Predicting User Decisions from Prediction Markets

Masters Project

Jack “Siggy” Sigler

Colorado School of Mines

April 21, 2026

Problem Motivation - Strong Human Collaboration

- **I believe collaboration is one of the most unique and powerful things humans do**
- Strong collaboration improves over time as people learn how others think .
- This shared understanding makes interaction more efficient and personalized.
- The best collaborators do not just communicate, they anticipate each other's decisions.

Problem Motivation - Strong Human Collaboration

Collaborting with LLMs: Reality

Collaborating with LLMs: The Goal

The Goal

What if I could have the same kind of implicit understanding with a model?

- I would not need to repeatedly explain my preferences, goals, or prior context.
- Instead of telling the model what to do each time, it would already understand how I approach decisions.
- This kind of personalization raises the ceiling of how effectively I can collaborate with a model.

Current Personalization Efforts

1. Stateless

One conversation context only

2. Prompt Personalization

Lightweight, manual customization

3. Memory / Retrieval

Deeper context, but still limited

What Is Missing?

What Current Systems Do

- Adapt to surface-level preferences and style
- Personalize through prompts, memory, or retrieval
- Improve continuity within or across conversations

What They Still Miss

- How a user reasons and weighs evidence
- How beliefs update over time
- A persistent representation of a specific user's decision process

Key point. Current systems personalize interaction, but they still do not learn how a specific user thinks through decisions over time.

Problem Proposal

Prior Work on Personalization

- Much of the literature focuses on **text-level personalization**, such as style, tone, or preference alignment.
- Many approaches model **synthetic personas** rather than real user behavior.
- Evaluation is typically based on text similarity or preference matching, not real-world actions.
- As a result, it remains unclear whether these methods capture how a user actually behaves.

Key point. Prior work personalizes how models respond, but not how they predict user behavior.

This Work

One-Sentence Summary

I study user-level personalization as the time-gated prediction of a specific person's decisions in real prediction markets.

Contributions

- Reframe personalization from text generation to **downstream decision prediction**.
- Build a benchmark from real Manifold user histories rather than synthetic personas.
- per-user PEFT, and agentic retrieval on the same long-horizon benchmark.
- Measure not only accuracy, but also training and inference-time resource tradeoffs.

Why Prediction Markets?

- Human decision-making data is rarely available at scale due to privacy concerns
- Sequential and timestamped user actions
- Rich contract context: question text, descriptions, market state, participant counts
- User reasoning signals such as comments and repeated participation
- Large enough histories to study persistent individual behavior

Data and Evaluation

Dataset at a Glance

Source and Scale

- Manifold Markets public API
- Approximately 180,000 users
- Approximately 9.5 million bets and 560,000 comments
- December 2021 to February 2026

Benchmark Focus

- Strongest-user cohort $\mathcal{U}^\star$
- 794,014 bets and 56,527 comments
- Histories capped at 13,000 bets per user
- **Binary and multiple-choice markets**

Strongest-User Cohort and Chronological Splits

Per-User Timeline

For each user $u \in \mathcal{U}^\star$, bets are ordered in canonical time as

$$(b_1^{(u)}, b_2^{(u)}, \ldots, b_{T_u}^{(u)})$$

and then split chronologically into $\mathcal{D}_{\text{train}}^{(u)}$, $\mathcal{D}_{\text{val}}^{(u)}$, and $\mathcal{D}_{\text{test}}^{(u)}$.

Shared Evaluation Contract The same per-user splits are used for every method, and headline results are reported on $\mathcal{D}_{\text{test}}^{(u)}$.

Task Definition: Predict the User, Not the Market

Prediction Target
For each time step i, the goal is to predict the outcome selected by user u.

$\mathcal{U}_i$: User Context

Prior bets, comments, and user-level statistics available before step i.

$\mathcal{C}_i$: Contract Context

Question text, description, outcome type, and market state for the current contract.

Experimental Unit

For bet $b_i^{(u)}$, the model sees $\mathcal{U}_i$ and $\mathcal{C}_i$, then predicts $\hat{y}_i$.

$$(\mathcal{U}_i, \mathcal{C}_i) \longrightarrow \hat{y}_i$$

Success means $\hat{y}_i = y_i$.

Task Definition Example

$\mathcal{C}_i$: Contract Context	$\mathcal{U}_i$: User Context
Question: How will Deadline rank the top 5 biggest box office bombs of 2023? **Highlighted Answer Choices:** • The Marvels • The Flash • Indiana Jones and the Dial of Destiny • Fast X • Wish	**User ID:** `Iua2KQvL6KYcfGLGNI6PVeGkseo1` **Reasoning Signal (Prior Comment):** *"Early tracking has been extremely strong... 'Deadpool & Wolverine' looking to slash records... strong contender... but unlikely to surpass Inside Out 2..."*

Ground Truth Decision: $y_i =$ Indiana Jones and the Dial of Destiny

Leakage Controls

Temporal Gating

At time step i, only information available before that step may appear in $\mathcal{U}_i$ or $\mathcal{C}_i$.

Contract Isolation

$\mathcal{U}_i$ excludes prior interactions on the same contract, preventing trivial leakage when a user splits one decision into multiple bets.

Why This Matters

All methods solve the same sequential task under the same leakage controls.

Primary metric: exact-match decision accuracy on $\mathcal{D}_{\text{test}}^{(u)}$.

System and Methods

RSB + LPB

Random Sampling Baseline (RSB)

- Simple statistical baseline that assigns uniform probability over all possible decisions ($\mathcal{Y}_i$)
- For a given $b_i^{(u)}$, the expected accuracy is $\frac{1}{|\mathcal{Y}_i|}$

LLM Prompting Baseline (LPB)

- Simple model baseline used to assess further upgrades.
- Model: **Gemini 3.1 Pro**
- For a given $b_i^{(u)}$, LPB is given $(\mathcal{U}_i, \mathcal{C}_i)$ and asked to predict $\hat{y}_i$

Per-User LoRA (PUL)

Training

- Each user $u \in \mathcal{U}^\star$ is assigned a persistent LoRA adapter
- Trained on $\mathcal{D}_{\text{train}}^{(u)}$ using retrieval-augmented prompts $(\mathcal{U}_i, \mathcal{C}_i)$
- Loss is computed on the true outcome y_i
- After each epoch, validation sweeps adapter scale $\lambda \in \{0.0, 0.33, 0.66, 1.0\}$
- The chosen λ controls how strongly the LoRA adapter is applied to the base
- Early stopping when validation performance no longer improves

Inference

- Past bets and comments are embedded and retrieved by similarity to the current contract
- Qwen2.5-7B + PEFT uses the trained adapter to predict $\hat{y}_i$ from $(\mathcal{U}_i, \mathcal{C}_i)$
- No additional training cost at inference time

Agent-Based Method (ABM)

Inference (Tool-Based Context Construction)

- The model issues tool calls to build $\mathcal{U}_i$ dynamically at prediction time
- Available tools:
  - Recent user bets (time-gated)
  - User statistics (aggregate behavior)
  - Semantically similar past bets
  - Semantically similar comments

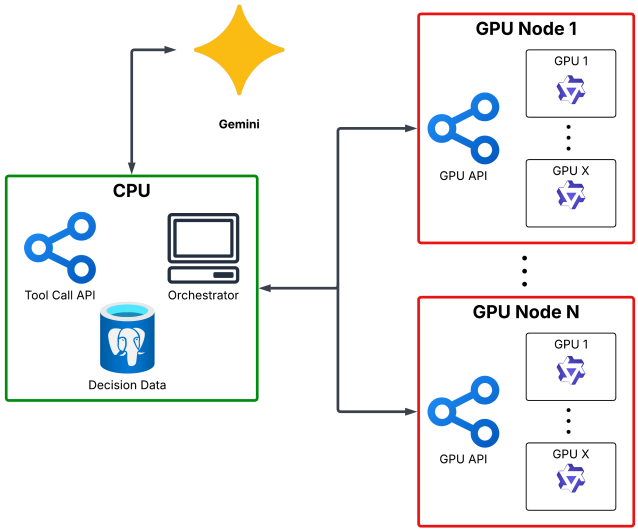
Long-Range Memory (Context Compactor)

- Past interactions are periodically compressed into a bounded text memory
- Only *future-safe* events are included to preserve temporal and contract isolation
- This compact memory is injected into future $(\mathcal{U}_i, \mathcal{C}_i)$

Prediction

- Given augmented $(\mathcal{U}_i, \mathcal{C}_i)$, the agent predicts $\hat{y}_i$

System Architecture



Results

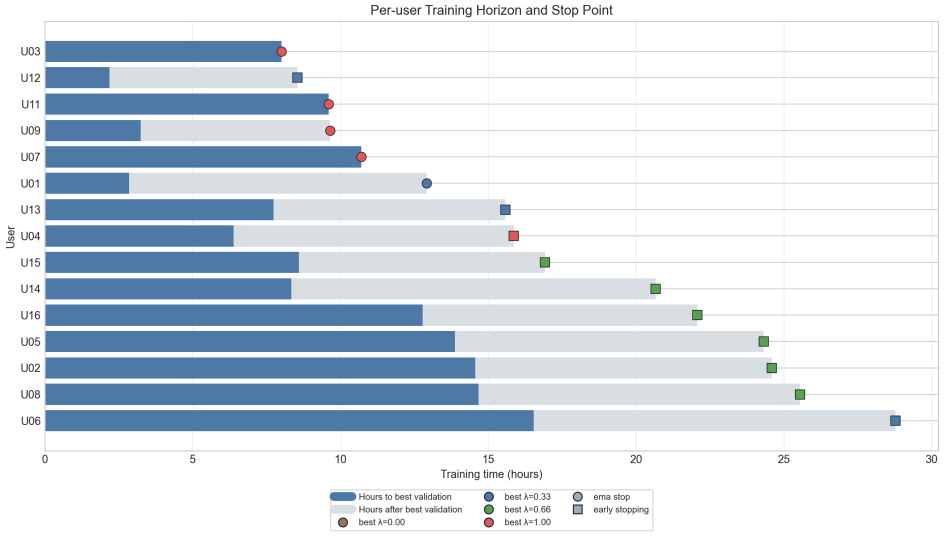
Long Horizon Expierment

- Fixed strongest-user cohort $\mathcal{U}^{\star}$: 16 users, 794K bets, 56K comments
- For each strong user decision round $b_i^{(u)} \in \mathcal{D}_{\text{test}}^{(u)}$, we record $\hat{y} = y_i$ for all methods

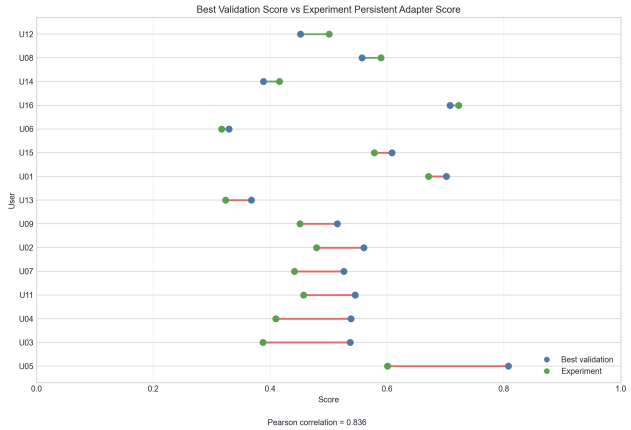
Experiment Summary

- PUL adapter training time: $\sim$250 hours
- Total runtime (excluding PUL training): $\sim$170 hours
- Rounds evaluated: 27,910
- Successful rounds: 98.3% (465 failures)

Training Diagnostics: Most Users Peak Early

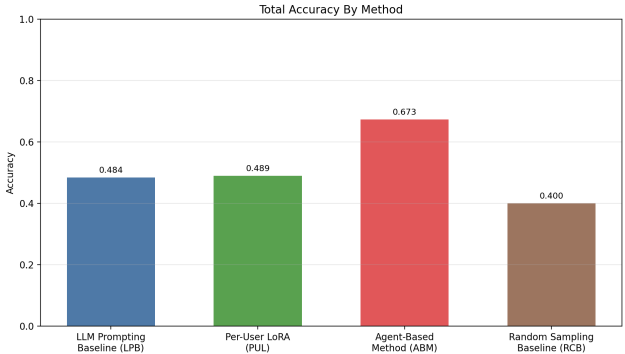


Validation Selection Transfers to Experiment Time



- Best validation score and final experiment score have Pearson correlation $r = 0.836$.
- Average experiment accuracy is only 0.053 lower than the best validation score.
- No user prefers $\lambda = 0$, so every analyzable user benefits from some non-zero adaptation.

Headline Result: The Agent-Based Method Wins

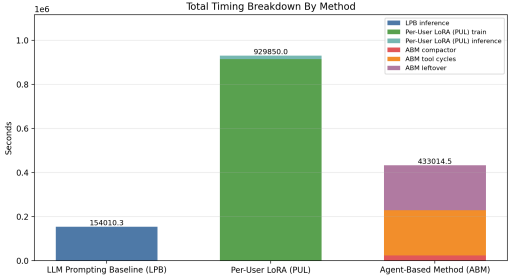


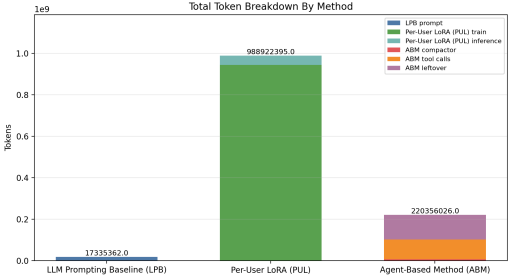
Overall Accuracy

- ABM: 67.3%
- PUL: 48.9%
- LPB: 48.4%
- RSB: 40.0%

Key point. ABM beats PUL by 18.4 points. PUL (Qwen 2.5-7B + retrieval) narrowly outperforms LPB with Gemini Pro.

Resource Tradeoffs Under Full Experiment Accounting





- When training is included, PUL uses ∼**2.2x more wall-clock time** and ∼**4.50x more tokens** than the ABM.
- Most of that excess cost comes from adapter training, not inference.
- Under this accounting, the ABM is both **more accurate** and **less expensive** than PUL.

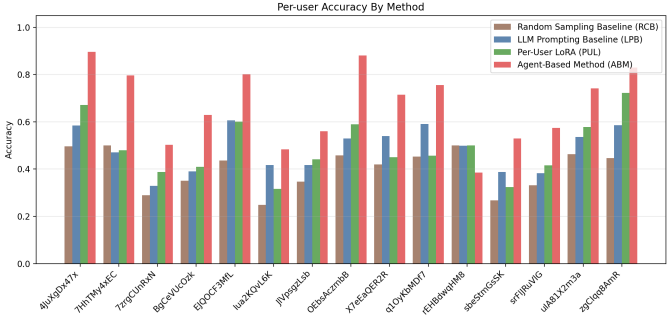
Inference-Only Cost Tells a Different Story

Method		Usage
PUL (with training)	929,850 sec	988.9M tokens
PUL (inference only)	16,211 sec	45.2M tokens
ABM	433,015 sec	220.4M tokens

Key Takeaway

- ABM is $\sim$27x slower at prediction time
- ABM is $\sim$5x more token-intensive
- PUL appears expensive only due to upfront training cost

Per-User Difficulty Is Shared Across Methods



- Average Accuracy Range from 0.406 to 0.717
- Cross-method user correlations stay high:
  - LPB vs. LoRA: $r = 0.788$
  - LPB vs. ABM: $r = 0.721$
  - LoRA vs. ABM: $r = 0.744$

What Makes a User Easier to Predict?

Ridge-regression coefficients

Feature	Coefficient
Binary bets	0.054
Comments	0.041
Total bets	-0.006
Multiple-choice bet share	-0.026

Post-Hoc Difficulty Model

- A four-feature ridge regression reaches $R^2 = 0.773$ and adjusted $R^2 = 0.682$.
- Users concentrated in binary markets are easier to personalize.
- Multiple-choice-heavy users are systematically harder.

Key point. A large part of "user difficulty" is really task composition. The benchmark is harder when a user's history contains more multiple-choice forecasting.

Discussion

Limitations

- **No ABM context control.** Cannot separate agentic retrieval gains from simply providing more context to the model.
- **No clean Qwen baseline.** Not measured in this experiment, though prior results suggest LoRA provides real gains.
  - Prior run: Qwen Baseline 44.1%, Gemini baseline 51.5%, LoRA 53.0%
- **Possible bot contamination.** Filtering removes known bots, but some automated behavior may remain.
- **Incomplete cost comparison.** Measured usage captures runtime and tokens, but not full API costs or training/compute overhead.

Conclusion

Main Result

ABM achieves **67.3% accuracy**, outperforming both PUL and LPB.

- **Inference-time reasoning dominates**: using user-specific context at prediction time outperforms static adapters.
- **Adapters still learn signal**: PUL matches LPB despite a smaller model, with user-specific λ variation.
- **Training is inefficient**: most users peak early, suggesting large cost reductions.
- **Implication**: If an agent-based system can be used to cheaply and accurately simulate user decisions in markets, it could provide an edge in predicting the market itself.

Next Steps for Personalization?

- New frontier models can handle large amounts of context
- Need better ways of identifying when a user gives a personalization signal
- Efficient ways of categorizing, storing, and retrieving these past signals

Expected Direction

I believe if these steps are met, models will see a huge jump in collaboration and usefulness based on becoming more personalized to a user

Learning

New Skills:

- embedding generation
- vector similarity
- retrieval-augmented prompt construction
- open-weight LLM inference
- LoRA adapter training
- distributed GPU work
- remote training/inference orchestration
- constrained tool-call API design
- agent-based context gathering
- memory compaction for bounded prompts

Questions