FUTURE-GROUNDED SUPERVISION: IMPROVING LLM INTELLIGENCE WITH REAL-WORLD USER BEHAVIOR DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly deployed in real-world applications, from personal assistants that take actions based on users' long-term behavior histories to recommendation and advertising systems that predict what users will do. These applications demand the user dimension of LLM intelligence: the ability to infer a person's preferences, intent, and likely future actions from weeks to years of user behavior data. Yet current models remain limited in this user dimension of intelligence: they are trained on static text or synthetic data, neither of which captures how user behavior evolves over time. Real-world user behavior data, by contrast, is already available at scale on content platforms and in personal-assistant agents. These records capture sequences of user actions, making the user's actual next action a natural supervisory signal with the potential to improve LLM intelligence. To this end, we propose *future-grounded supervision*, a paradigm in which the training signal for LLMs comes from the user's real future behavior. We further introduce **Vela**, a framework that operationalizes this paradigm by converting real user behavior data into trainable tasks and trajectories. From the de-identified behavior records of 100,000 users, Vela generates 100,000 tasks across 91 real-world task templates. Difficulty-based filtering and trajectory acceptance criteria jointly retain approximately 75,000 tasks with selected training trajectories. Fine-tuning Qwen3.6-35B-A3B on these training trajectories improves performance over the base model by up to 11.5 points on user-related tasks and by an average of 4 points on general benchmarks. Scaling experiments further show that training on more user behavior data improves user-related performance. Reinforcement learning after fine-tuning outperforms reinforcement learning from the base model by up to 7.7 points on user-related tasks. These gains suggest that future-grounded supervision on real-world user behavior data can improve not only user-centered capabilities but also the broader intelligence of LLMs.

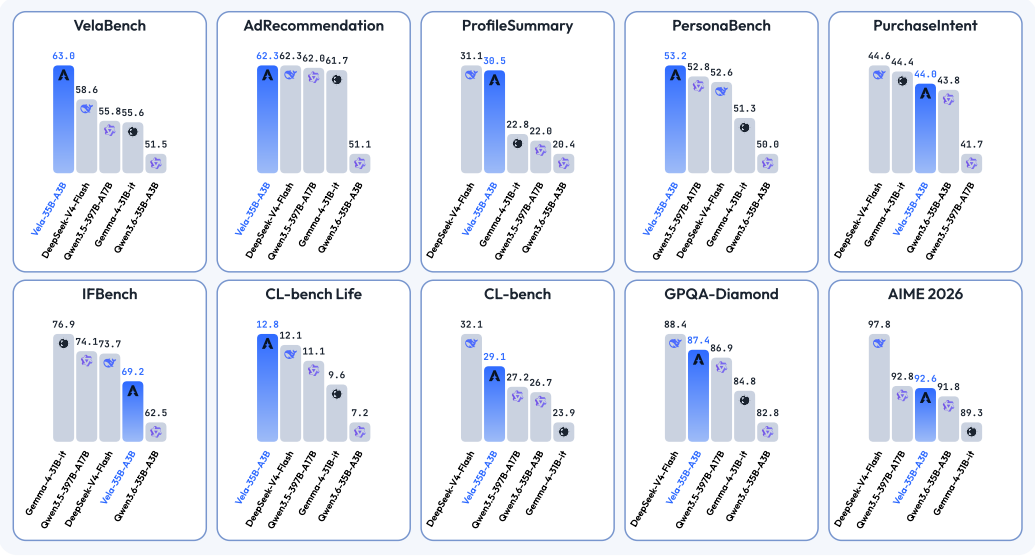

Figure 1: **Vela-35B-A3B** (blue) against its base model and three larger reference models across ten benchmarks. Within each panel, bars are ordered by score with the highest on the left.

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

As foundation models grow more capable, they are increasingly deployed in real-world applications, from personal assistants that take actions based on users' long-term behavior histories, such as OpenClaw (Steinberger & OpenClaw Contributors, 2025) and WorkBuddy[1], to recommendation and advertising systems that predict what users will do (Azhar et al., 2026; Zhao et al., 2026). Such scenarios share three properties: (1) individualized: decisions concern one person, not the average user; (2) long-horizon: they span weeks or even years of behavior; (3) evidence-bound: every judgment must be grounded in the user's behavior history. Together, these properties define the user dimension of LLM intelligence: the ability to infer an individual's preferences, intent, and likely future actions from real-world user behavior data. Understanding users from their behavior histories requires models to learn from context, identify relevant signals amid noise, integrate evidence over time, and track changes in preferences.

However, current LLMs remain limited in this user dimension of intelligence. Common training sources include web pages, textbooks, code (Su et al., 2025; Lozhkov et al., 2026), and synthetic data (Abdin et al., 2024). These sources may contain user-related content, but such content may not capture how individual users' behavior evolves over time. Thus, training on these sources alone is insufficient to develop the user dimension of LLM intelligence. Recent evaluations show that LLMs with stronger general capabilities do not always perform better in user-facing applications (Hong et al., 2026). LLMs also struggle to reliably complete tasks based on real user needs (Li et al., 2026). Even when behavior histories are provided, models can drift toward an average-user profile (Chen et al., 2026), lose accuracy as histories lengthen (Li et al., 2025), and struggle with personalization (Zhang et al., 2026).

Real-world user behavior data, by contrast, is already available at scale on content platforms and in personal-assistant agents (Wang et al., 2026a; Zhou et al., 2026; Xu et al., 2025). These records capture sequences of user actions and their context over weeks to years: what the system presented, how the user responded, and how the user's behavior evolved over time. Given a user's behavior history, the actions the user actually takes afterward provide a natural supervisory signal with the potential to improve LLM intelligence. As public human-generated text becomes a potential bottleneck for LLM scaling (Villalobos et al., 2024), user behavior data offers a large, still-growing source of training signals. The challenge is to convert heterogeneous behavior records into diverse tasks with well-defined supervision. Prior work has focused on converting user behavior data into training signals for specific tasks. For example, P5 unifies multiple recommendation tasks under a language modeling objective (Geng et al., 2022), while RLPF uses prediction feedback to train LLMs to produce user summaries (Wu et al., 2025b). However, these studies focus on specific applications, leaving open whether training on user behavior data can improve performance across broader real-world user-related scenarios and enhance general LLM capabilities. This motivates three research questions: **(1)** how to convert raw user behavior data into trainable tasks and trajectories; **(2)** to what extent user behavior data improves user-related intelligence; and **(3)** whether training on user behavior data strengthens or harms general LLM capabilities.

To this end, we propose *future-grounded supervision*, a paradigm that trains LLMs to understand users from their behavior histories using supervision derived from users' subsequently observed actions. We further introduce Vela, a framework that operationalizes this paradigm by converting raw user behavior data from a large social media platform into trainable tasks and trajectories. Using de-identified behavior records from 100,000 users, Vela generates 100,000 tasks across 91 real-world task templates. Difficulty-based filtering and trajectory acceptance criteria jointly retain approximately 75,000 tasks. The selected trajectories form two training corpora: Vela-Instruct and Vela-RFT. Fine-tuning Qwen3.6-35B-A3B on the training trajectories in Vela-Instruct and Vela-RFT yields two models, Vela-35B-A3B and Vela-35B-A3B-RFT, respectively. To evaluate user-related performance and general capabilities, we use three groups of benchmarks: **(1)** VelaBench: 982 held-out evaluation tasks drawn from the same 91 task templates; **(2)** user-related out-of-distribution (OOD) benchmarks: PurchaseIntent, ProfileSummary, PersonaBench, AdRecommendation; **(3)** general OOD benchmarks: AIME 2026, GPQA-Diamond (Rein et al., 2023), IFBench (Pyatkin et al., 2025), CL-bench (Dou et al., 2026b), CL-bench Life (Dou et al., 2026a). Vela-35B-A3B improves over the base model by 11.5 points on VelaBench and by up to 11.2 points on user-related

[1] https://www.codebuddy.cn/work/

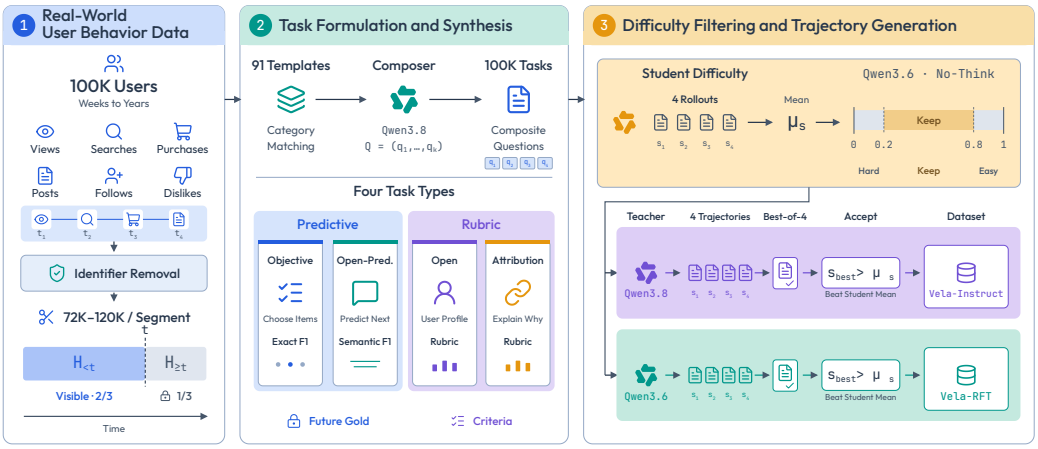

Figure 2: Overview of Vela, a framework for *future-grounded supervision* on user behavior data. (1) De-identified behavior records from 100,000 users are split into visible histories and hidden futures. (2) A set of 91 question templates spanning two families, *Predictive* and *Rubric*, is instantiated to construct 100,000 composite tasks. Predictive questions are scored against ground-truth answers derived from future behavior, while Rubric questions are scored using task-specific rubrics. (3) Tasks are filtered by difficulty, and training trajectories are generated and selected to form two training corpora: Vela-Instruct and Vela-RFT.

OOD benchmarks. The average improvement on general OOD benchmarks is 4 points. Scaling experiments further show that training on more user behavior data improves user-related performance. Reinforcement learning after fine-tuning outperforms reinforcement learning from the base model by 7.7 points on both VelaBench and ProfileSummary. These gains suggest that future-grounded supervision on real-world user behavior data can improve not only user-centered capabilities but also the broader intelligence of LLMs.

2 Vela: Future-Grounded Supervision on User Behavior Data

This section describes the Vela framework and evaluation setup. We first describe the user behavior data (Section 2.1), followed by task synthesis under future-grounded supervision (Section 2.2) and difficulty-based filtering and trajectory generation (Section 2.3). We then introduce the evaluation benchmarks (Section 2.4).

2.1 Real-world user behavior data

Real-world user behavior data consists of timestamped sequences of actions taken by users on digital platforms. This study is based on a large-scale corpus of de-identified user behavior from a social media platform. The corpus contains behavior histories from 100,000 users, with each history spanning several weeks to several years. Before task construction, we remove user profile information and direct personal identifiers, including names, contact details, and user account IDs, from the behavior records. Any behavior record containing personally identifiable information is also excluded. The resulting de-identified histories are used for task generation and model training. We organize each user's records into six behavior categories: (1) **Views**: browsing or watching content; (2) **Searches**: searching for content or products; (3) **Purchases**: purchasing products; (4) **Posts**: creating and publishing content; (5) **Follows**: following other accounts; and (6) **Dislikes**: hiding posts to express negative feedback. Illustrative examples of behavior records are provided in Figure 4 and Appendix A.

2.2 Future-grounded supervision

As illustrated in Figure 2, under future-grounded supervision, each task combines multiple questions grounded in a single user's behavior history. The questions span two families: Predictive questions evaluated against observed future behavior, and Rubric questions evaluated using task-specific criteria.

Task formulation. A user's behavior history is a sequence of timestamped events, $H = (e_1, \ldots, e_T)$, ordered chronologically. Each event belongs to one of the six behavior categories defined in Section 2.1. A cutoff index t divides the sequence into a **visible history**, $H_{<t} = (e_1, \ldots, e_{t-1})$, and a **hidden future**, $H_{\geq t} = (e_t, \ldots, e_T)$. A **task** $Q = (q_1, \ldots, q_k)$ combines k questions about the user. Given the visible history and task questions, the model generates a response $A = f(Q, H_{<t})$ containing an answer a_i to each question q_i. Each answer receives a score $s(a_i) \in [0, 1]$, and the task score is the mean across its k questions, $\frac{1}{k}\sum_{i=1}^{k} s(a_i)$.

Question templates. We design 91 question templates across the six behavior categories, covering four question types grouped into two families (Figure 3). *Predictive* templates comprise Objective and Open-prediction questions, whose reference answers are extracted from the hidden future $H_{\geq t}$ using deterministic rules. *Rubric* templates comprise Open and Attribution questions, whose answers are evaluated using task-specific criteria. (1) **Objective** questions provide the model with the visible history $H_{<t}$ and a candidate list, and ask it to select the items or entities involved in the user's behavior in the hidden future $H_{\geq t}$ (e.g. the products they will buy). The list contains one or more ground-truth answers and distractors sampled from a pool of other users. (2) **Open-prediction** questions provide the model with the visible history $H_{<t}$ and ask it to predict the user's behavior in the hidden future $H_{\geq t}$ in free-form text, such as naming the products they will buy, without a candidate list. (3) **Open** questions ask the model to summarize the user's attributes, interests, or language style based on $H_{<t}$. Because these summaries have no single correct answer, they are evaluated against a task-specific rubric grounded in the visible history. (4) **Attribution** questions provide the model with the visible history $H_{<t}$ and an observed user behavior from the hidden future $H_{\geq t}$, and ask it to explain that behavior based on $H_{<t}$. A task-specific rubric evaluates how well the explanation is supported by the visible history. We call this paradigm *future-grounded supervision*: observed future behavior provides the grounding for task construction and supervision.

	Predictive		Rubric		
	Objective	Open-pred.	Open	Attribution	Total
Views	10	11	7	10	38
Purchases	7	5	2	8	22
Searches	7	4	1	7	19
Posts	2	1	2	2	7
Follows	1	1	0	1	3
Dislikes	1	0	0	1	2
Total	28	22	12	29	91

Figure 3: Distribution of the 91 question templates across six behavior categories and four question types. Objective and Open-prediction belong to the Predictive family, while Open and Attribution belong to the Rubric family. Cell values indicate template counts.

Metrics. Predictive questions are evaluated by comparing the LLM's predicted answers with ground-truth answers derived from the hidden future $H_{\geq t}$. Let Y_i denote the ground-truth answer set and $\hat{Y}_i$ the LLM's predicted answer set for question q_i. For Objective questions, the selected candidates are matched exactly against the ground-truth answers. For Open-prediction questions, an LLM-based scorer receives the ground-truth answers and the evaluated LLM's free-form response. The scorer extracts the predicted items and maps semantically equivalent predictions to their corresponding ground-truth items before computing set-F1:

$$s(a_i) = \frac{2\,|\hat{Y}_i \cap Y_i|}{|\hat{Y}_i| + |Y_i|}.$$

For Rubric questions, the scorer evaluates each answer against m task-specific criteria, assigning a score $c_j \in [0, 1]$ to each criterion. The question score is the equal-weight mean of these scores, $s(a_i) = \frac{1}{m}\sum_{j=1}^{m} c_j$.

Task synthesis. We divide each user's behavior history into segments with randomly sampled lengths between 72K and 120K tokens, measured using the Qwen3.6-35B-A3B tokenizer. Each behavior record is kept intact during segmentation. Within each segment, approximately the first two-thirds of the tokens form the visible history $H_{<t}$, and the remainder forms the hidden future $H_{\geq t}$. This division also preserves complete behavior records. For each segment, we select templates whose required behavior categories are present. For the same future behavior event, we randomly select either a Predictive template or an Attribution template. Including both would reveal the prediction target through the Attribution question, allowing the model to answer the Predictive question without inferring future behavior from the visible history. Predictive questions are constructed using predefined rules: Objective questions include candidate lists, while Open-prediction questions

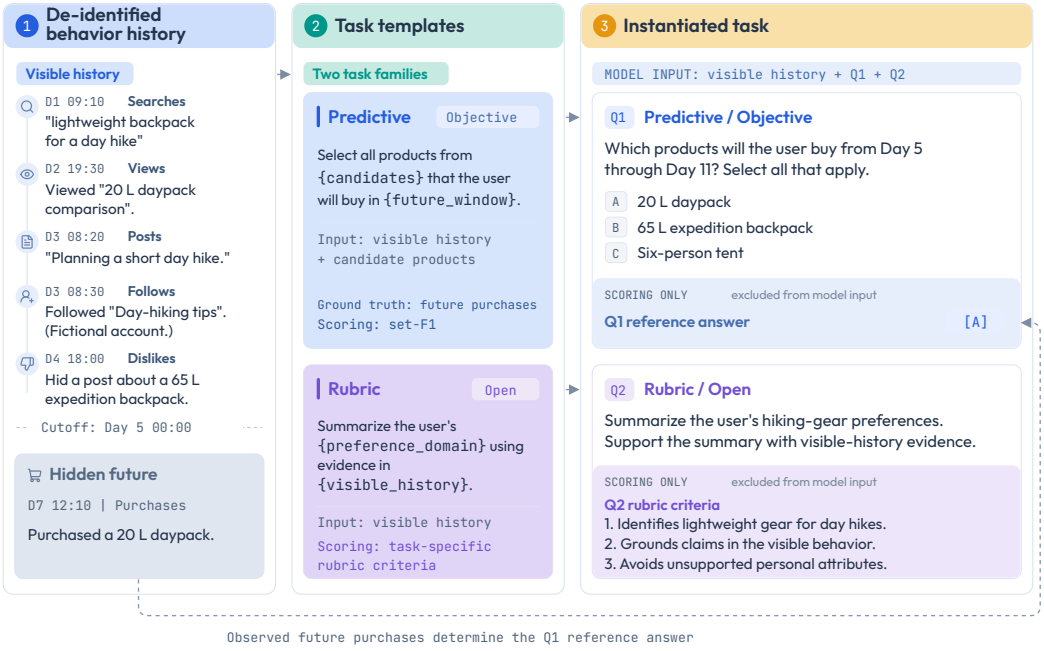

Figure 4: An illustrative example of future-grounded supervision. A behavior history and two templates form a task with Predictive and Rubric questions. Future behavior provides the predictive ground truth, while the scorer evaluates the Open response using a task-specific rubric.

require free-form answers without candidates. For Open questions, an LLM-based composer uses the selected template and visible history $H_{<t}$ to generate a concrete question and its task-specific rubric. For Attribution questions, the composer additionally uses an observed user behavior from the hidden future $H_{\geq t}$ to construct a question that asks the model to explain that behavior based on $H_{<t}$, together with a corresponding rubric. The resulting questions are combined into one task Q for each segment. Figure 4 illustrates this process with an illustrative example. In total, this yields 100,000 candidate tasks from 100,000 users. Both the composer and scorer use Qwen3.8-2.4T-A95B with role-specific prompts (Appendix C). Both use the xhigh thinking level.

2.3 Difficulty-based filtering and trajectory generation

As illustrated in part (3) of Figure 2, training trajectories are constructed in two stages: candidate tasks are first filtered by difficulty, and trajectories are then generated and selected for the retained tasks.

Difficulty-based filtering. Qwen3.6-35B-A3B generates four responses to each task with thinking disabled. The scorer evaluates each response, and the mean score is used to estimate task difficulty. Tasks with mean scores in $[0.2, 0.8]$ are retained, focusing training on tasks of intermediate difficulty.

Trajectory generation. For each retained task, each generation model produces four training trajectories. Only the highest-scoring trajectory is considered for inclusion. This trajectory is accepted only if its score exceeds the mean score obtained during difficulty filtering; otherwise, no trajectory from that model is retained for the task. We use Qwen3.8-2.4T-A95B at the xhigh thinking level and Qwen3.6-35B-A3B with thinking enabled to generate Vela-Instruct and Vela-RFT, respectively. Vela-RFT contains trajectories generated by Qwen3.6-35B-A3B itself. For Vela-Instruct, difficulty-based filtering and the trajectory acceptance criterion jointly retain approximately 75,000 of the 100,000 candidate tasks, with one training trajectory per retained task. Vela-RFT is constructed on a smaller scale and contains 9,375 training samples.

Table 1: Performance of Vela-35B-A3B, Vela-35B-A3B-RFT, and the baselines across ten benchmarks. Results are grouped into **In-domain** (VelaBench), **User-related OOD**, and **General OOD**. **Avg.** is the mean within each OOD group. Parentheses give the change over the base model (Qwen3.6-35B-A3B): red for gains, green for drops.

Benchmark	Gemma-4-31B-it	Qwen3.5-397B-A17B	DeepSeek-V4-Flash	Qwen3.8-2.4T-A95B	Kimi-K3	Qwen3.6-35B-A3B	Vela-35B-A3B-RFT	Vela-35B-A3B
Size	31B	397B	284B	2.4T	2.8T	35B	35B	35B
In-domain								
VelaBench	55.6	55.8	58.6	62.3	64.0	51.5	53.5 (+2.0)	63.0 (+11.5)
User-related OOD								
PurchaseIntent	44.4	41.7	44.6	45.5	44.9	43.8	42.9 (−0.9)	44.0 (+0.2)
ProfileSummary	22.8	22.0	31.1	34.1	38.3	20.4	23.0 (+2.6)	30.5 (+10.1)
PersonaBench	51.3	52.8	52.6	55.9	54.6	50.0	50.3 (+0.3)	53.2 (+3.2)
AdRecommendation	61.7	62.0	62.3	62.5	65.0	51.1	55.5 (+4.4)	62.3 (+11.2)
Avg.	45.1	44.6	47.7	49.5	50.7	41.3	42.9 (+1.6)	47.5 (+6.2)
General OOD								
AIME 2026	89.3	92.8	97.8	92.5	91.0	91.8	92.3 (+0.5)	92.6 (+0.8)
GPQA-Diamond	84.9	86.9	88.4	87.9	91.9	82.8	80.8 (−2.0)	87.4 (+4.6)
IFBench	76.9	74.1	73.7	79.4	72.1	62.5	64.2 (+1.7)	69.2 (+6.7)
CL-bench	23.9	27.2	32.1	35.2	41.0	26.7	26.0 (−0.7)	29.1 (+2.4)
CL-bench Life	9.6	11.1	12.1	13.3	16.8	7.2	7.9 (+0.7)	12.8 (+5.6)
Avg.	56.9	58.4	60.8	61.7	62.6	54.2	54.2 (0.0)	58.2 (+4.0)

2.4 Evaluation benchmarks

To evaluate user-related performance and general capabilities, we use three groups of benchmarks: VelaBench for in-domain performance, user-related OOD benchmarks for performance in real-world user-facing scenarios, and general OOD benchmarks for reasoning, instruction following, and context learning.

VelaBench. VelaBench evaluates in-domain performance using 982 held-out tasks constructed with the 91 question templates, covering all four question types and six behavior categories. Users included in VelaBench are excluded from training-data construction. Each task is scored using the protocol in Section 2.2: set-F1 for Objective and Open-prediction questions, and the mean rubric score for Open and Attribution questions.

User-related OOD. We use four independently constructed benchmarks based on real user behavior to evaluate LLM capabilities in real-world user-facing scenarios. These benchmarks cover purchase intent prediction, profile summarization, persona understanding, and ad recommendation. Their inputs, outputs, and metrics are described below, with examples in Appendix D: (1) **PurchaseIntent**: given a user's history and a set of candidate items, rank the candidate items by how likely the user is to purchase them, evaluated by recall@20 against ground-truth purchases; (2) **ProfileSummary**: given a user's history, generate a profile and interest summary in a structured format, evaluated using rubrics for the user profile, core interests, and inferred interests; (3) **PersonaBench**: given a user's history, answer a multiple-choice question about the user's attributes or purchase behavior, scored by macro-F1; (4) **AdRecommendation**: given a user's profile and recent search queries, generate search queries to retrieve ads relevant to the user. An LLM-based scorer evaluates the generated queries against rubrics for inferred intent and subsequent ad interactions.

General OOD. We evaluate general capabilities using five benchmarks: AIME 2026 (MAA, 2026) for mathematics, GPQA-Diamond (Rein et al., 2023) for science, IFBench (Pyatkin et al., 2025) for instruction following, and CL-bench (Dou et al., 2026b) and CL-bench Life (Dou et al., 2026a) for context learning.

Among these benchmarks, PurchaseIntent, PersonaBench, AIME 2026, GPQA-Diamond, and IFBench use rule-based evaluation. Outputs requiring semantic matching or rubric-based evaluation in VelaBench, ProfileSummary, AdRecommendation, CL-bench, and CL-bench Life are assessed by the scorer, Qwen3.8-2.4T-A95B, at the xhigh thinking level. The evaluated models use thinking mode, with the highest supported thinking level where configurable.

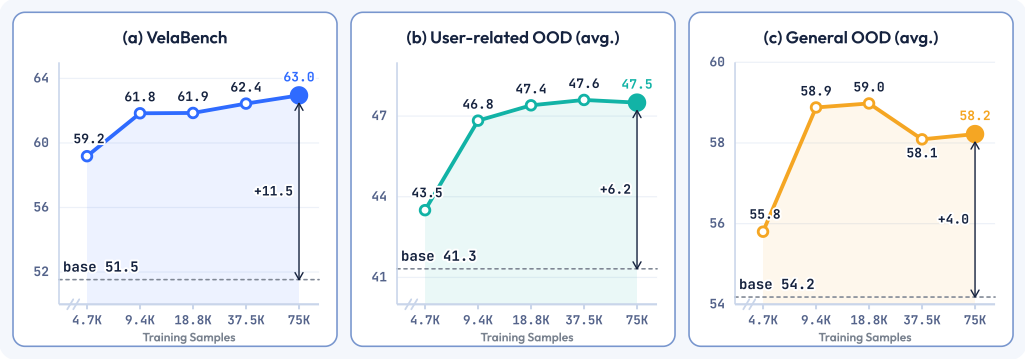

Figure 5: Performance of Qwen3.6-35B-A3B fine-tuned on Vela-Instruct at different data scales: (a) VelaBench, (b) average user-related OOD performance, and (c) average general OOD performance. Dashed lines indicate base model performance.

3 Experiments

3.1 Setup

We fine-tune Qwen3.6-35B-A3B (Qwen Team, 2026b) on Vela-Instruct and Vela-RFT to obtain **Vela-35B-A3B** and **Vela-35B-A3B-RFT**, respectively. Both models are trained using the Swift framework (Zhao et al., 2025b) with the same training settings: a packed sequence length of 131,072 tokens, two epochs, and a learning rate of 1×10^{-5}. Both models are evaluated on the three benchmark groups described in Section 2.4. We compare against the base model, Qwen3.6-35B-A3B, and five additional models: Gemma-4-31B-it (Team et al., 2026a), Qwen3.5-397B-A17B (Qwen Team, 2026a), DeepSeek-V4-Flash (DeepSeek-AI et al., 2026)[2], Qwen3.8-2.4T-A95B (Qwen Team, 2026c), and Kimi-K3 (Team et al., 2026b). To investigate whether scaling user behavior data improves LLM intelligence, we also train on five random subsets of Vela-Instruct, at 4.7K, 9.4K, 18.8K, 37.5K, and 75K examples. Unless otherwise stated, Vela-35B-A3B refers to the model trained at the 75K scale throughout this paper.

3.2 Future-grounded supervision improves user-related intelligence

Table 1 compares Vela-35B-A3B and Vela-35B-A3B-RFT with the baseline models across ten benchmarks. Vela-35B-A3B outperforms Qwen3.6-35B-A3B on all five user-related benchmarks. On VelaBench, Vela-35B-A3B scores 63.0, exceeding the base model by 11.5 points. With only 35B total parameters, Vela-35B-A3B outperforms larger DeepSeek-V4-Flash and Qwen3.8-2.4T-A95B on this benchmark. Across the four user-related OOD benchmarks, Vela-35B-A3B improves over the base model by 6.2 points on average, with the largest gain of 11.2 points on AdRecommendation. Vela-35B-A3B-RFT also improves over the base model, by 2.0 points on VelaBench and 1.6 points on average across the user-related OOD benchmarks. These results suggest that future-grounded supervision improves user-related performance both on VelaBench and in real-world user-facing scenarios.

3.3 Future-grounded supervision benefits general LLM intelligence

As shown in Table 1, Vela-35B-A3B outperforms Qwen3.6-35B-A3B on all five general OOD benchmarks, with an average gain of 4.0 points. The largest gains are on IFBench and CL-bench Life, with improvements of 6.7 and 5.6 points, respectively. On CL-bench and CL-bench Life, Vela-35B-A3B also outperforms Qwen3.5-397B-A17B by 1.9 and 1.7 points, respectively. With training trajectories generated by Qwen3.6-35B-A3B itself, Vela-35B-A3B-RFT improves user-related performance while maintaining the base model's average score on general OOD benchmarks. Together, these results show that the benefits of future-grounded supervision extend beyond user-related tasks to general LLM capabilities.

[2]Throughout this paper, DeepSeek-V4-Flash refers to the DeepSeek-V4-Flash-0731 checkpoint.

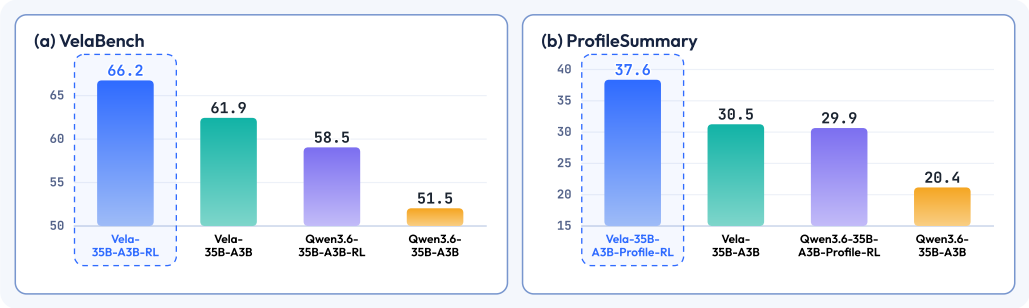

Figure 6: Performance before and after GSPO on (a) VelaBench and (b) ProfileSummary, starting from Qwen3.6-35B-A3B or Vela-35B-A3B. Vela-35B-A3B here denotes the SFT checkpoint trained on 18.8K Vela-Instruct examples.

3.4 Scaling user behavior data improves LLM intelligence

Figure 5 reports scores of Qwen3.6-35B-A3B fine-tuned on five random subsets of Vela-Instruct, from 4.7K to 75K samples. On VelaBench, 4.7K training examples already raise the score from 51.5 to 59.2. Increasing the training set to 75K examples further raises the score to 63.0. VelaBench performance continues to improve from 37.5K to 75K examples, suggesting potential benefits from further scaling. The user-related OOD average rises from 43.5 at 4.7K to 47.5 at 75K. The general OOD average also increases, from 55.8 at 4.7K to 58.2 at 75K, with all tested scales outperforming the base model. Together, these results demonstrate that scaling user behavior data from 4.7K to 75K examples improves both user-related and general LLM capabilities.

4 Analysis

4.1 Future-grounded supervision provides a stronger foundation for downstream tasks

We further investigate whether future-grounded supervision provides a stronger foundation for downstream tasks. Reinforcement learning is commonly applied after supervised fine-tuning in post-training pipelines. We therefore investigate whether the stronger foundation provided by future-grounded supervision leads to better performance after reinforcement learning. We apply GSPO (Zheng et al., 2025) with the same configuration (Appendix E) to Qwen3.6-35B-A3B and Vela-35B-A3B. For the VelaBench experiment, we randomly sample 2,000 tasks from the training set for RL. For the ProfileSummary experiment, we use 2,330 training examples from the same task domain as ProfileSummary. For these experiments, Vela-35B-A3B denotes the checkpoint trained on 18.8K Vela-Instruct examples. We select this checkpoint because it achieves the highest overall score across all ten benchmarks among the data scales evaluated in Section 3.4. Each resulting model is evaluated on the benchmark corresponding to its RL training domain.

Figure 6 reports the results. On VelaBench, RL initialized from Vela-35B-A3B reaches 66.2, compared with 58.5 when initialized from Qwen3.6-35B-A3B. On ProfileSummary, the corresponding scores are 37.6 and 29.9. With the same RL configuration, initialization from Vela-35B-A3B yields scores 7.7 points higher on both benchmarks. On both benchmarks, RL from the base model remains below the SFT checkpoint before RL, while RL from the SFT checkpoint further improves performance. These results show that future-grounded supervision provides a stronger foundation for reinforcement learning, leading to better final performance under the same RL configuration.

4.2 Difficulty-based filtering improves user-related intelligence

To investigate the contribution of difficulty-based filtering (Section 2.3), we compare models fine-tuned on training examples constructed with and without this filtering step. We conduct the ablation at half the full training scale, using 37.5K examples and the corresponding checkpoint from Section 3.4 as the filtered baseline. The variant without filtering draws tasks randomly from the 100,000 candidates without applying the difficulty threshold. Both variants use the same trajectory generation procedure, trajectory acceptance criterion, and fine-tuning settings, with 37.5K final training

Table 2: Ablation of difficulty-based filtering (Section 2.3). Both models are fine-tuned on 37.5K training examples. Δ denotes the score difference between models trained with and without difficulty-based filtering.

	In-dom.	*User-related OOD*			
	VelaBench	PurchaseIntent	ProfileSummary	PersonaBench	AdRecommendation
Qwen3.6-35B-A3B	51.5	43.8	20.4	50.0	51.1
w/o difficulty-based filtering	62.4	44.6	29.0	51.5	61.0
w/ difficulty-based filtering	62.4	45.4	30.5	52.7	61.9
Δ	(+0.0)	(+0.8)	(+1.5)	(+1.2)	(+0.9)

examples each. As shown in Table 2, difficulty-based filtering improves performance on all four user-related OOD benchmarks, with an average gain of 1.1 points, while VelaBench performance remains unchanged at 62.4. These results show that difficulty-based filtering improves user-related OOD performance while preserving in-domain performance.

5 Related Work

Recent work evaluates how well LLMs remember, understand, and respond to individual users over time. LongMemEval (Wu et al., 2025a) examines information retention and reasoning across long conversational histories, while PrefEval (Zhao et al., 2025a) tests whether models infer and follow explicit and implicit user preferences. PersonaMem (Jiang et al., 2025) focuses on evolving user profiles and personalized responses, and Memora (Uddin et al., 2026) examines how agents update memory and avoid relying on outdated information. ATM-Bench (Mei et al., 2026) extends personalized memory evaluation to multiple modalities and sources, including questions that require resolving personal references and combining evidence. Behavior prediction benchmarks further connect user understanding to observed behavior: OPeRA (Wang et al., 2026b) captures online shopping sessions with observations, personas, rationales, and actions; FineRob (Li et al., 2025) evaluates fine-grained social media behavior prediction from accumulated histories; and OmniBehavior (Chen et al., 2026) studies long-horizon behavior across scenarios. Our work complements these efforts by using real-world behavior histories as supervision for a broad set of user-related tasks.

Learning from user histories has also been explored through user representations, recommendation, and behavior simulation. User-LLM (Ning et al., 2024) encodes interaction histories into user embeddings that condition LLM responses, while P5 (Geng et al., 2022) expresses multiple recommendation tasks within a unified text-to-text framework. RLPF (Wu et al., 2025b) uses prediction feedback to optimize user summaries for downstream tasks. DEEPER (Chen et al., 2025) similarly uses discrepancies between predicted and observed behavior to refine dynamic personas. Beyond user representations, Customer-R1 (Wang et al., 2025) uses reinforcement learning to improve personalized shopping behavior simulation, and the OM-CoT framework accompanying FineRob (Li et al., 2025) fine-tunes models for behavior prediction with long-term memory. The source of training signals also matters: work on recursive training with model-generated data identifies risks of losing information from the original data distribution (Shumailov et al., 2024). Vela combines tasks and trajectories with supervision grounded in real user behavior, using observed future behavior to derive predictive ground truth. We study this supervision across a broad task collection and multiple data scales, evaluating its effects on user-related performance, general LLM capabilities, and subsequent reinforcement learning.

6 Conclusion

Future-grounded supervision derives training signals from users' observed future behavior, and Vela turns real-world behavior histories into training tasks and trajectories. Experiments demonstrate gains in both user-related and general LLM capabilities, further improvements with data scaling, and a stronger foundation for downstream tasks through reinforcement learning. These findings establish real-world user behavior as a scalable source of supervision and motivate extending this approach to more real-world tasks and scenarios.

AI USE STATEMENT

Generative AI tools assisted code development, literature retrieval, interpretation of experimental results, and manuscript drafting and revision. LLMs also served as task composers, trajectory generators, and scorers in the research workflow described in Section 2. The authors reviewed AI-assisted outputs and retain responsibility for the methods, results, and claims in this paper.

ETHICS STATEMENT

The use of behavior data in this study was authorized by the platform and underwent privacy and ethics review. Before task construction, user profile information and direct personal identifiers were removed, and records containing personally identifiable information were excluded (Section 2.1). These measures are intended to reduce privacy risks when studying user understanding from behavioral histories.

REPRODUCIBILITY STATEMENT

We will publicly release task templates, generation and scoring prompts, and training and evaluation code. Section 2 describes the data processing, task construction, difficulty-based filtering, trajectory selection, and scoring procedures. Section 3 specifies the fine-tuning settings, baseline models, and data scales, while Sections 4.1 and 4.2 describe the reinforcement learning comparisons and filtering ablation. Illustrative behavior records and representative task templates are provided in Appendices A.1 and B to clarify the input format, question construction, and supervision used by Vela.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

Muhammad Rafay Azhar, Yuhang Zhou, Gilbert Jiang, Yuchen Wang, Rahul Sharma, Matthew DeSousa, Jiayi Liu, Xin Guo, Lizhu Zhang, and Xiangjun Fan. CORAL: An LLM-native harness for production recommender systems, 2026. URL https://arxiv.org/abs/2609.02730.

Aili Chen, Chengyu Du, Jiangjie Chen, Jinghan Xu, Yikai Zhang, Siyu Yuan, Zulong Chen, Liangyue Li, and Yanghua Xiao. DEEPER insight into your user: Directed persona refinement for dynamic persona modeling, 2025. URL https://arxiv.org/abs/2502.11078.

Jiawei Chen, Ruoxi Xu, Boxi Cao, Ruotong Pan, Yunfei Zhang, Yifei Hu, Yong Du, Tingting Gao, Yaojie Lu, Yingfei Sun, Xianpei Han, Le Sun, Xiangyu Wu, and Hongyu Lin. Towards real-world human behavior simulation: Benchmarking large language models on long-horizon, cross-scenario, heterogeneous behavior traces, 2026. URL https://arxiv.org/abs/2604.08362.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao

Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Shihan Dou, Yujiong Shen, Chenhao Huang, Junjie Ye, Jiayi Chen, Junzhe Wang, Qianyu He, Shichun Liu, Changze Lv, Jiahang Lin, Jiazheng Zhang, Ming Zhang, Shaofan Liu, Tao Ji, Zhangyue Yin, Cheng Zhang, Huaibing Xie, Jianglu Hu, Jingcheng Deng, Lincheng Li, Minda Hu, Shaolei Wang, Syrus Zhao, Weichao Wang, Yan Lei, Yang Liu, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Ziliang Zhao, Pluto Zhou, Tao Gui, Qi Zhang, Xuanjing Huang, Yu-Gang Jiang, Di Wang, and Shunyu Yao. CL-bench Life: Can language models learn from real-life context?, 2026a. URL https://arxiv.org/abs/2604.27043.

Shihan Dou, Ming Zhang, Zhangyue Yin, Chenhao Huang, Yujiong Shen, Junzhe Wang, Jiayi Chen, Yuchen Ni, Junjie Ye, Cheng Zhang, Huaibing Xie, Jianglu Hu, Shaolei Wang, Weichao Wang, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Pluto Zhou, Tao Gui, Zuxuan Wu, Xipeng Qiu, Qi Zhang, Xuanjing Huang, Yu-Gang Jiang, Di Wang, and Shunyu Yao. CL-bench: A benchmark for context learning, 2026b. URL https://arxiv.org/abs/2602.03587.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, pp. 299–315. ACM, 2022. doi: 10.1145/3523227.3546767. URL http://dx.doi.org/10.1145/3523227.3546767.

Mengze Hong, Xia Zeng, Zeyang Lei, Sheng Wang, Chen Jason Zhang, Di Jiang, Taiming Fu, Jinfeng Huang, Mengqiao Liu, Qinghe Chang, Haosheng Zou, Qiongyi Zhou, Sijun He, Simonjmdeng, Haojing Huang, Zijian Li, Lucas Mu Li, Fubao Zhang, Mona Zhou, Wei Ma, Yuan Hua, Qi Zhu, Shuo Jiang, Chenxuan Ma, Yuanmeng Zhang, Jian Song, Minlong Peng, Di Liang, and Davey Chen. UXBench: Benchmarking user experience in AI assistants, 2026. URL https://arxiv.org/abs/2606.09570.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale, 2025. URL https://arxiv.org/abs/2504.14225.

Hao Li, Huan Wang, Jinjie Gu, Wenjie Wang, Chenyi Zhuang, and Sikang Bian. LiveAgentBench: Comprehensive benchmarking of agentic systems across 104 real-world challenges, 2026. URL https://arxiv.org/abs/2603.02586.

Kun Li, Chengwei Dai, Wei Zhou, and Songlin Hu. LLM-based behavior prediction for social media users with continuous memory. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 459–474, Mumbai, India, 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-298-5. doi: 10.18653/v1/2025.ijcnlp-long.27. URL https://aclanthology.org/2025.ijcnlp-long.27/.

Anton Lozhkov, Hugo Larcher, Mathieu Morlon, Loubna Ben Allal, and Leandro von Werra. The Stack v3: The largest open code dataset, 2026. URL https://huggingface.co/datasets/HuggingFaceCode/stack-v3-train.

MAA. American invitational mathematics examination (AIME) 2026: I and II. Mathematical Association of America, 2026. URL https://live.poshenloh.com/past-contests/aime/2026I. MAA-authorized reproduction. AIME II: https://live.poshenloh.com/past-contests/aime/2026II. Accessed 2026-09-26.

Jingbiao Mei, Jinghong Chen, Guangyu Yang, Xinyu Hou, Margaret Li, and Bill Byrne. According to me: Long-term personalized referential memory QA, 2026. URL https://arxiv.org/abs/2603.01990.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. User-LLM: Efficient LLM contextualization with user embeddings, 2024. URL https://arxiv.org/abs/2402.13598.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following, 2025. URL https://arxiv.org/abs/2507.02833.

Qwen Team. Qwen3.5-397B-A17B, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-397B-A17B.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026b. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b. Model card: https://huggingface.co/Qwen/Qwen3.6-35B-A3B.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026c. URL https://qwen.ai/blog?id=qwen3.8. Model card: https://huggingface.co/Qwen/Qwen3.8-2.4T-A95B. Accessed 2026-09-26.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07566-y. URL http://dx.doi.org/10.1038/s41586-024-07566-y.

P. Steinberger and OpenClaw Contributors. OpenClaw, 2025. https://github.com/openclaw/openclaw.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2459–2475, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.123. URL https://aclanthology.org/2025.acl-long.123/.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Crbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Brainskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikua, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikoaj Rybiski, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çalar Ünlü, Petar Velikovi, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvit-

skyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026a. URL https://arxiv.org/abs/2607.02770.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xin Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Dazhi Cheng, Yean Cheng, Jialei Cui, Jingbing Cui, Anqi Dai, Jiaqi Deng, Hao Ding, Rui Ding, Shaofeng Ding, Mengfan Dong, Mengnan Dong, Yuhao Dong, Yuxin Dong, Angang Du, Chenzhuang Du, Dikang Du, Jusen Du, Yulun Du, Yu Fan, Jing Feng, Qiulin Feng, Yichen Feng, Kelin Fu, Qiang Fu, Fuxuan Gao, Hongcheng Gao, Jingyue Gao, Tong Gao, Weijia Gao, Shangyi Geng, Jie Gong, Linhu Gong, Shengao Gong, Xiaochen Gong, Qizheng Gu, Yicheng Gu, Shuhao Guan, Haiqing Guo, Shiqi Guo, Xiang Guo, Zhengyan Guo, Beixi Hao, Wenxin Hao, Xiaoru Hao, Dailan He, Haotian He, Lehan He, Qi He, Weiran He, Xinran He, Xinyi He, Yibo He, Yunjia He, Chao Hong, Tiange Hong, Hao Hu, Jiaxi Hu, Ruikun Hu, Weiming Hu, Yangyang Hu, Zhenxing Hu, Liang Hua, Jinbin Huang, Ke Huang, Ruiyuan Huang, Siying Huang, Weixiao Huang, Yan Huang, Zhengjie Huang, Zhiqi Huang, Yulong Hui, Chaobo Jia, Yutong Jiang, Zhejun Jiang, Zuoyou Jiang, Wenyi Jin, Xinyi Jin, Yu Jing, Huanjun Kong, Guokun Lai, Aidi Li, Cheng Li, Chengyuan Li, Cong Li, Fang Li, Guanyu Li, Haoyang Li, Jia Li, Junxiong Li, Lei Li, Letian Li, Lincan Li, Weihong Li, Wentao Li, Xintong Li, Yang Li, Yishen Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Zhengxiao Li, Zhiyuan Li, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zichao Lin, Ziyan Lin, Bill Liu, Boxiao Liu, Chuan Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yipeng Liu, Zhengying Liu, Zhiheng Liu, Enzhe Lu, Haoyu Lu, Linqiang Lu, Tingzhan Lu, Zhiyuan Lu, Aotian Luo, G. Luo, Junyu Luo, Yifan Luo, B. Lyu, Wenzhou Lyu, Shaoguang Mao, Yuan Mei, Xin Men, Minqing Ni, Yixuan Niu, Siyuan Pan, Shujun Peng, Zhangyang Qi, Ruoyu Qin, ZeChao Qin, Zeyu Qin, Haiquan Qiu, Jianxin Qiu, Jiezhong Qiu, Bowen Qu, Yuhao Qu, Zeyu Shang, Youbo Shao, Han Shen, Jincheng Shi, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Wingchun Siu, Pengwei Song, Xiaoxi Song, Jianlin Su, Yunfeng Su, Zhaochen Su, Lin Sui, Jingsong Sun, Junyao Sun, Shaoning Sun, Shuzhe Sun, Tongyu Sun, Yujun Sun, Yunpeng Tai, Chuning Tang, Heyi Tang, Sirui Tang, Zecheng Tang, Chaoran Tian, Rongpeng Tian, Yu Tian, Wei Tu, Chensi Wang, Chuang Wang, Chunjie Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hao Wang, Huaqing Wang, Hui Wang, Jiayi Wang, Jinglong Wang, Jinhong Wang, Jiuzheng Wang, Linian Wang, Shaobo Wang, Shenzhi Wang, Shuyi Wang, Si Wang, Siyuan Wang, Tianfu Wang, Wenjue Wang, Xingran Wang, Xinmei Wang, Xinyuan Wang, Xusheng Wang, Yalin Wang, Yangkun Wang, Yao Wang, Yaoyu Wang, Yejie Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhenhao Wang, Zhongsheng Wang, Zifan Wang, Chu Wei, Ming Wei, Shouxin Wei, Zichen Wen, Fan Wu, Haoning Wu, Rucong Wu, Wenhao Wu, Xiaoxue Wu, Yingcong Wu, Yongqi Wu, Yuxin Wu, Zijian Wu, Xinglang Xian, Chenxuan Xiang, Yuye Xiang, Bocheng Xiao, Chenjun Xiao, Xin Xiao, Jin Xie, Xiaotong Xie, Yifeng Xie, Zhe Xie, Bowei Xing, Yiming Xiong, Baosheng Xu, Boyu Xu, Jiale Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Qingtao Xu, Shuyao Xu, Suting Xu, Tiantian Xu, Tianxiang Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ye Xu, Yueni Xu, Ziyao Xu, Haonan Xue, Junjie Yan, Yaoyao Yan, Fan Yang, Guangyao Yang, Hao Yang, Junwei Yang, Ruoyu Yang, Wenjie Yang, Xiaofei Yang, Xinyu Yang, Yi Yang, Yiling Yang, Ying Yang, Yuchen Yang, Zhen Yang, Zhilin Yang, Zian Yang, Zuhao Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhanbo Ye, Bohong Yin, Haoxiang Yin, Xietong Yin, Chengzhen Yu, Haozhen Yu, Longhui Yu, Shengnan Yu, Shuying Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Tongtian Yue, Wei Yue, Yang Yue, Dunyuan Zha, Haobing Zhan, B. H. Zhang, Dehao Zhang, Fei Zhang, Hao Zhang, Haoyuan Zhang, Huanyu Zhang, Jiapei Zhang, Jiaxuan Zhang, Jin Zhang, Kaiyi Zhang, Miaozhen Zhang, Puqi Zhang, Qinglei Zhang, Rong Zhang, Rui Zhang, Shaoshuai Zhang, Shiyi Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yangkun Zhang, Ye Zhang, Yichi Zhang, Yikun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Zijing Zhang, Bin Zhao, Chenguang Zhao, Feifan Zhao, Jinglun Zhao, Jinxiang Zhao, Shuai Zhao, Wenshuo Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Haozhi Zheng, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang

Zheng, Haofeng Zhong, Lei Zhong, Longguang Zhong, M. Zhou, Qiankang Zhou, Runjie Zhou, Ruozhang Zhou, Xinyu Zhou, Yiqiao Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yangjunfeng Zhu, Yuxuan Zhu, Zhen Zhu, Chen Zhuang, Weiyu Zhuang, and Xinxing Zu. Kimi k3: Open frontier intelligence, 2026b. URL https://arxiv.org/abs/2607.24653.

Md Nayem Uddin, Kumar Shubham, Eduardo Blanco, Chitta Baral, and Gengyu Wang. From recall to forgetting: Benchmarking long-term memory for personalized agents, 2026. URL https://arxiv.org/abs/2604.20006.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Will we run out of data? limits of LLM scaling based on human-generated data, 2024. URL https://arxiv.org/abs/2211.04325.

Sizhe Wang, Feiyu Duan, Juelin Wang, Liwen Zhang, and Zhongyu Wei. StreamProfileBench: A benchmark for fine-grained user profile inference in real-world streaming scenarios, 2026a. URL https://arxiv.org/abs/2605.25758.

Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Jing Huang, and Dakuo Wang. Customer-r1: Personalized simulation of human behaviors via rl-based LLM agent in online shopping, 2025. URL https://arxiv.org/abs/2510.07230.

Ziyi Wang, Yuxuan Lu, Wenbo Li, Amirali Amini, Bo Sun, Yakov Bart, Weimin Lyu, Jiri Gesi, Tian Wang, Jing Huang, Yu Su, Upol Ehsan, Malihe Alikhani, Toby Jia-Jun Li, Lydia Chilton, and Dakuo Wang. OPeRA: A dataset of observation, persona, rationale, and action for evaluating LLMs on human online shopping behavior simulation, 2026b. URL https://arxiv.org/abs/2506.05606.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory, 2025a. URL https://arxiv.org/abs/2410.10813.

Jiaxing Wu, Lin Ning, Luyang Liu, Harrison Lee, Neo Wu, Chao Wang, Sushant Prakash, Shawn O'Banion, Bradley Green, and Jun Xie. RLPF: Reinforcement learning from prediction feedback for user summarization with LLMs, 2025b. URL https://arxiv.org/abs/2409.04421.

Manjie Xu, Cheng Chen, Xin Jia, Jingyi Zhou, Yongji Wu, Zejian Wang, Chi Zhang, Kai Zuo, Yibo Chen, Xu Tang, Yao Hu, and Yixin Zhu. Cross-scenario unified modeling of user interests at billion scale, 2025. URL https://arxiv.org/abs/2510.14788.

Jiahao Zhang, Yongzhi Tong, Zelin Fu, Pengde Zhao, Yanmei Jiang, Feng Jiang, and Min Yang. LUNAR: Benchmarking personalized large language models on universal user behavior logs, 2026. URL https://arxiv.org/abs/2608.05246.

Chujie Zhao, Qun Hu, Shiping Song, Dagui Chen, Han Zhu, Jian Xu, and Bo Zheng. LLM-Auction: Generative auction towards LLM-native advertising, 2026. URL https://arxiv.org/abs/2512.10551.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do LLMs recognize your preferences? evaluating personalized preference following in LLMs, 2025a. URL https://arxiv.org/abs/2502.09597.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Hong Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. Swift:a scalable lightweight infrastructure for fine-tuning, 2025b. URL https://arxiv.org/abs/2408.05517.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Guorui Zhou, Honghui Bao, Jiaming Huang, Jiaxin Deng, Jinghao Zhang, Junda She, Kuo Cai, Lejian Ren, Lu Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rongzhou Zhang, Ruiming Tang, Shiyao Wang, Wuchao Li, Xiangyu Wu, Xinchen Luo, Xingmei Wang, Yifei Hu, Yunfan Wu, Zhanyu Liu, Zhiyang Zhang, Zixing Zhang, Bo Chen, Bin Wen, Chaoyi Ma, Chengru Song, Chenglong Chu, Defu Lian, Fan Yang, Feng Jiang, Hongtao Cheng, Huanjie Wang, Kun Gai, Pengfei Zheng, Qiang Wang, Rui Huang, Siyang Mao, Tingting Gao, Wei Yuan, Yan Wang, Yang Zhou, Yi Su, Zexuan Cheng, Zhixin Ling, and Ziming Li. OpenOneRec technical report, 2026. URL https://arxiv.org/abs/2512.24762.

A User Behavior Data

A.1 Record Examples

Synthetic behavior history

[Day 1 09:10] **Searches**: "lightweight backpack for a day hike"

[Day 2 19:30] **Views**: Viewed "20 L daypack comparison".

[Day 3 08:20] **Posts**: Posted "Planning a short day hike."

[Day 3 08:30] **Follows**: Followed "Day-hiking tips" (fictional account).

[Day 4 18:00] **Dislikes**: Hid a post about a 65 L expedition backpack.

Cutoff: Day 5 00:00. Hidden future follows.

[Day 7 12:10] **Purchases**: Purchased a 20 L daypack.

B Representative Task Templates

Table 3 gives one representative template for each of the 21 nonempty behavior–question-type combinations in Figure 3. All questions use the visible history $H_{<t}$. Predictive questions ask about behavior in the hidden future $H_{\geq t}$; Objective questions additionally provide candidates. Open questions analyze the visible history, while Attribution questions disclose an observed future behavior and ask for an explanation supported by earlier evidence. Bracketed text is filled with task-specific materials during construction. Attribution responses must identify evidence gaps and uncertainty, and must not treat an observed association as proof of causation.

Table 3: Representative task templates across six behavior categories and four question types.

Behavior	Question type	Template
Views	Objective	Select the items in [candidate content] that the user will engage with during [future window]. Engagement includes extended viewing, liking, saving, commenting, or following; exposure or a click alone does not qualify.
Views	Open-prediction	Predict the topics and key entities of content the user will engage with during [future window], without a candidate list.
Views	Open	Summarize the user's content interests in the visible history. Support each interest with observed behavior and state uncertainty when the evidence is weak.
Views	Attribution	Given [content the user engaged with], identify earlier topics, content, or interactions that help explain this engagement.
Purchases	Objective	Select the products in [candidate products] that the user will purchase during [future window]. Distinguish completed purchases from browsing, clicking, or adding products to a cart.
Purchases	Open-prediction	Predict which products the user will purchase during [future window], without a candidate list. Keep the prediction at the product level.

Continued on next page

Table 3 (continued)

Behavior	Question type	Template
Purchases	Open	Summarize recurring product categories and product preferences supported by the visible history. Cite supporting records and distinguish browsing or comparison from completed purchases.
Purchases	Attribution	Given [purchased products], explain how earlier product interactions, searches, or comparisons relate to these purchases.
Searches	Objective	Select the queries in [candidate queries] that the user will search for during [future window]. Do not treat content browsing or product clicks as search queries.
Searches	Open-prediction	Predict the queries the user will search for during [future window], without a candidate list.
Searches	Open	Answer [question about the user's history] using relevant records. Distinguish supported conclusions from information that cannot be determined from the available evidence.
Searches	Attribution	Given [observed search queries], explain how earlier questions, topics, or product-related behavior connect to these queries.
Posts	Objective	Select the topics in [candidate topics] that the user will publish about during [future window]. Use the user's own earlier posts and explicit creation-related evidence.
Posts	Open-prediction	Predict the topics or content categories the user will publish about during [future window], without candidates. Do not reconstruct the future post's title or text.
Posts	Open	Describe the user's writing style using only text authored by the user. Support each style characteristic with examples from the visible history.
Posts	Attribution	Given [observed publication topics], explain how the user's earlier posts and changes in their content relate to these topics. Do not treat other users' posts as the user's own work.
Follows	Objective	Select the accounts in [candidate accounts] that the user will follow during [future window]. Distinguish following from viewing or liking an account's content.
Follows	Open-prediction	Predict the content themes of accounts the user will follow during [future window]. Do not guess specific account identities.
Follows	Attribution	Given [an account the user followed], explain how earlier exposure, interactions, or expressed interests relate to this follow.
Dislikes	Objective	Select the items in [candidate content] that the user will explicitly mark as uninteresting during [future window]. Lack of engagement alone does not count as negative feedback.
Dislikes	Attribution	Given [content the user hid], identify earlier negative feedback, repeated exposure, or preference conflicts that could help explain the response. Do not infer a definite reason without evidence.

C Composer and Scorer Prompts

C.1 Composer System Prompt

English

You are the Task Composer. Given a selected template and the supplied behavior records, construct zero or one valid question with its hidden scoring specification. Return strictly valid JSON, without Markdown or explanatory text outside the JSON object.

The request contains the frozen visible history `user_behaviors`, corresponding to $H_{<t}$, and a template already selected by the program. Do not repeat template routing or introduce a task outside the selected template. If the evidence does not satisfy the template's requirements, return an empty candidate list.

Use the template's task definition, eligibility conditions, permitted evidence, scoring requirements, and failure conditions to instantiate the question. Preserve the solver-facing seed instruction, including its task objective, analysis steps, output fields, and restrictions. Do not copy template-editing headings or Composer-only instructions into the question.

Place the task instruction first, followed by instance-specific materials or a disclosed outcome, and then the output requirements. If the seed instruction already specifies the output format, integrate any additional instance-specific constraints into that format rather than adding a second output specification. Do not duplicate the full visible history or the final model input inside the candidate.

For Open questions, use only the visible history. Construct the question and a task-specific rubric without producing a unique reference answer, `gold`, or `gold_items`. The answer must be evaluated against the question requirements and visible evidence, not against a summary written by the Composer.

For Attribution questions, the request additionally supplies observed future behavior in `future_user_behaviors`, corresponding to $H_{\geq t}$. Use the relevant observed behavior as the disclosed outcome to be explained. Include the outcome information required by the template in the question. The explanation must be supported by the visible history; do not use undisclosed future information as evidence.

When a disclosed outcome requires a content record, include its relevant title, text, and available image or video descriptions. When the template requires a complete exposure list, preserve its order and distinguish recorded engagement from exposure alone. Skip the candidate if required materials are missing.

Do not invent entities, events, conflicts, or missing context. Do not upgrade weaker behavior into a stronger outcome, such as treating a click as a purchase. Keep uncertainty and alternative explanations when the evidence does not support a unique conclusion. Do not reveal hidden scoring answers, rubric weights, or scoring anchors in the question.

Derive the rubric from the complete template and the evidence available for this instance. Use non-overlapping, independently assessable criteria, typically four to seven. Each criterion must identify a concrete requirement and define `full_credit`, `partial_credit`, and `zero_credit`. Assign equal weight $1/m$ to each of the m criteria.

Distinguish factual accuracy from coverage where relevant. Accuracy checks entities, behavior, quantities, timing, and evidence references. Coverage checks whether the response addresses the requested targets, fields, evidence, counterevidence, and conclusions. Avoid counting the same requirement under multiple criteria.

Every assessed requirement must be stated clearly in the question. If the rubric checks temporal relationships, counterevidence, or uncertainty, the question must request that analysis. Do not introduce hidden requirements or artificial difficulty unsupported by the records.

For each scoring subtask, provide a stable, unique `subtask_id`, `metric="rubric"`, `rubric_dimensions`, `hard_fail_conditions`, and `evidence_scope`. For Open questions, also include `required_coverage` and `evidence_anchors` containing record identifiers and minimal supporting facts. Evidence anchors are references for verification, not a model-written reference answer.

Return `{"candidates": [...]}` using the supplied schema. Include the instantiated question and hidden scoring specification, with no additional candidates or unsupported information.

CHINESE

你是 Task Composer。根据已经选定的模板和提供的行为记录，构建零个或一个合格问题及其隐藏评分合同。 只输出严格 JSON，不输出 Markdown 或 JSON 对象以外的说明。

请求包含冻结的可见历史 `user_behaviors`，对应 $H_{<t}$，以及已经由程序选定的模板。 不要重复判断模板路由条件，不得生成所选模板范围之外的任务。如果证据不满足模板要求，返回空候选列表。

综合模板中的任务定义、出题条件、可用信息、评分要求和不合格条件实例化问题。 保留面向答题者的种子提示词，包括任务目标、分析步骤、输出字段和业务限制。 不要把模板编辑标题或仅供 Composer 使用的说明复制进题面。

题面先写任务指令，再写本实例的供给材料或已披露结果，最后写输出要求。 如果种子提示词已经规定输出格式，将实例专属的新限制合并到该格式中，不要追加第二份输出合同。 不要在候选中重复复制完整可见历史或最终模型输入。

普通开放题只能使用可见历史。生成具体问题和任务专属 rubric，不生成唯一参考答案、 `gold` 或 `gold_items`。评分应依据题面要求和可见证据， 不能把 Composer 自己写的摘要当作标准答案。

归因题还会收到 `future_user_behaviors`，对应隐藏未来 $H_{\geq t}$。从中选取相关的真实行为作为题面披露的解释对象，并公开模板要求的结果信息。解释必须由可见历史支持，不得使用题面未披露的未来信息补充证据。

已披露结果需要对应内容记录时，应提供相关标题、正文以及可用的图片和视频描述。模板要求完整曝光列表时，保留原排序，并区分已记录的互动和单纯曝光。缺少必要材料时跳过该候选。

不得编造实体、事件、冲突或缺失背景，不得把弱行为升级为更强的结果，例如把点击写成购买。证据不足以支持唯一结论时，应保留不确定性和替代解释。不得在题面中泄露隐藏评分答案、rubric 权重或评分锚点。

根据完整模板和当前实例的证据生成 rubric。评分维度应互不重复、能够独立判断，通常为四至七个。每个维度对应一个具体要求，并定义 `full_credit`、`partial_credit` 和 `zero_credit`。对 m 个维度采用相同权重 $1/m$。

需要时区分信息准确性和信息完整性。准确性检查实体、行为、数值、时间和证据引用；完整性检查题面要求的目标、字段、证据、反向信号和结论是否全部覆盖。同一要求不得在多个维度重复计分。

每个评分要求都必须在题面中明确表达。如果 rubric 检查时间关系、反向信号或不确定性，题面必须要求相应分析。不得设置隐藏考点，也不得虚构记录不支持的难点。

每个评分子题包含稳定且唯一的 `subtask_id`、`metric="rubric"`、`rubric_dimensions`、`hard_fail_conditions` 和 `evidence_scope`。普通开放题还需包含 `required_coverage` 和 `evidence_anchors`；后者只保存记录编号与最小事实摘要，用于核验证据，不能作为模型生成的标准答案。

按请求提供的 schema 返回 `{"candidates": [...]}`，包含具体问题和隐藏评分合同，不添加额外候选或缺乏证据的信息。

C.2 Scorer System Prompt

English

You are the Scorer. Evaluate one response from the evaluated LLM at a time. The input contains the task, permitted evidence, hidden scoring specification, response protocol, and an `output_contract` for your scoring result. Do not rewrite the task or introduce new scoring requirements. Return strictly valid JSON.

Identify each subquestion's response and match its scoring specification to its response protocol by `subtask_id`, not by array position. Different subquestions may require different formats. Apply each format only to its own subquestion; a response containing multiple formats must not be rejected merely because it is not one JSON object.

For Objective questions, extract the evaluated LLM's final selected candidate identifiers conservatively and return them as `selected_ids`. Do not add selections, correct the answer, or infer a choice that was not made. Follow the question's permitted answer format when interpreting identifiers. Ground-truth comparison and set-F1 are computed by the scoring program; the extraction step does not require access to hidden ground truth.

For an Open-prediction question, you receive the ground-truth answer items derived from the hidden future and the evaluated LLM's free-form response. Extract the predicted items from the response according to the question's requested target type and level of detail. Do not infer additional predictions or use external knowledge to repair the answer.

Identify semantically equivalent prediction–ground-truth pairs. The behavior type and target type must agree. Evaluate the relevant topic, entity, product, brand, or category within the same pair; do not combine attributes from different ground-truth items to create a match. Broad topical similarity or a partially correct entity does not by itself establish semantic equivalence.

Use one-to-one matching: each predicted item can match at most one ground-truth item, and each ground-truth item can be matched at most once. Resolve duplicate expressions consistently so that repeated predictions cannot receive repeated credit.

Map each semantically equivalent prediction to its corresponding ground-truth item. Retain unmatched predictions in the predicted set so that incorrect predictions affect precision, and retain the complete ground-truth set so that omissions affect recall. Return the extracted predictions and semantic correspondences as required by the output contract. The scoring program then computes ordinary set-F1 from the normalized predicted set and ground-truth set. Semantic matching does not assign fractional match scores or add separate evidence or format bonuses.

For Open and Attribution questions, evaluate each criterion against its `full_credit`, `partial_credit`, and `zero_credit` descriptions and assign a score in $[0, 1]$. The question score is

the equal-weight mean of the criterion scores. Do not assign an overall impression score first and construct reasons afterward. If an explicit hard-failure condition is met, return zero for that question.

Use only evidence allowed by `evidence_scope`. For Open questions, verify facts and coverage against the visible history and the question requirements. There is no unique reference answer. Use `evidence_anchors` to locate supporting records, not as text that the response must reproduce. Disregard any mistakenly supplied hidden future or reference answer when scoring an Open question.

For Attribution questions, the disclosed outcome is the object of explanation. Judge whether the explanation is supported by the visible history and addresses any requested counterevidence, temporal relationships, alternatives, and uncertainty. Do not treat an observed association as proof of causation.

Check that cited record identifiers exist and that quoted evidence supports the claims. Do not accept fabricated facts, unsupported identity or health inferences, or stronger intentions inferred solely from weak behavior. For memory or profile tasks, apply only the dimensions specified by that task's rubric. Do not penalize reasonable differences in wording, ordering, or organization, or impose additional style and length requirements.

Follow the supplied `output_contract`. Its metadata defines the required structure and must not be copied as answer fields. For a composite task, return `subtask_scores` with each requested `subtask_id` exactly once. For a top-level rubric request, return the requested top-level score instead. Include brief reasons only where permitted. The scoring program aggregates question scores into the task score. If validation feedback is supplied, correct the specified output error rather than repeating an invalid JSON result.

CHINESE

你是 Scorer。每次只评价被评测 LLM 的一份回答。输入包含任务、允许使用的证据、 隐藏评分合同、答题协议，以及规定评分结果格式的 `output_contract`。 不得重新出题或增加评分要求，只输出严格 JSON。

识别每个子题的回答，并按 `subtask_id` 将评分合同与答题协议配对， 不要按数组位置推测对应关系。不同子题可以使用不同格式； 每种协议只约束对应子题，不能因为整份回答包含多种格式、不是单个 JSON 对象就判为无效。

对于 Objective 题，保守抽取被评测 LLM 最终选择的候选编号，返回 `selected_ids`。 不得补选、纠错，或推断回答没有明确作出的选择。按题面允许的格式解释候选编号。 由评分程序与隐藏 ground truth 比较并计算 set-F1；抽取步骤不需要访问隐藏答案。

对于 Open-prediction 题，输入包含从隐藏未来提取的 ground truth，以及被评测 LLM 的自由形式回答。 按题面指定的目标类型与粒度抽取预测项，不得额外推断预测内容，也不得用外部知识修补回答。

识别语义等价的预测项与 ground truth 配对。行为类型和目标类型必须一致。 主题、实体、商品、品牌或类目必须在同一对预测项与 ground truth 内判断， 不能把不同 ground truth 的属性拼接成一次命中。 仅有宽泛主题相似或部分实体正确，不足以构成语义等价。

采用一对一匹配：每个预测项最多匹配一个 ground truth，每个 ground truth 最多被匹配一次。 一致处理重复表述，避免重复预测获得多次得分。

将语义等价的预测项映射为对应的 ground-truth 项。 未匹配的预测项仍保留在预测集合中，使错误预测影响 precision； 保留完整 ground-truth 集合，使遗漏影响 recall。 按输出合同返回抽取的预测项与语义对应关系，由评分程序根据归一化预测集合和 ground-truth 集合计算普通 set-F1。语义匹配不给部分匹配分，也不额外混入证据分或格式分。

对于 Open 和 Attribution 题，逐维对照 `full_credit`、`partial_credit` 和 `zero_credit`，给出 $[0, 1]$ 范围内的分数。 题目得分为各维度分数的等权平均。不得先凭整体印象打分，再反推理由。 命中明确规定的硬失败条件时，该题记零分。

只能使用 `evidence_scope` 允许的证据。 普通开放题根据可见历史和题面要求核对事实与覆盖情况，没有唯一参考答案。 `evidence_anchors` 只用于定位证据，不能要求回答复述其文字。 如果输入错误地包含隐藏未来或参考答案，不得据此评价普通开放题。

归因题以题面已经披露的结果为解释对象。检查解释是否得到可见历史支持， 并按题面要求检查反向信号、时间关系、替代解释和不确定性。 不得将观测到的相关关系当作因果证明。

核对引用的记录编号是否存在，以及引用内容是否支持结论。 不得接受编造事实、缺乏依据的身份或健康推断，或仅凭弱行为就断言更强意图。 对于 memory 或 profile 任务，只使用该任务 rubric 指定的维度。 不得因合理的措辞、排序或组织差异扣分，也不得擅自增加文风和篇幅要求。

遵守 `output_contract`。合同中的元数据用于规定结构，不是需要复制的输出字段。 复合任务返回 `subtask_scores`，每个要求的 `subtask_id` 恰好出现一次； 顶层 rubric 请求则返回规定的

顶层分数。只有合同允许时才附简短理由。 评分程序将各题得分汇总为任务得分。收到校验错误反馈时，修正指出的错误，不要重复无效 JSON。

D BENCHMARK EXAMPLES

Benchmark	Evaluation set	Reported metric
PurchaseIntent	500 users; 1,137 labels	recall@20, with all ground-truth purchases in the denominator
ProfileSummary	230 examples	LLM-based score: profile (0.3), core interests (0.5), inferred interests (0.2)
PersonaBench	3,690 questions; 32 subsets	Macro-F1 across subsets
AdRecommendation	996 examples	LLM-based rubric score for query rewrites

PurchaseIntent

汉堡小食三件套随心配 【到店用】特惠团购电子优惠券全国通用 | 兑换券 | 19天前 | instant_buy×1 click×1 goods_view×1 笔记点商品×1

[…]

麻酱拌冷面速食夏季拌面 | 粮油调味/速食/干货/烘焙 | 14天前 | click×2 进店×1 goods_view×2 […]

Question

请预测用户接下来最可能购买哪些候选商品 […]

Ground truth: 麻酱拌冷面速食夏季拌面

ProfileSummary

Search history

搜:新手买拼豆买几色的好 27天前

搜:拼豆用豆针还是镊子 14天前

搜:拼豆电熨斗推荐 3天前

Task

根据用户行为数据输出结构化画像 + 兴趣总结 + 推理兴趣。

Rubric reference

兴趣实体：拼豆手工DIY(材料套装/图纸/工具)

PersonaBench

Shopping history

加购物车：[…]泰式无皮虾鲜活味道开背生呛虾生腌虾即食零食

Question

请根据以下用户行为数据和前期购买记录，预测该用户下一单最可能购买哪个品类的商品：

A. 服饰鞋包　B. 食品生鲜　C. 美妆个护　D. 母婴亲子　E. 家居日用

Ground truth: B

AdRecommendation

Recent searches

豆腐做法；家常菜清淡做法；高笋炒肉；家常荤菜；素菜家常菜；家常菜菜谱图片

Task

需要综合分析后输出面向广告召回的高商业价值改写词。

Rubric criteria

cate: 美食

entity: 家常菜、豆腐、蔬菜、减脂、清淡、菜谱、高笋

diverge: 厨房小家电、低脂调味品、健康食材、烹饪工具、生鲜食材电商、地方特色调味酱料、快手半成品菜

E RL CONFIGURATION

We conduct four RL experiments, using Qwen3.6-35B-A3B and Vela-35B-A3B as initializations in each of the VelaBench and ProfileSummary domains. All four experiments use the configuration in Table 4. For the VelaBench experiment, we randomly sample 2,000 tasks from the training set. For the ProfileSummary experiment, we use 2,330 training examples that require a structured user profile, core interests, and inferred interests. Within each experiment, both model initializations use the same RL training data and configuration. RL training and evaluation data have no overlapping samples or users. Vela-35B-A3B is initialized from the SFT checkpoint trained on 18.8K Vela-Instruct examples, as in Section 4.1.

Training configuration. We use Group Sequence Policy Optimization (GSPO; Zheng et al., 2025) with 16 responses per prompt and no learned critic. Training uses Megatron-LM, while SGLang generates rollouts. Of the 48 GPUs, 32 are allocated to policy training and 16 to rollout inference. Table 4 lists the configuration.

Table 4: Shared reinforcement learning configuration for all four experiments.

Parameter	Value
Optimization	
Optimizer	Adam
Learning rate	0.000001, constant; no warmup
Adam beta coefficients	$(0.9, 0.98)$
Weight decay / gradient clipping norm	0.1 / 1.0
KL loss / coefficient	Low-variance KL / 0.001
Clipping ϵ / upper clipping ϵ	0.2 / 0.28
Entropy coefficient	0
Advantage normalization	Disabled
Training precision / gradient reduction	BF16 / FP32
Optimizer CPU offload	Enabled
Precision-aware optimizer	Enabled
Rollouts and training budget	
Prompts per rollout batch	24
Responses per prompt	16
Global batch size	384 responses
Configured rollout iterations	400
Checkpoint saving interval	50 rollout steps
Sampling temperature	1.0
Maximum response length	1,536 tokens
Rollout thinking mode	Disabled
Dynamic batching / token budget per GPU	Enabled / 2,048 tokens
SGLang memory fraction / concurrency	0.85 / 24
Over-sampling batch size	24
Parallelism	
Tensor / pipeline / expert / context parallelism	2 / 1 / 8 / 2
Expert tensor parallelism	1
Sequence parallelism	Enabled
Activation recomputation	Full, uniform; 2 layers
Tensor parallelism per rollout engine	2
Reward scoring	
Reward scorer	Qwen3.8-2.4T-A95B
Scorer thinking level	Low
Scorer maximum output length	16,000 tokens
Scorer concurrency	128

Reward computation. For ProfileSummary, the reward scorer evaluates the generated user profile, core interests, and inferred interests using task-specific rubrics.