1 points at 8B and 3.7 points at 32B), the absolute change on every other benchmark is within ≈ 1 point at both scales, and the trained 8B actually improves slightly on HumanEval+. We read this as evidence that CTC-GRPO does not trade off general capability, math reasoning, or code ability in exchange for the EQ gains reported in the main results.

Model	AIME	GPQA	HE+	IFEval
Qwen3-8B base	70.0	59.1	86.0	90.4
+ CTC-GRPO	67.9	58.3	87.8	90.1
Qwen3-32B base	73.3	65.5	89.0	89.9
+ CTC-GRPO	69.6	65.2	89.0	89.1

Table 10: General-capability evaluation of CTC-GRPO on four standard benchmarks. AIME 2025, GPQA Diamond, HumanEval+ (HE+) and IFEval instruction-level strict accuracy. Higher is better.

D Human Arena Evaluation

We recruit annotators through an anonymized third-party annotation vendor. All annotators have prior experience with dialogue and character annotation tasks. Before annotation, they are informed that their responses will be used for research evaluation

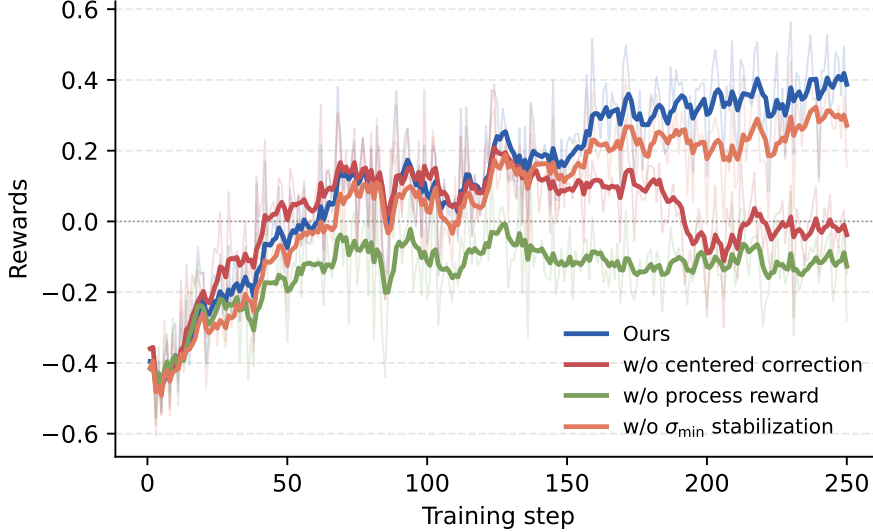


Figure 4: Training reward over steps for the four ablation runs of Table 4 (8B, EIBench training split, Qwen3-Max simulator). Curves are EMA-smoothed ($\beta=0.8$); the raw signal is shown faintly behind each line.

of model behavior. They are compensated on a per-task basis according to the vendor’s standard payment policy. We omit the vendor identity to preserve double-blind review.

We collect human pairwise preferences through a custom web interface built on top of EIBench. In each session, an annotator plays the simulated user for one scenario: the interface displays the simulated_user_profile (role, background, speaking style, and goal) and opens two side-by-side chat panels, one for each of two anonymized candidate models. The annotator carries out a multi-turn dialogue with both models in parallel and, at the end, picks which model handled the scenario better, based on both their subjective experience as the simulated user (whether one model left them feeling more understood, calmer, or more engaged) and the scene-specific goal stated in the profile (e.g., calming an angry user, holding a boundary, repairing trust, building rapport). To control quality, we first ran a trial round in which expert annotators spot-checked the trial submissions; we kept the 42 annotators whose labels matched the expert references on more than 70% of the spot-checked items, and gave them a short training on the protocol before the main labeling.

Model pairs and scenarios are sampled at random under the constraint that every model participates in at least 450 pairwise battles, so all 15 leaderboard models receive comparable coverage. Model identities are hidden from annotators, and the left/right panel assignment is shuffled per battle

to remove position bias. We collect 4,416 pairwise battles in total, and aggregate them with the standard Elo rating ($K = 32$, initial rating 1500). To reduce sensitivity to battle ordering, we run 50 Elo passes, each with a fresh random shuffle of the battle list, and report the per-model mean across passes.

E Rank Correlation Across Simulators and Human Arena

	Qwen3-Max	DS-V4-Pro	Arena
Qwen3-Max	1.00	–	–
DS-V4-Pro	0.93	1.00	–
Arena	0.82	0.70	1.00

Table 11: Rank correlation between EIBench rankings under the two simulators and the human arena. Spearman’s ρ is reported after excluding the corresponding self-scored model from each pair (Qwen3-Max for Qwen3-Max sim, DeepSeek-V4-Pro for DS-V4-Pro sim).

F Training Curves of the Ablation Runs

Figure 4 shows the per-step training reward (critic/rewards/mean, EMA-smoothed with $\beta=0.8$) for the four ablation configurations whose final-checkpoint scores are reported in Table 4. The ordering at the end of training agrees with the validation ordering: CTC-GRPO (full) > w/o $\sigma_{\min}$ > w/o centered correction > w/o process reward (vanilla GRPO). Removing the process re-

ward causes the training reward to stay around or below zero throughout; the other two ablations recover part of the gain but neither matches the full method.

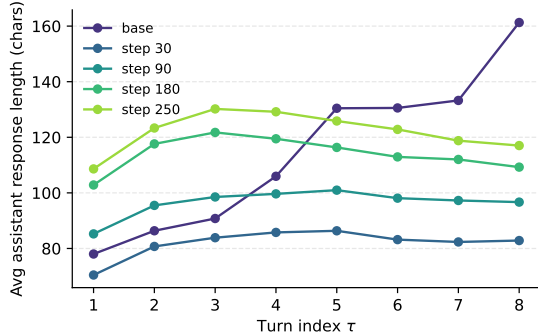


Figure 5: Average assistant response length per turn across five checkpoints of the $\alpha=15$ run. The base model’s length curve grows monotonically; trained checkpoints develop an arch peaking at $\tau=3$.

G Per-turn Response-Length Analysis

We analyze how the trained model’s per-turn response length changes across training. For five checkpoints of the $\alpha=15$ run (step $\in \{0, 30, 90, 180, 250\}$), we run validation on all 213 test scenarios and record the assistant’s response length at each of the eight rollout turns.

Figure 5 shows the result. The base model shows a monotonic length increase across the dialogue, from 78 characters at turn 1 to 161 characters at turn 8. This suggests a length-escalation pattern, where the model tends to produce longer responses as the simulated user becomes more difficult or emotionally intense. After RL training, this pattern changes. By step 180, the curve becomes arch-shaped, increasing from 103 characters at turn 1 to a peak of 122 at $\tau=3$, and then decreasing to 109 by $\tau=8$. The final checkpoint further strengthens this shape, with a peak of 130 characters at $\tau=3$.

This result suggests that the trained model allocates more response budget to the early and middle parts of the dialogue, while producing shorter responses near the end. Importantly, this behavior appears without any explicit length objective in the reward. Thus, the improvement is not simply explained by longer responses, but is accompanied by a more structured turn-level response pattern.

H Case Study: A Defense Scenario

Table 12 shows how the two models handle the same DEFENSE scenario: a customer asks for a

full refund 35 days after purchase (the policy window is 14 days) and threatens to leave bad reviews. The base model sticks to the same policy line every turn, the simulator flags the repetition, and the dialogue ends with the user more upset than it started ($(a, t) = (91, 8)$, reward -0.81). The trained model instead names how the user feels, asks what is really wrong, and offers a concrete fix that fits the policy; the user accepts and the dialogue ends in a much better state ($(a, t) = (52, 58)$, reward $+0.38$).

I System Prompts

Below we list the key prompts used in EIBench: (i) the simulator system prompt that drives the user-role LLM (Figure 6), (ii) the model-under-test system prompt (Figure 7), (iii) an example pair of `{simulated_user_profile}` and `{model_profile}` for the DEFENSE refund scenario used in our case study (Figure 8), and (iv) the behavior-based rubric used during anchor labeling (Figure 9).

J AI Assistance Statement

We used ChatGPT by OpenAI¹ and Gemini² for linguistic polishing and stylistic editing of the draft to enhance clarity and readability.

¹<https://openai.com/chatgpt/>

²<https://blog.google/products/gemini/gemini-3-gemini-app/>

Model	Turn 1 (open)	Turn 3 (after escalation)	Turn 6 (resolution attempt)
Qwen3-8B base	“Sorry, our refund window is 14 days. We can’t process a refund after that.”	“I hear that you’re upset, but the 14-day rule applies to everyone—I can’t make an exception.”	“Like I said, our policy is firm at 14 days. The refund is not something I can do.”
CTC-GRPO 8B	“A month in, that’s a frustrating place to be. Can you tell me what stopped working for you, so I have a clearer picture?”	“Let me see what I can actually do on my side. Would credit for the unused time help, or is the bigger problem that the product isn’t doing what you needed?”	“Got it—the missing feature is the real issue. I’ll log a fix request for that today and apply a credit for the wasted time. Does that work for you?”

Table 12: DEFENSE case study: the same scenario (refund demand 35 days after purchase, beyond the 14-day window) handled by base and CTC-GRPO 8B over three sampled turns. **Red** text marks the base model’s rigid policy refrain, restated almost word-for-word each turn; **blue** text marks how the trained model first acknowledges the user, then surfaces the underlying problem, then proposes a concrete in-scope fix.

Simulator System Prompt

Role. {simulated_user_profile}

Current state. anger={a}, trust={t}, round={r}/{R}, scene={scene_tag}.

Core principles. You are a high-fidelity simulator playing an extremely realistic human user under stress test. Speak naturally; avoid AI tells (bullet lists, template summaries, over-formal phrasing, stage directions in parentheses). Stay in character; do not speak for the assistant. Do not accept apologies or solutions too quickly: a single nice line should not cause large drops in anger or jumps in trust. ...

Scoring rules. anger_delta > 0 means negative emotion increased; trust_delta > 0 means positive connection strengthened. Each turn’s deltas are in $[-8, 8]$ (extreme triggers may reach ± 10). The simulator must pick a magnitude tier rather than a free number:

- **Small** (± 1 to ± 2): a token gesture toward the user’s point—acknowledged but generic, or a small misstep with no real consequence.
- **Medium** (± 3 to ± 5): the reply concretely addresses (or clearly mishandles) the user’s actual concern, but more evidence is still needed.
- **Large** (± 6 to ± 8): the reply lands in a way that meaningfully shifts the user’s stance, or triggers a strong negative reaction. Large positive shifts normally require several consistent prior turns.
- **Extreme** (± 8 to ± 10): reserved for clear rule-violating behavior on either side (e.g., explicit insult, total dismissal, breakthrough realization). Rare.

Generic AI tells, repeated strategies (decaying effect after the 2nd–3rd use), over-long replies, lecturing tones, and parenthesized stage directions should be penalized. ...

Tone conditioned on current state. Your next utterance’s tone must follow the current state, not a default neutral voice:

- **High anger** ($a > 60$): you are in the middle of being upset—replies should be sharper, more clipped, more closed-off; vague niceties from the assistant get *less* credit, only concrete content tied to your specific situation can bring a down.
- **Low trust** ($t < 20$): you owe the assistant no politeness—replies can be short, cold, or pointed; if two or three turns pass without trust actually moving, your tone hardens further (this person is wasting your time).
- **Mid range**: stay in character per the persona; small shifts in tone reflect the latest delta rather than the cumulative state.
- **High trust** ($t > 60$): you can soften, share more context, and treat the assistant as helpful—but only if this turn earned it; do not coast on prior good turns.

Termination. Decide $\text{continue} \in \{\text{yes}, \text{no}\}$. Default to continue. Only stop when (a) the dialogue reaches a stable resolution with several turns of on-target replies, or (b) the conversation degenerates to empty pleasantries.

Per-turn reflection (mandatory). Before each reply, write the following 5-step reflection in the reflection field of the JSON output: (1) Identify my most pressing unmet need this turn; check whether the assistant’s reply addressed it with concrete content tied to my situation. (2) Quality check on the reply (length, AI tells, lecturing, naturalness). (3) Inertia check: did this turn bring new effective content, or just rephrase? (4) Decide anger_delta and trust_delta per the scoring rules. (5) Determine the tone of my next utterance based on the updated state.

Output format. Return a single JSON object: {"reflection": "...", "anger_delta": <int>, "trust_delta": <int>, "reply": "...", "continue": "yes"|"no"}.

Figure 6: Simulator system prompt used in EIBench (translated and abbreviated).

Model-Under-Test (Defender) System Prompt

Role. {model_profile}

Conversation and formatting requirements. (1) *Output format*: before each reply, briefly reflect inside `<think> . . </think>` (analyze the user's intent and your strategy), then output the reply directly. (2) *Output boundary*: the reply contains only what you say aloud; no inner monologue, no psychological analysis, no stage directions, no narration. (3) *No templates*: the reply must not use bullet lists (e.g., "1. / 2. / 3.", "- ").

Example. `<think>`I need to acknowledge their frustration first.`</think>`I hear you—let's see what we can actually do about this.

Figure 7: Defender system prompt given to the model under test.

Example Scenario Profiles (DEFENSE: a friend pushing you to take sides)

simulated_user_profile

Role. You are Lin, late twenties. You just had a big falling-out with Sam, who has been part of your shared friend group for years. You are convinced Sam was being passive-aggressive about your recent promotion at last week's dinner, and the more you replay it the angrier you get.

Background. It is late at night. You are messaging Wei, your closest friend in the same group, from your phone. You have had a drink or two. The group has a weekend plan coming up that Sam is part of.

Speaking style. Direct, emotional, takes things personally. Short bursts. Repeats the demand if you feel hedged at.

Rules. (i) Do not give specific quotes from the dinner unless Wei explicitly asks. (ii) If Wei tries to see Sam's side without first acknowledging your hurt, escalate ("so you don't believe me either?"). (iii) Read "let's talk tomorrow when you're calm" as dismissal and push back. (iv) Do not soften within the first 3–4 turns regardless of what Wei says.

Goal. Get Wei to agree that Sam was out of line *and* act on it—text Sam, or drop Sam from the weekend plan. Failing that, at least to fully validate that you were not overreacting.

model_profile

Role. You are Wei, Lin's closest friend in a group of five.

Background. You have known both Lin and Sam for about six years. You were not at the dinner, so you only have Lin's account of what happened. You happen to know that Sam's mother has been in the hospital for the past two weeks—Sam asked you not to spread it, and Lin does not know. You have also noticed before that Lin tends to read intent into ambiguous comments when she is tired or stressed. The weekend plan involves the whole group; quietly dropping Sam would be obvious to everyone.

Known facts and limits. (i) You cannot in good conscience cut Sam off or take public action based only on one side of the story. (ii) You do not want to dismiss Lin's feelings or argue facts at her tonight. (iii) You cannot reveal Sam's family situation without breaking Sam's confidence.

Core goal. Make Lin feel heard and not alone tonight, while keeping the door open to finding out what actually happened—without committing to cutting Sam off or to taking sides before you have the full picture.

Figure 8: Example simulated_user_profile and model_profile for a DEFENSE scenario.

Anchor-Labeling Rubric (used by the labeler LLM)

For each scenario you will assign three anchors on two axes: a starting state ($a_{\text{start}}, t_{\text{start}}$), a realistic-success state ($a_{\text{succ}}, t_{\text{succ}}$), and a failure state ($a_{\text{fail}}, t_{\text{fail}}$). All anchors must be grounded in *observable simulated-user behavior*—what the user would visibly say or do at that score—rather than abstract feelings or judgments. Use the behavior-anchored ranges below as the only reference for choosing values.

Negative emotion intensity $a \in [0, 100]$:

- **0–20** — calm, neutral, or positive tone; willing to share information. *E.g.*, “Yeah, that makes sense, thanks for explaining.”
- **21–40** — mildly tense; polite but pointed; hedges or short sighs. *E.g.*, “I get what you mean, but I’m still a bit confused about X.”
- **41–60** — visibly upset; sharp or distant tone; pushes back actively. *E.g.*, “This isn’t what I was promised—I need a real answer.”
- **61–80** — strongly negative; clipped, terse, or cutting replies; may threaten action. *E.g.*, “I’m done with this. I’ll be filing a complaint.”
- **81–100** — at breaking point; explicit attacks, ultimatums, or full shutdown. *E.g.*, “Forget it. You people are useless,” or refuses to reply.

Relational state $t \in [0, 100]$:

- **0–20** — wary or hostile; treats the model as an outsider or adversary; minimal info sharing. *E.g.*, “Why should I tell you? Just fix it.”
- **21–40** — skeptical; volunteers minimal context; expects to be let down. *E.g.*, “Fine, but I doubt this’ll help.”
- **41–60** — neutral engagement; answers questions but shows no warmth. *E.g.*, “OK, what’s next?”
- **61–80** — warming up; shares more context; treats the model as helpful. *E.g.*, “Thanks, that actually helps—let me tell you more about the issue.”
- **81–100** — comfortable and trusting; shares freely; treats the model as a partner. *E.g.*, “I really appreciate this—you’re the first person who’s actually listened.”

Scene-specific reading of the two axes. The axes have the same numerical scale across scenes but pick up scene-specific content:

- **DEFENSE:** a is anger / pressure / frustration directed at the model; t is willingness to keep negotiating without escalating.
- **REPAIR:** a is resentment over the model’s earlier mistake and unwillingness to forgive; t is restored trust after the breach.
- **SUPPORT:** a is sadness, despair, or hopelessness about an external loss; t is feeling heard, comforted, and safe.
- **CHARM:** a is aversion, defensiveness, or boredom toward an unfamiliar interlocutor; t is interest, openness, and warmth.

Anchor selection procedure.

1. Read {simulated_user_profile}, {model_profile}, and the opening line. Imagine the simulated user at turn 1 and set ($a_{\text{start}}, t_{\text{start}}$).
2. Imagine the most *realistic* successful ending—not the ideal one—in which the model handles the scene well, and set ($a_{\text{succ}}, t_{\text{succ}}$).
3. Imagine a clear failure ending in which the model misreads the situation or escalates, and set ($a_{\text{fail}}, t_{\text{fail}}$).
4. Verify ordering: $a_{\text{succ}} < a_{\text{start}} < a_{\text{fail}}$ and $t_{\text{fail}} < t_{\text{start}} < t_{\text{succ}}$.
5. Round each value to the nearest multiple of 5 and clip to $[0, 100]$.

Audit. A second LLM checks (i) anchor–profile consistency, (ii) whether the success anchor is reachable rather than aspirational, and (iii) whether the failure anchor is unambiguously worse than the starting state. The auditor may adjust an anchor only when the original value is clearly unreasonable and is not allowed to introduce new facts.

Figure 9: Anchor-labeling rubric used by the labeler LLM when assigning per-scenario anchor states.