Aligning LLM Agents by Learning Latent Preference from User Edits

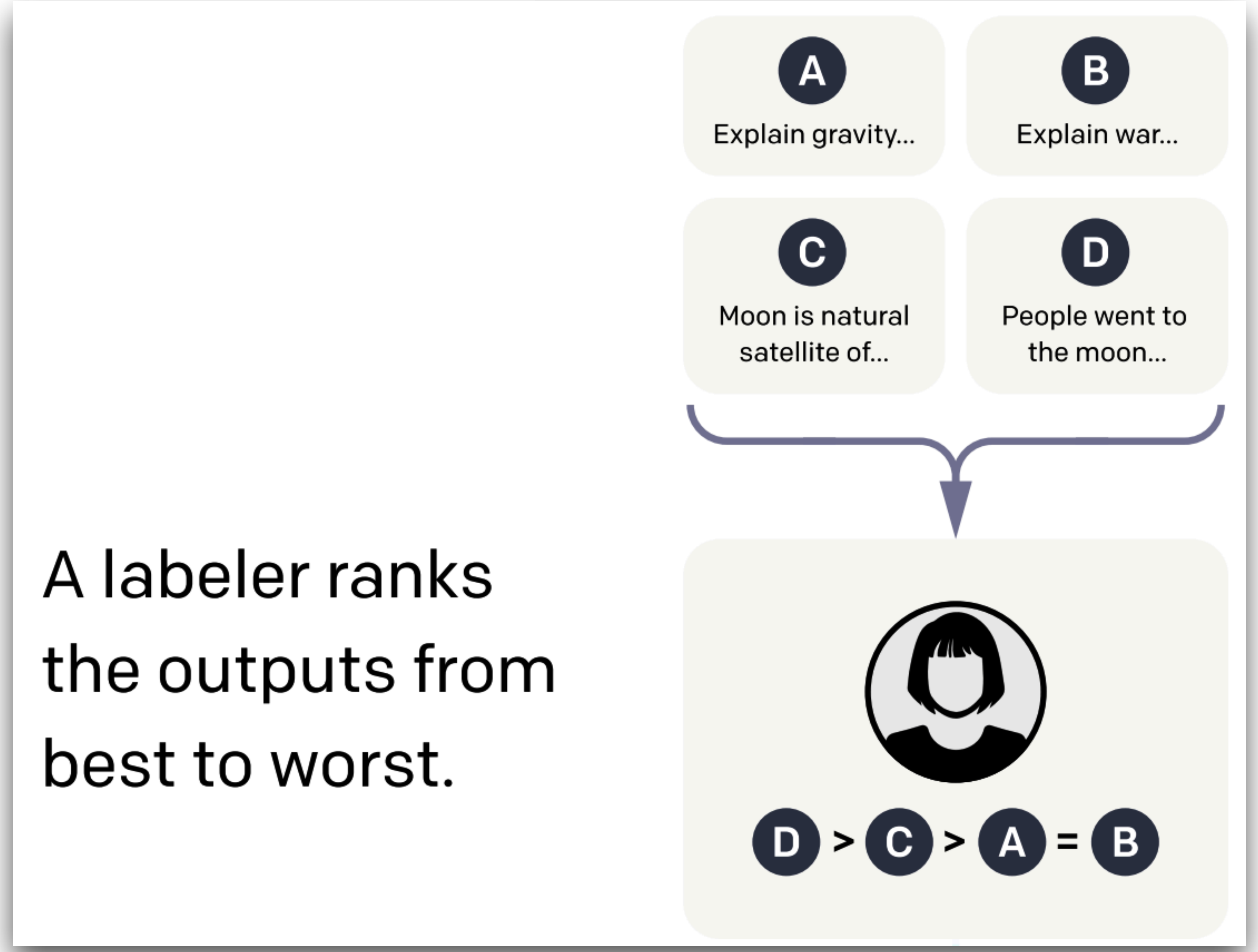
Ge Gao*, Alexey Taymanov*, Eduardo Salinas, Paul Mineiro, Dipendra Misra

Human Feedback

- Learning from human feedback is useful [RLHF, inter alia]

Human Feedback

- Learning from human feedback is useful [RLHF, inter alia]



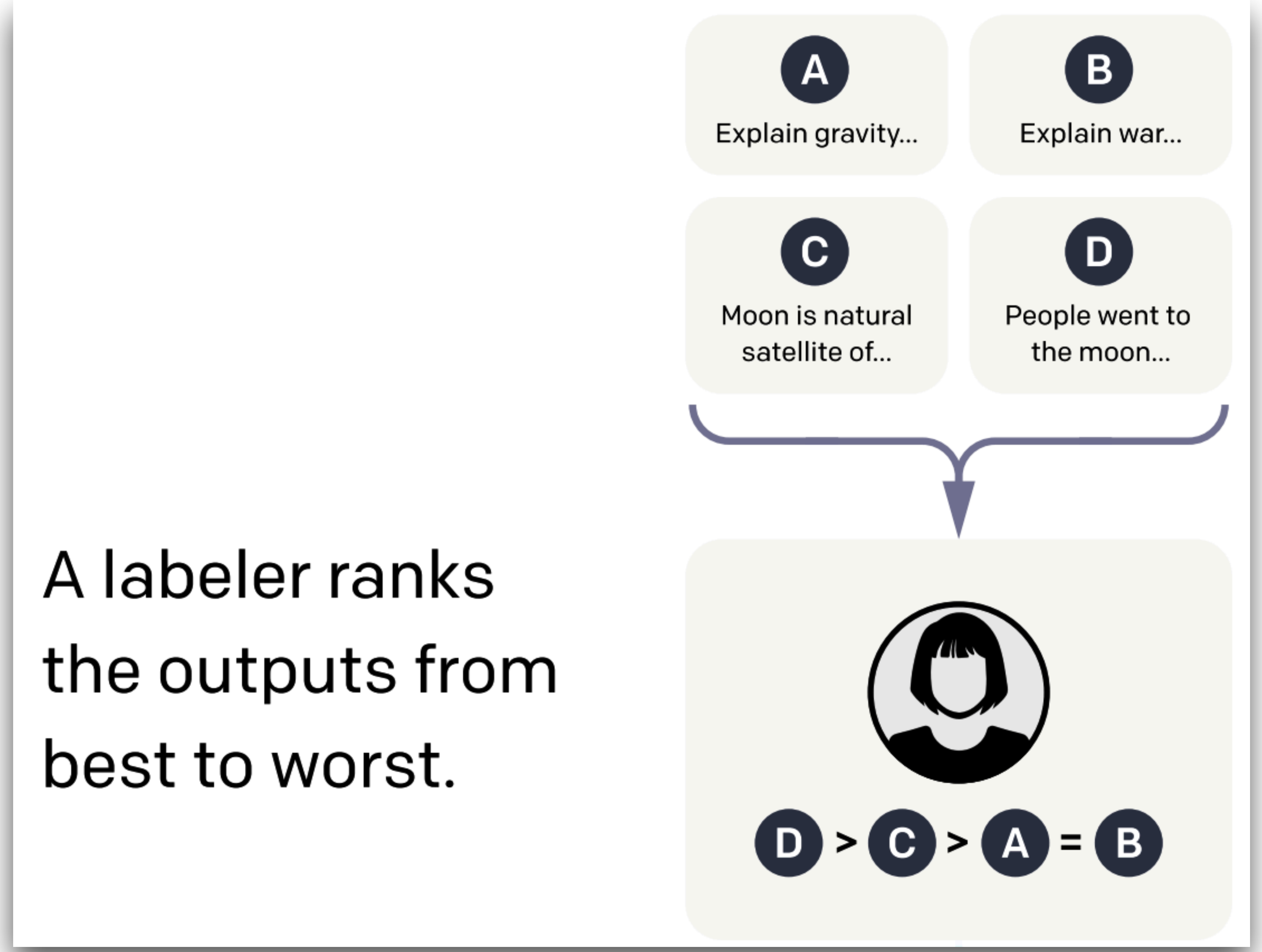
[Figure from Ouyang et. Al, 2022]

Human Feedback

annotator-provided

- Learning from ~~human~~ feedback is useful [RLHF, inter alia]

comparison-based



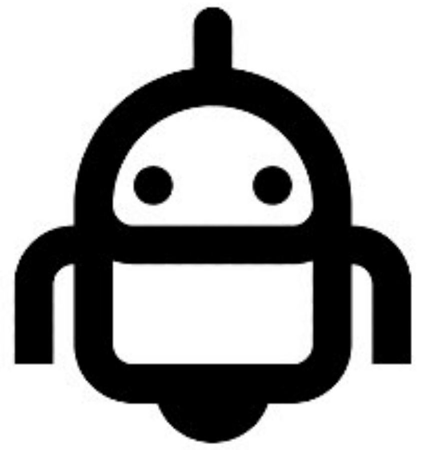
[Figure from Ouyang et. Al, 2022]

Human Feedback

- Learning from ~~human~~ annotator-provided feedback (comparison-based) is useful [RLHF, inter alia]

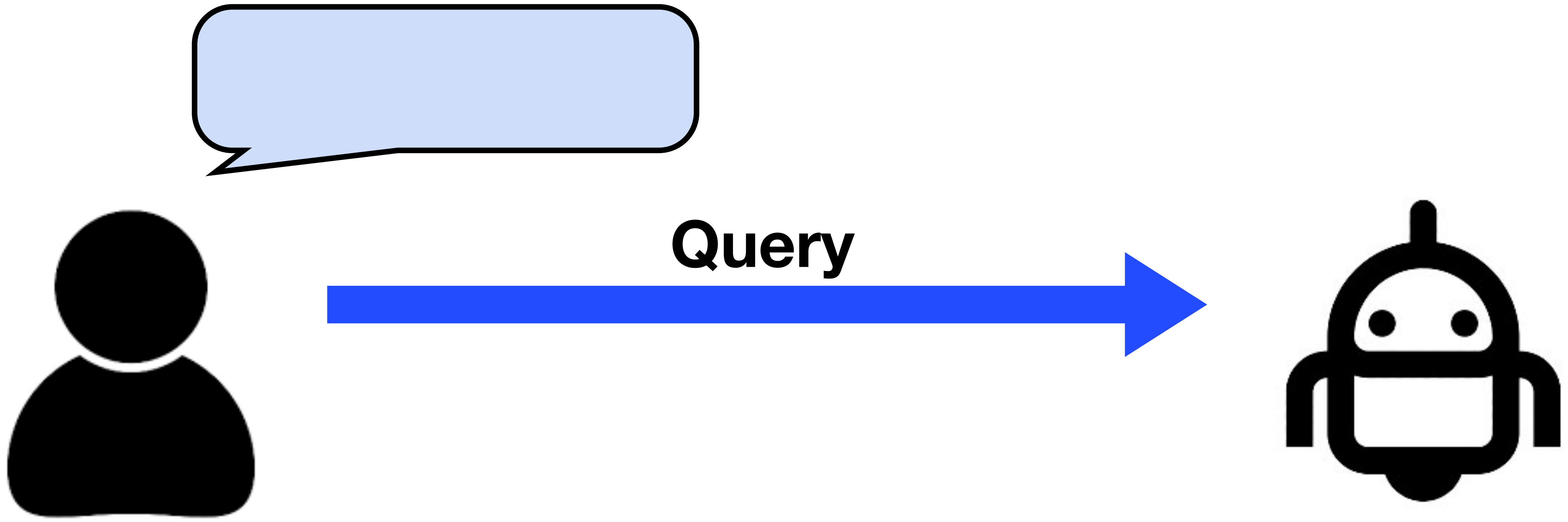
- But…
    - Annotations are expensive to collect
    - Comparison-based feedback rarely occurs in practice

Human Feedback in Practice



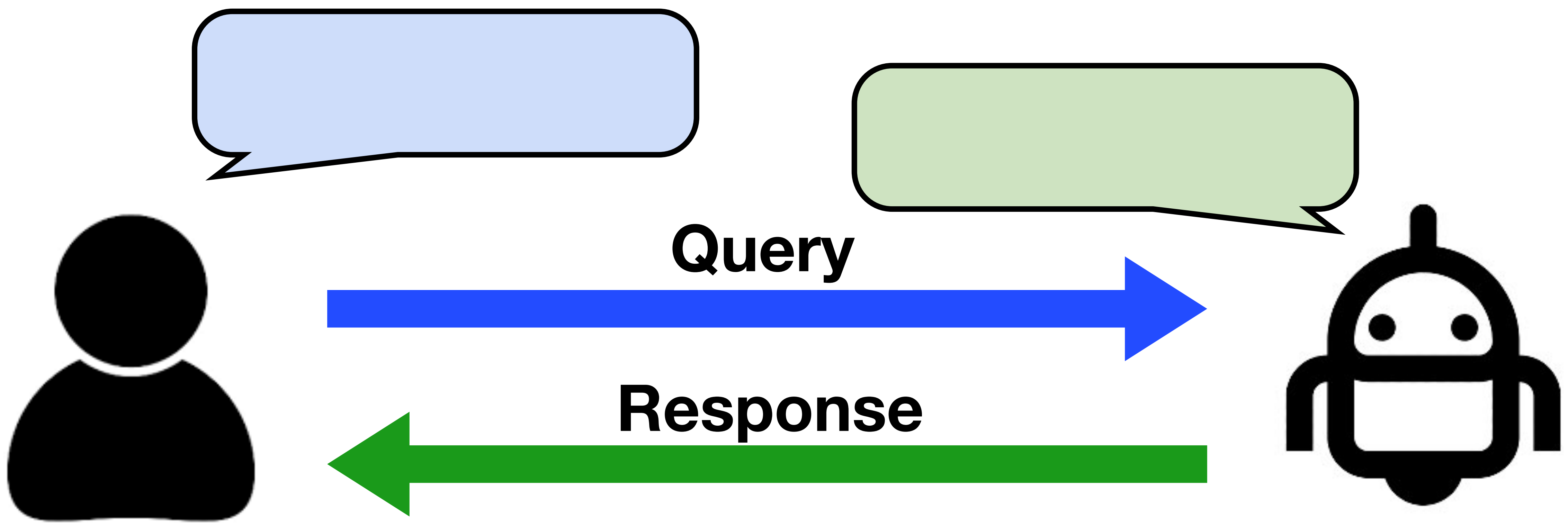
Human Feedback in Practice

- Agent interacts with a user



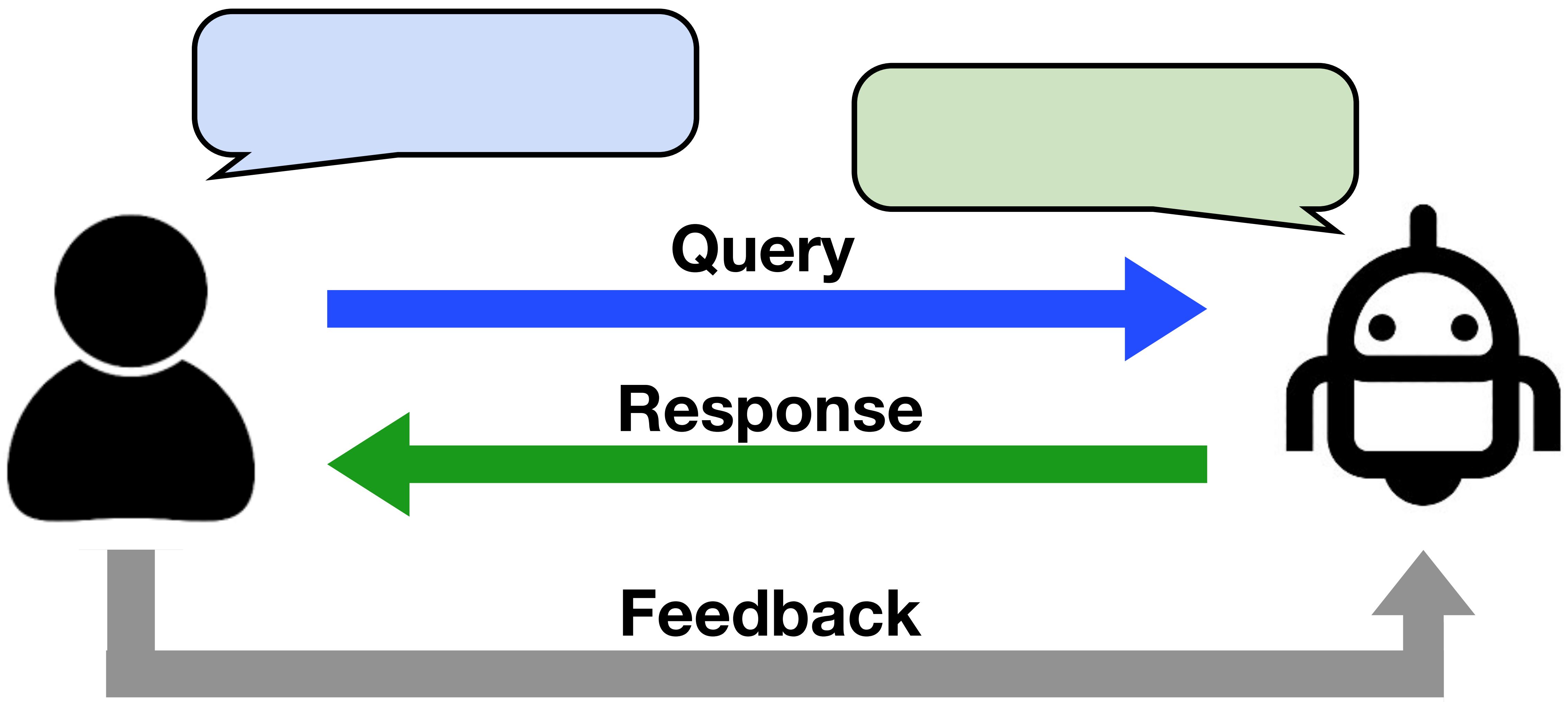
Human Feedback in Practice

- Agent interacts with a <u>user</u>
- Agent provides a <u>single</u> response



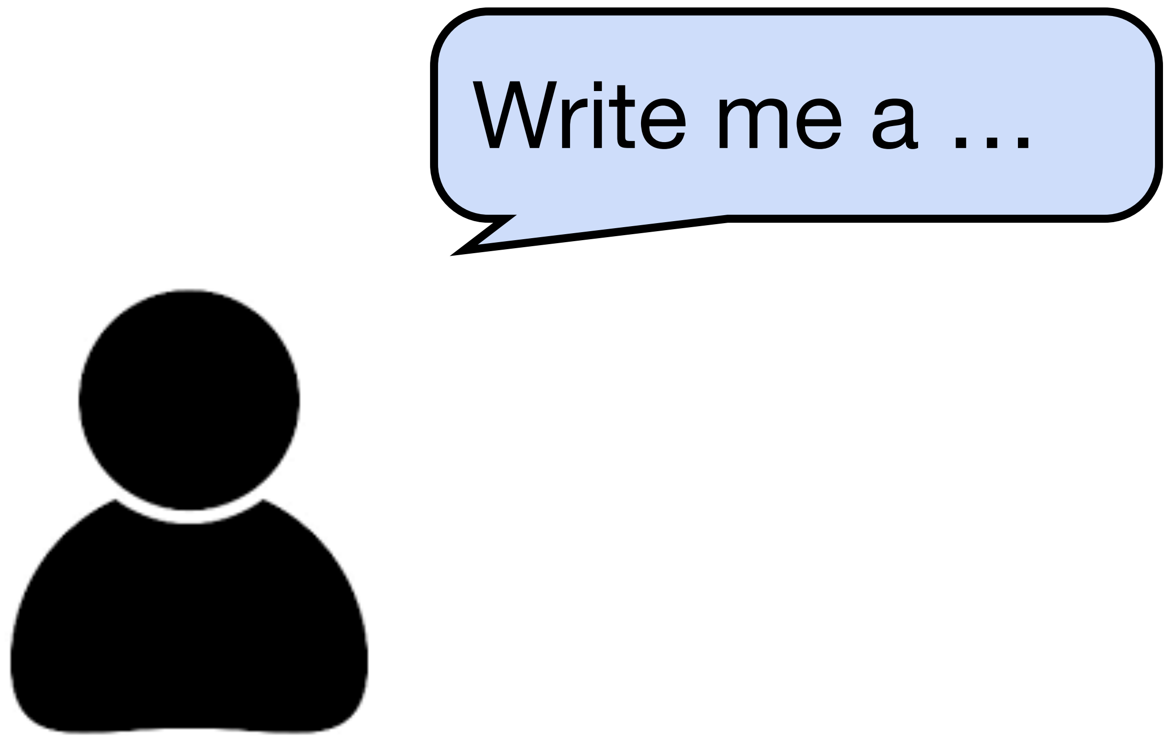
Human Feedback in Practice

- Agent interacts with a user
- Agent provides a single response
- Feedback occurs in various forms
  - Thumb up / down (explicit)
  - User rephrases the query (implicit)
  - ...



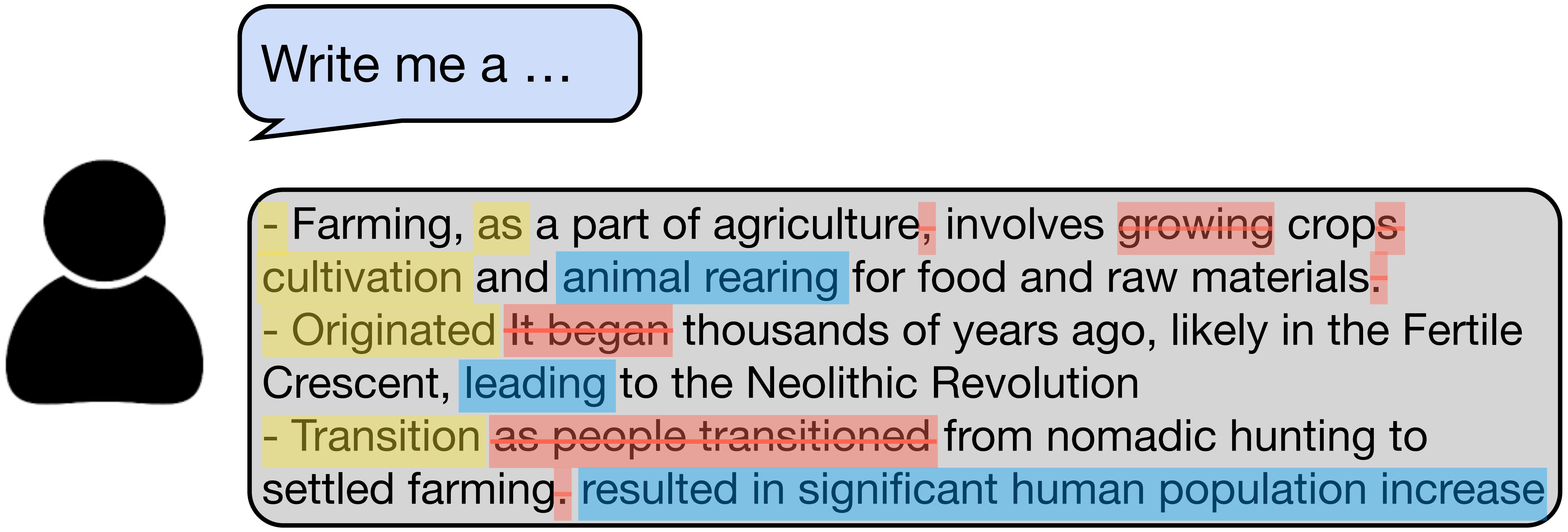
Feedback to Writing Assistant

- The use of AI writing assistants is prevalent nowadays



Feedback to Writing Assistant

- The use of AI writing assistants is prevalent nowadays
- Users often <u>revise</u> the agent response before own final use



Feedback to Writing Assistant

- The use of AI writing assistants is prevalent nowadays
- Users often revise the agent response before own final use

- **Every natural use of the agent yields an edit feedback for learning**

Feedback to Writing Assistant

- The use of AI writing assistants is prevalent nowadays
- Users often revise the agent response before own final use

- **Every natural use of the agent yields an edit feedback for learning**
- Such feedback reflects the user’s authentic expectation and individual preference, beyond the generic writing task

Research Question

- How to learn from user feedback in the form of edits?

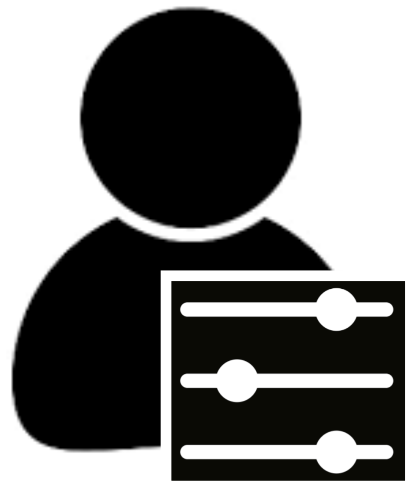
Research Question

- How to learn from user feedback in the form of edits?
  - Infer latent user preference based on edits feedback
  - Generate satisfactory responses that align with the user’s need
  - Take accounts of user’s efforts on making such edits

correct preference induction ⟶ satisfactory response ⟶ few user edits

Challenges

- User preference is multifaceted and complex



- Some preference is context-dependent, and may even vary over time
- Feedback in the form of edits is implicit
    - lacking direct expressions of the underlying preference
    - may lead to diverse interpretations

Outline

- A framework that formulates the interaction process and learning problem
- A method that implements this framework for LLM agents
- Experimental setup, results and analysis

Learning Framework

Round t ① User provides a context x_t to the LLM agent

Article: {user-provided article}
Please summarize the above article.

Learning Framework

Round t

① User provides a context x_t to the LLM agent

Article: {user-provided article}
Please summarize the above article.

② LLM agent generates a response y_t given the context

Farming, a part of agriculture, involves growing crops and rearing animals for food and raw materials. It began thousands of years ago, likely in the Fertile Crescent, and led to the Neolithic Revolution as people transitioned from nomadic hunting to settled farming. This allowed for a significant increase in human population.

Learning Framework

Round t

1. User provides a context x_t to the LLM agent

> Article: {user-provided article}
> Please summarize the above article.

2. LLM agent generates a response y_t given the context

> Farming, a part of agriculture, involves growing crops and rearing animals for food and raw materials. It began thousands of years ago, likely in the Fertile Crescent, and led to the Neolithic Revolution as people transitioned from nomadic hunting to settled farming. This allowed for a significant increase in human population.

3. User edits the agent response, resulting in a revision y'_t

> - Farming, as a part of agriculture~~,~~ involves ~~growing~~ crop~~s~~ cultivation and animal rearing for food and raw materials~~.~~
> - Originated ~~It began~~ thousands of years ago, likely in the Fertile Crescent, leading to the Neolithic Revolution
> - Transition ~~as people transitioned~~ from nomadic hunting to settled farming~~.~~ resulted in significant human population increase

Learning Framework

Round t

1. User provides a context x_t to the LLM agent
2. LLM agent generates a response y_t given the context
3. User edits the agent response, resulting in a revision y'_t

Learning Framework

Round t

1. User provides a context x_t to the LLM agent
2. LLM agent generates a response y_t given the context
3. User edits the agent response, resulting in a revision y'_t according to a latent preference f^*_t

Learning Framework

Round t

1. User provides a context x_t to the LLM agent
2. LLM agent infers a preference f_t based on history given the context

   LLM agent generates a response y_t given the context
3. User edits the agent response, resulting in a revision y'_t ,

   according to a latent preference f^*_t

Learning Framework

Round t

1. User provides a context x_t to the LLM agent

2. LLM agent infers a preference f_t based on history given the context

   LLM agent generates a response y_t given the context

3. User edits the agent response, resulting in a revision y'_t ,

   according to a latent preference f^*_t

Minimize cumulative cost $\sum_{t=1}^{T} c_t$

Agent incurs a cost $c_t = \Delta_{\text{edit}}(y_t, y'_t)$

Learning Framework

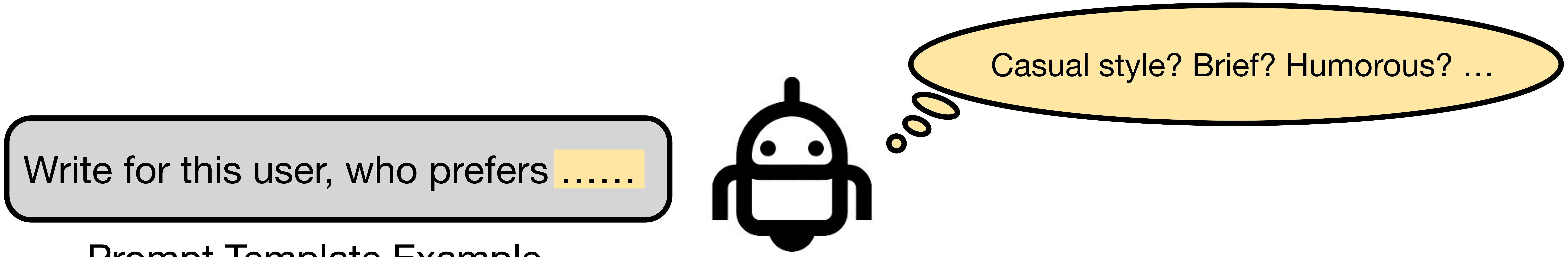
- We formulate the interaction progress and preference learning problem as **PRELUDE** (**PRE**ference **L**earning from **U**ser's **D**irect **E**dits)
    - Assume that the user directly makes edits to the agent response based on a latent preference
    - Agent infers a user preference from the interaction history, and uses it to generate a response
    - Cost minimization to account for the amount of efforts spent by the user on making edits

Method

- Agent leverages LLMs by prompting
- We learn a <u>prompt policy</u> that can infer a descriptive user preference, and then use it in the prompt to directly drive the response generation

Method

- Agent leverages LLMs by prompting
- We learn a prompt policy that can infer a descriptive user preference, and then use it in the prompt to directly drive the response generation



Prompt Template Example

Method

- Agent leverages LLMs by prompting
- We learn a prompt policy that can infer a descriptive user preference, and then use it in the prompt to directly drive the response generation
    - When user makes edits, induce a description of the user preference
    - Manage a collection of preference history
    - Given a new context, infer a descriptive preference based on retrieving similar contexts from the history

Method

Round t

1. User provides a context x_t to the LLM agent
2. LLM agent infers a preference f_t based on history given the context

   LLM agent generates a response y_t given the context and inferred preference
3. User edits the agent response, resulting in a revision y'_t , according to a latent preference f^*_t

Agent incurs a cost $c_t = \Delta_{\text{edit}}(y_t, y'_t)$

Method

Round t

1. User provides a context x_t to the LLM agent

Retrieve top-k examples from history based on $\phi(x_t)$
Aggregate induced preferences in those retrieved examples $\{\tilde{f}_{z_i}\}_{i=1}^{k}$

2. LLM agent infers a preference f_t based on history given the context

LLM agent generates a response y_t given the context and inferred preference

3. User edits the agent response, resulting in a revision y_t' ,
according to a latent preference f_t^*

Agent incurs a cost $c_t = \Delta_{\text{edit}}(y_t, y_t')$

Method

Round t

1. User provides a context x_t to the LLM agent

Retrieve top-k examples from history based on $\phi(x_t)$
Aggregate induced preferences in those retrieved examples $\{\tilde{f}_{z_i}\}_{i=1}^{k}$

2. LLM agent infers a preference f_t based on history given the context

LLM agent generates a response y_t given the context and inferred preference

3. User edits the agent response, resulting in a revision y'_t ,
according to a latent preference f^*_t

Agent incurs a cost $c_t = \Delta_{\text{edit}}(y_t, y'_t)$

- $c_t \geq \delta$: LLM agent generates a preference $\tilde{f}_t$ to explains the user edits
- $c_t < \delta$: $\tilde{f}_t \leftarrow f_t$

Method

Round t

1. User provides a context x_t to the LLM agent

Retrieve top-k examples from history based on $\phi(x_t)$
Aggregate induced preferences in those retrieved examples $\{\tilde{f}_{z_i}\}_{i=1}^{k}$

2. LLM agent infers a preference f_t based on history given the context

LLM agent generates a response y_t given the context and inferred preference

3. User edits the agent response, resulting in a revision y'_t ,
according to a latent preference f^*_t

Agent incurs a cost $c_t = \Delta_{\text{edit}}(y_t, y'_t)$

$c_t \geq \delta$: LLM agent generates a preference $\tilde{f}_t$ to explains the user edits

$c_t < \delta$: $\tilde{f}_t \leftarrow f_t$

History $D \leftarrow D \cup \{(\phi(x_t), \tilde{f}_t)\}$

Method

- **CIPHER** (**C**onsolidates **I**nduced **P**references based on **H**istorical **E**dits with **R**etrieval)
- Computationally efficient
    - 4 LLM calls at max per interaction; only a small increase in prompt length
    - Low memory storage: save context representation instead of the context itself
- User-friendly and interpretable
    - Users are not required to do heavy prompt engineering
    - Users could read and understand the preference learned by the agent

Task & User Setup

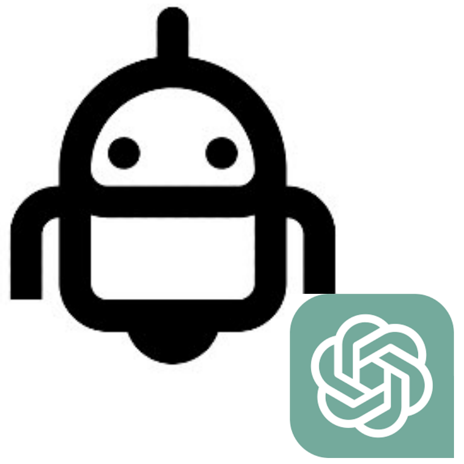
- Writing task for the agent: summarize a document
- GPT-4 user as a simulation 
    - Provide a context (i.e., specify the writing task, includes a document)
    - Can provide context documents from different sources
    - Have context-depend preference for different use cases

Task & User Setup

Use Case	Latent User Preference	Doc Source
Introduce a political news to kids	targeted to young children, storytelling, short sentences, playful language, interactive, positive	News article
Promote a paper to invoke more attention and interests	tweet style, simple English, inquisitive, skillful foreshadowing, with emojis	Paper abstract
Take notes for factual knowledge	bullet points, parallel structure, brief	Wikipedia page
Use online stories to inspire character developments in creative writing	second person narrative, brief, show emotions, invoke personal reflection, immersive	Reddit post
Extract main opinions from a review	question answering style	Movie review

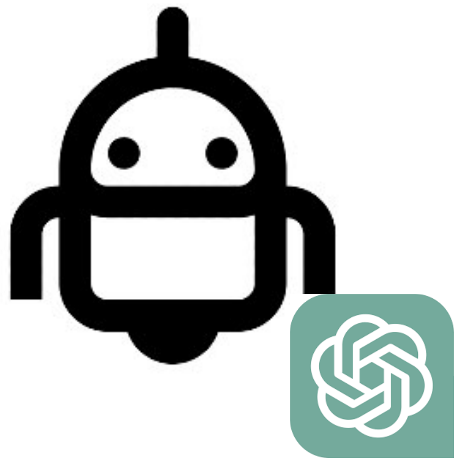
Experimental Setup

- 200 interactions in total $T = 200$; different context per round
- Implementation details of CIPHER
  - GPT-4 as the base LLM
  - MPNeT as the context representation function $\phi = \mathrm{MPNet}$
  - Top 5 retrieval with cosine similarity $k = 5$



Experimental Setup

- 200 interactions in total $T = 200$; different context per round
- Implementation details of CIPHER
  - GPT-4 as the base LLM
  - MPNeT as the context representation function $\phi = \mathrm{MPNet}$
  - Top 5 retrieval with cosine similarity $k = 5$



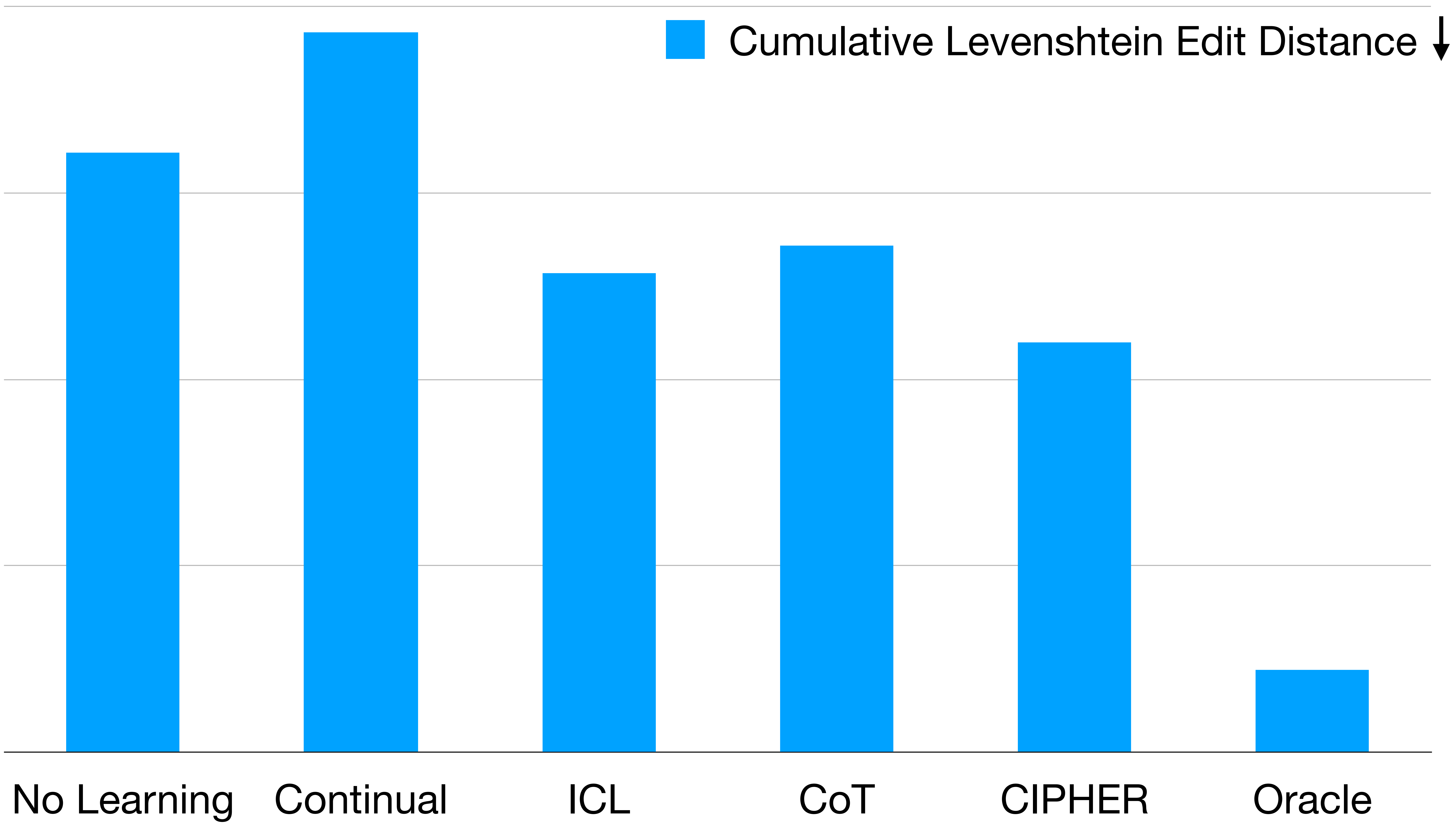
- Evaluation metrics
  - Cumulative Levenshtein edit distance: removal, insertion, or substitution (BPE tokens)
  - Expense of using LLM: total number of input and output BPE tokens

Experimental Setup

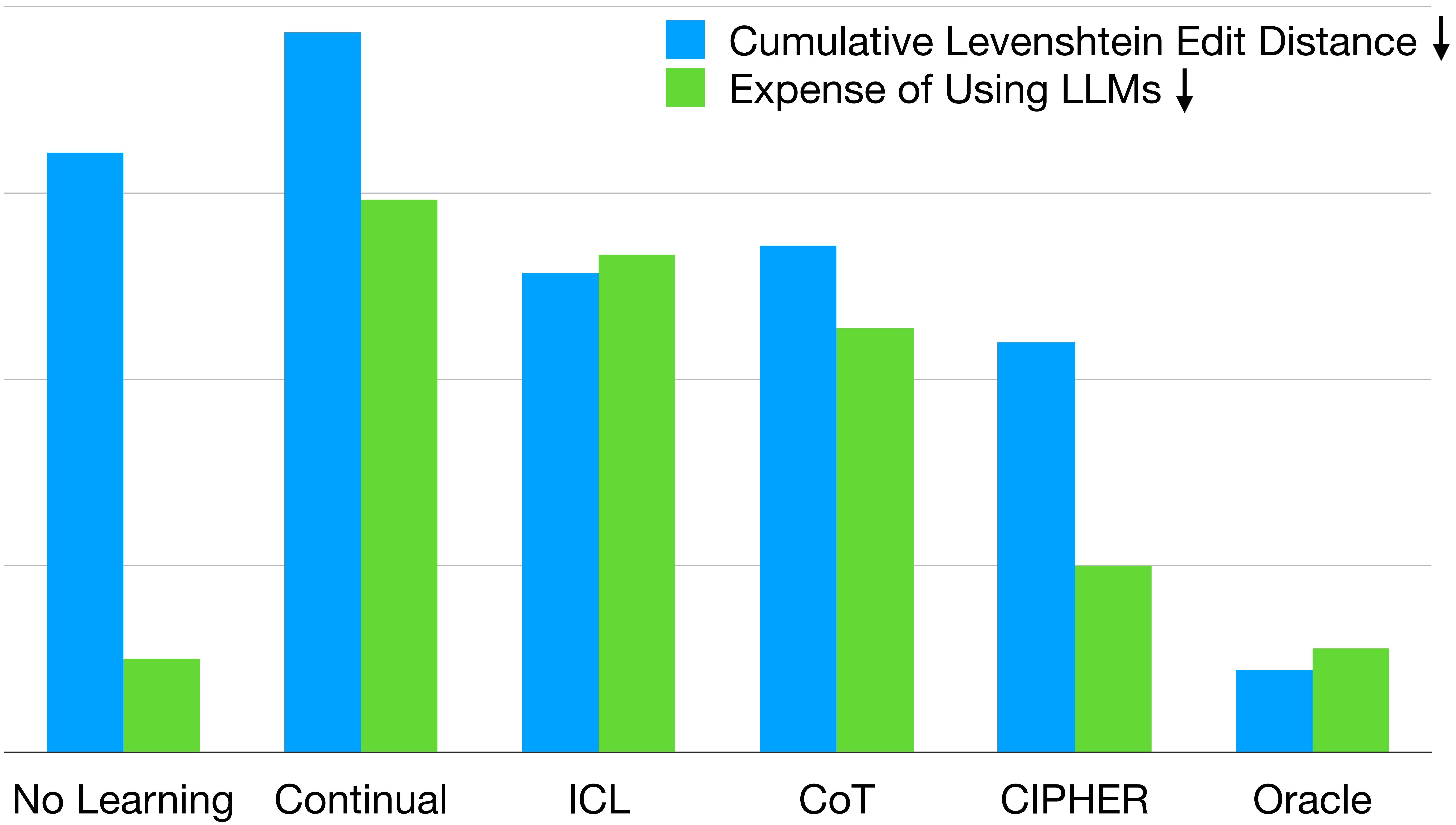
- Comparison systems

	Interpretable	Retrieval
• No Learning: does not perform any preference learning		
• Continual Learning: infer a preference using the most recent k interactions	✓	
• In-Context Learning: retrieve top k historical examples, and use them as demonstration examples in the prompt for response generation		✓
• Chain-of-Thought: the prompt for response generation specifies two steps: 1) infer a descriptive user preference based on retrieved top k examples, and 2) generate a response accordingly	✓	✓
• Oracle: let the agent use the true latent preference to generate a response		

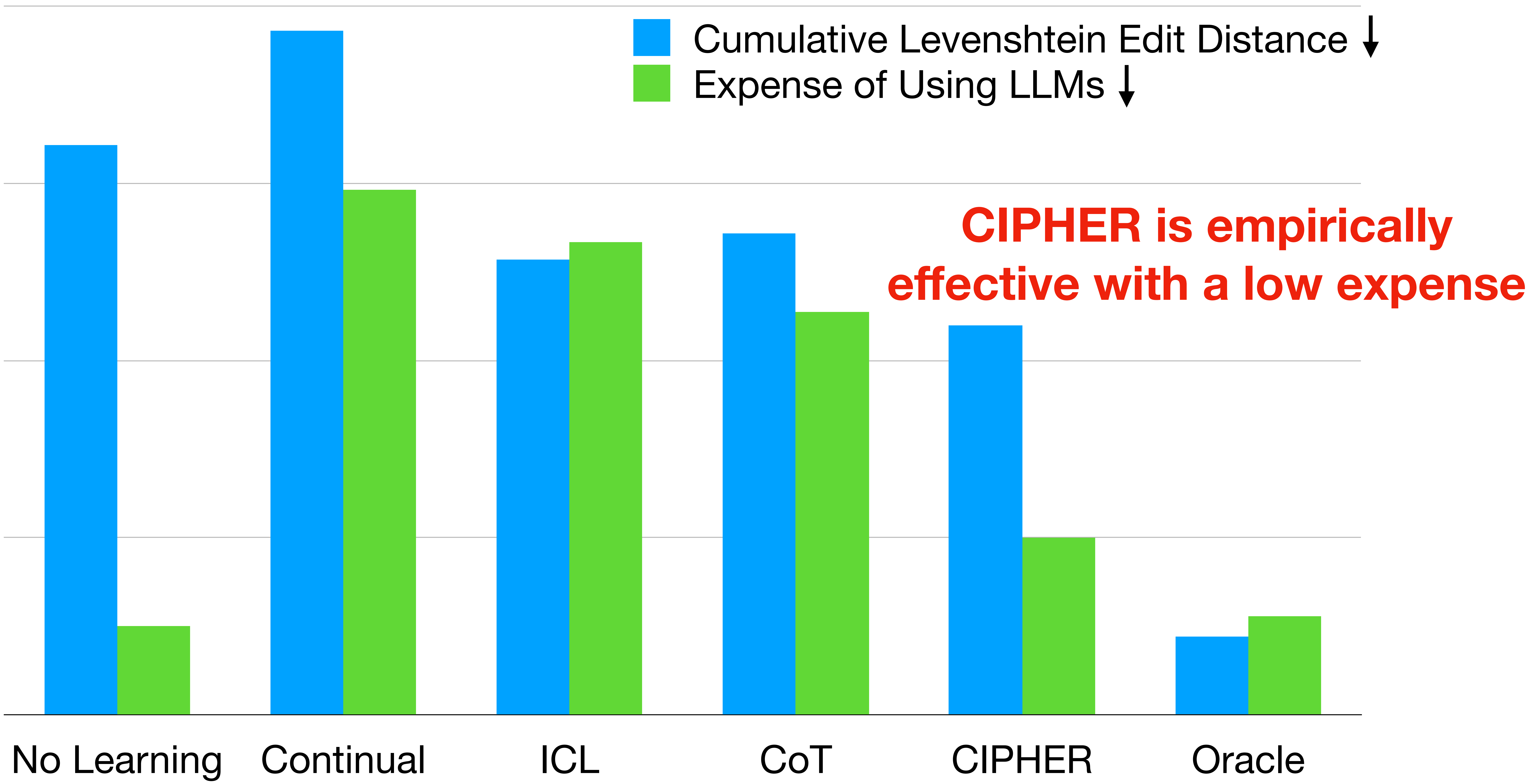
Experimental Result



Experimental Result



Experimental Result

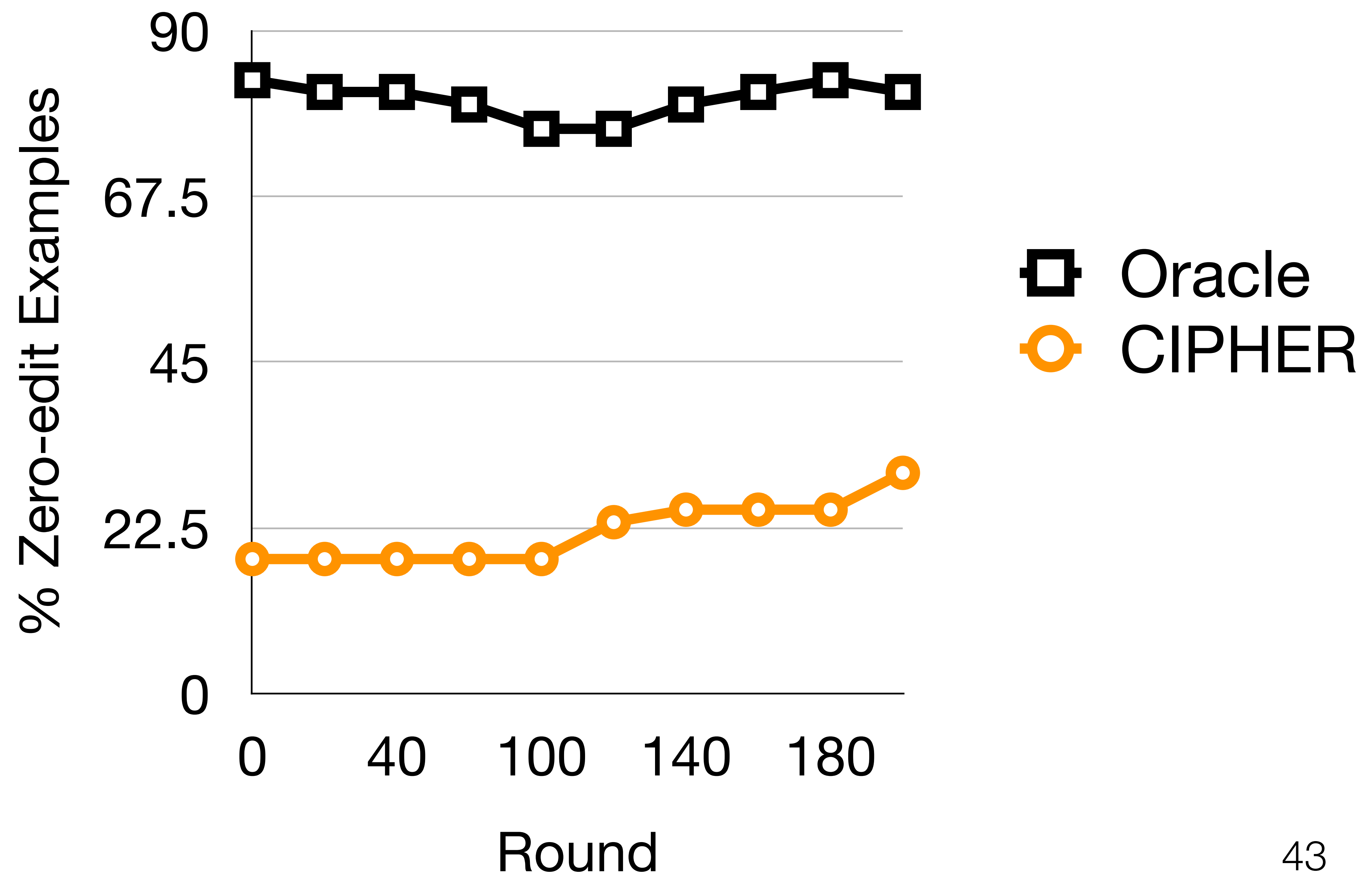


Experimental Analysis

- Does the user make fewer edits to CIPHER over time?

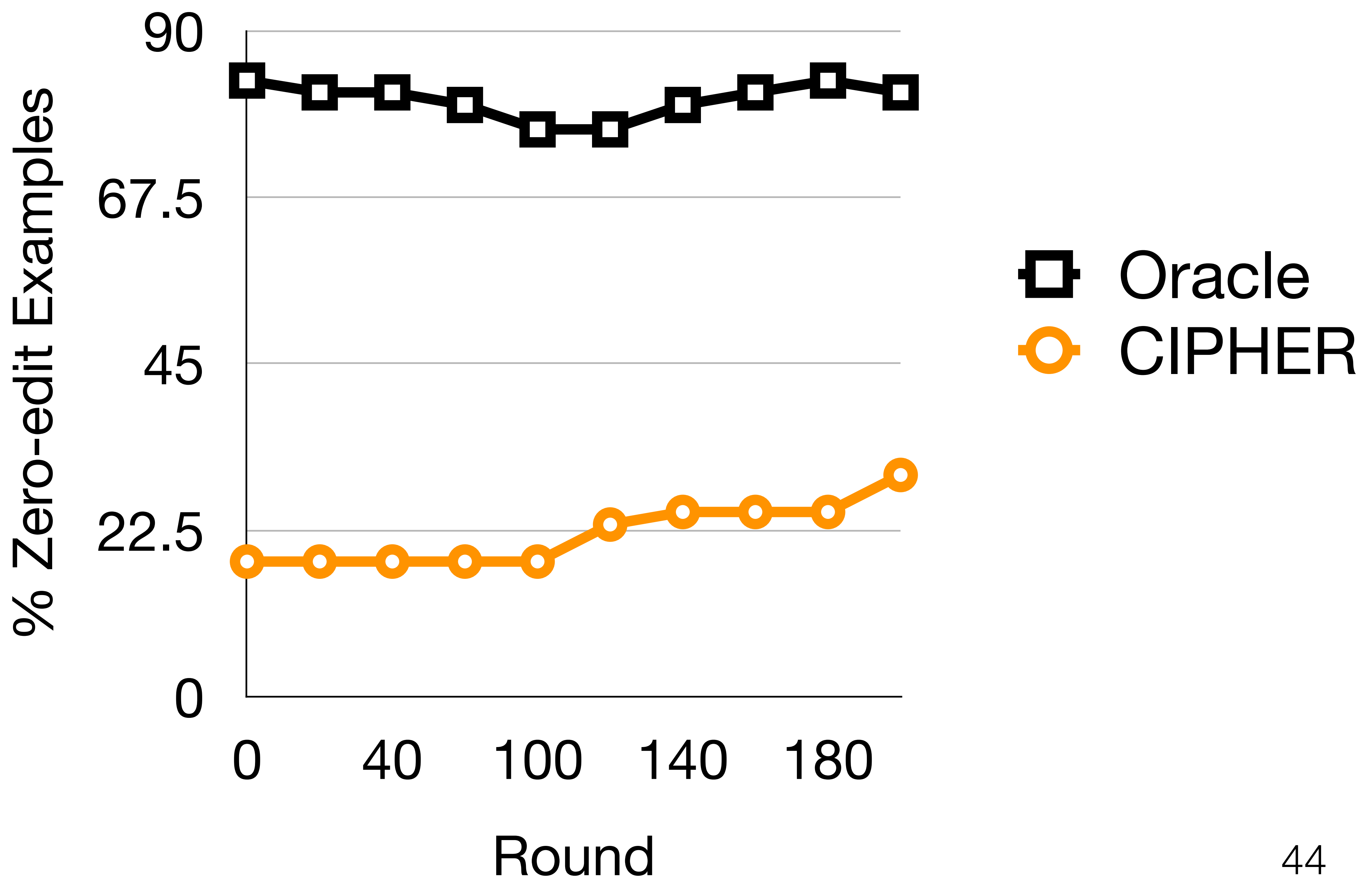
Experimental Analysis

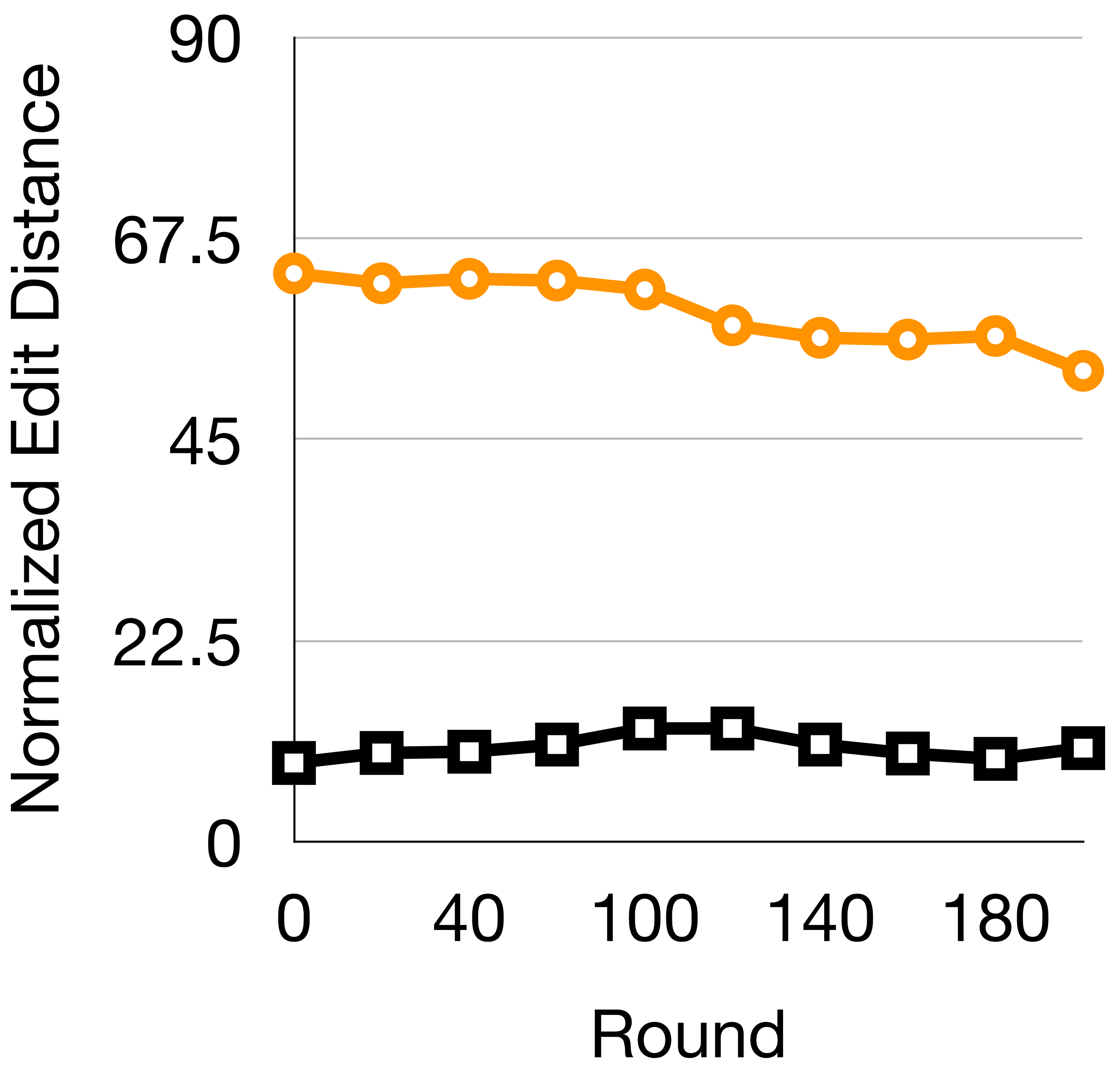
- Does the user make fewer edits to CIPHER over time?
  - Percentage of the zero-edit examples (binned per 20 rounds) ↑



Experimental Analysis

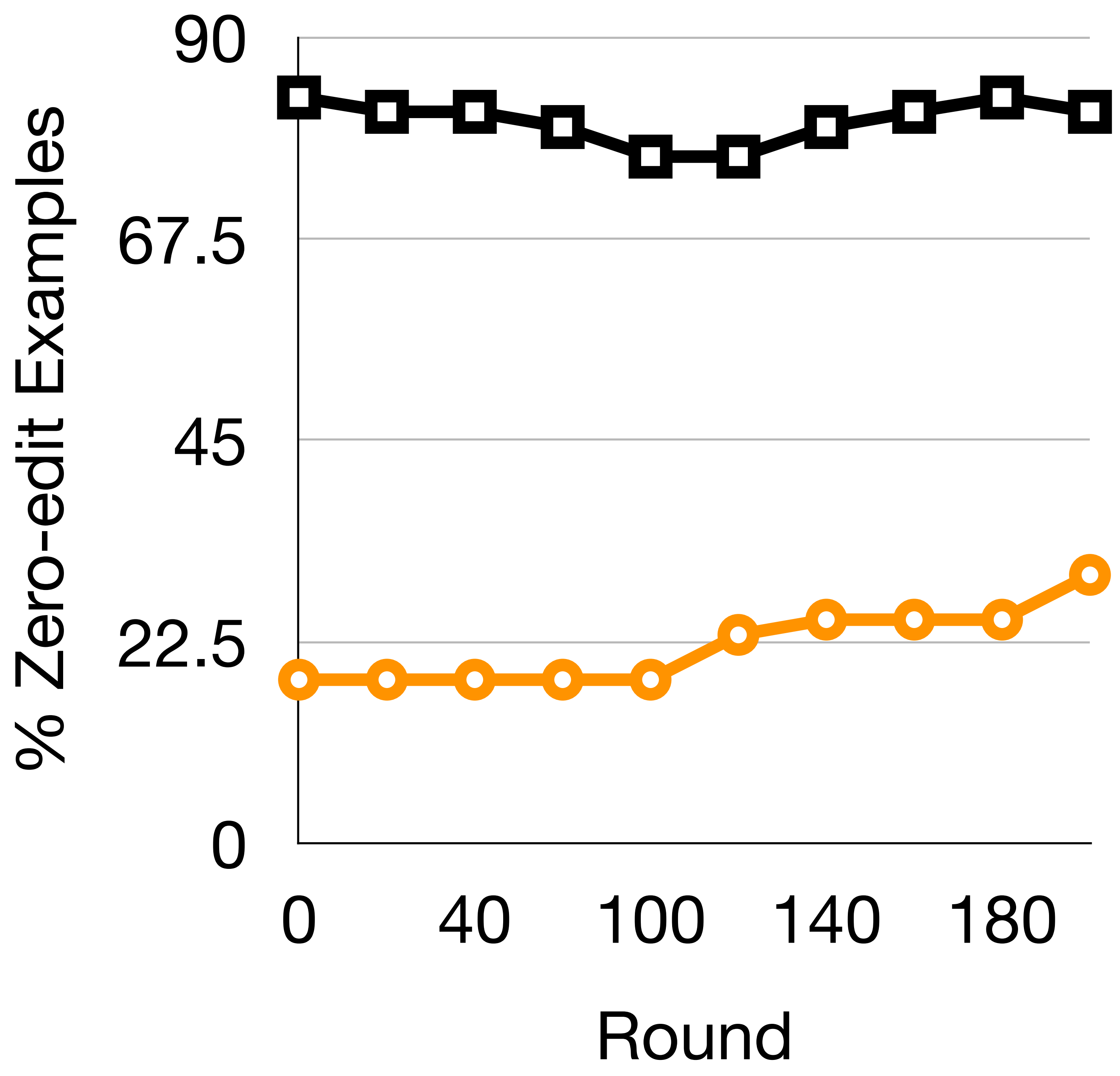
- Does the user make fewer edits to CIPHER <u>over time</u>?
  - Percentage of the zero-edit examples (binned per 20 rounds) ↑
  - Edit distance normalized by the response length (averaged per 20 rounds) ↓

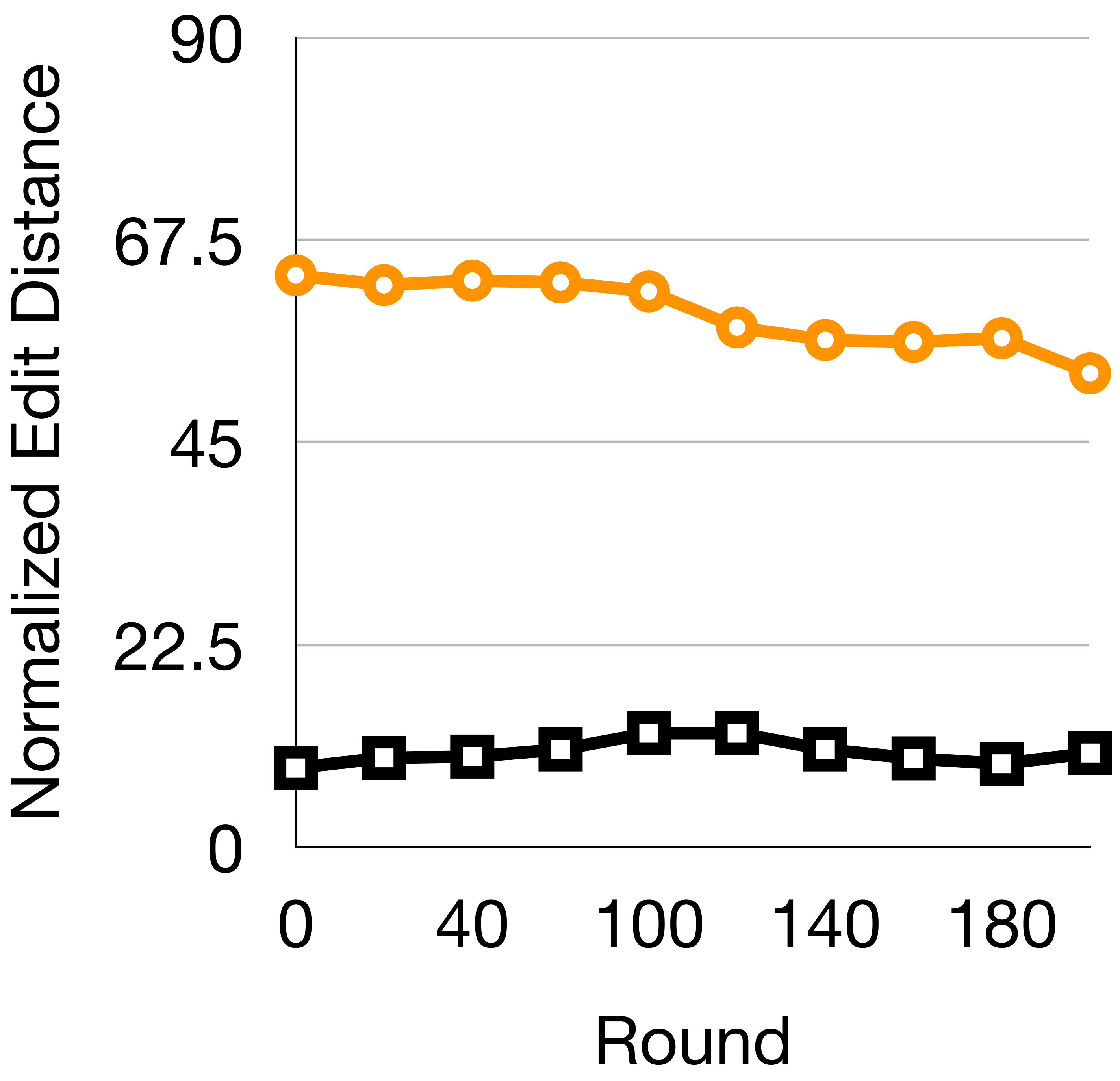




Experimental Analysis

- Does the user make fewer edits to CIPHER <u>over time</u>? **Yes!**
    - Percentage of the zero-edit examples (binned per 20 rounds) ↑
    - Edit distance normalized by the response length (averaged per 20 rounds) ↓





Experimental Analysis

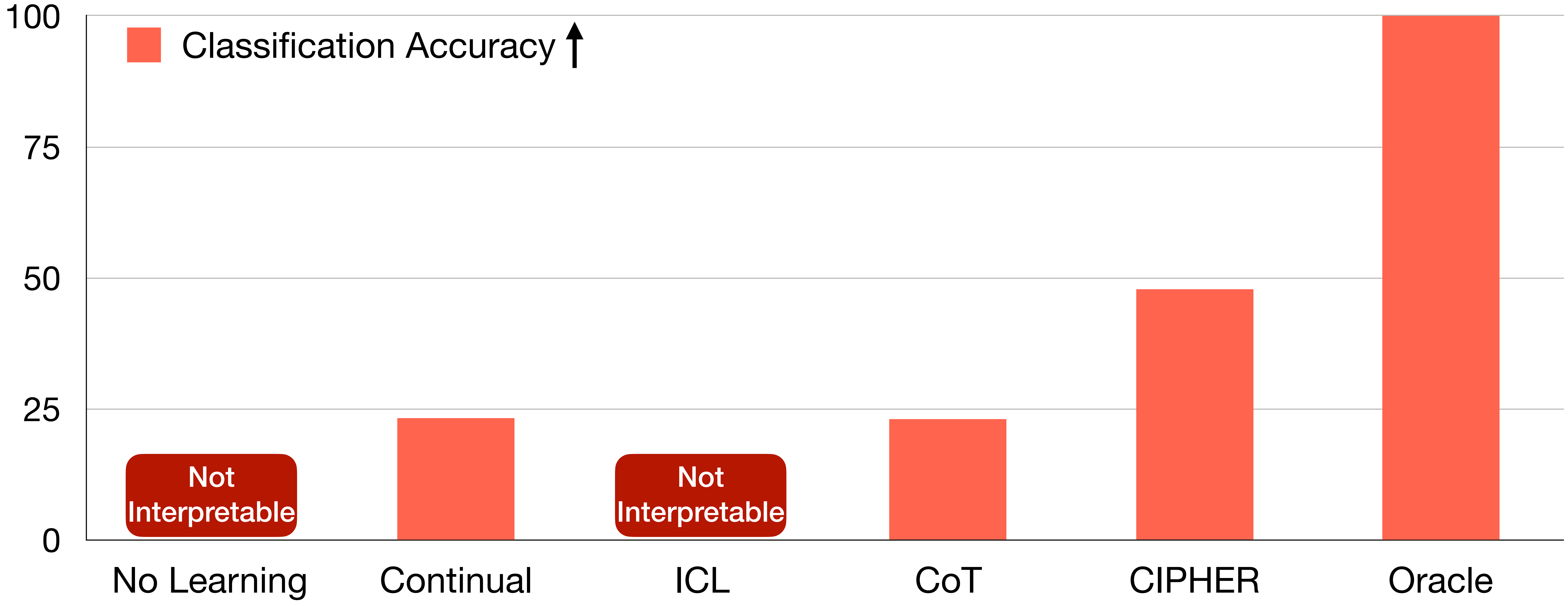
- How is the quality of the learned preference by CIPHER?
- We conduct two types of evaluation:
    1. Automatic analysis based on similarity measures
    2. Human evaluation

Experimental Analysis

- This analysis assumes access to all the latent user preference across different context
- Is the preference learned by CIPHER most similar to the correct latent preference?

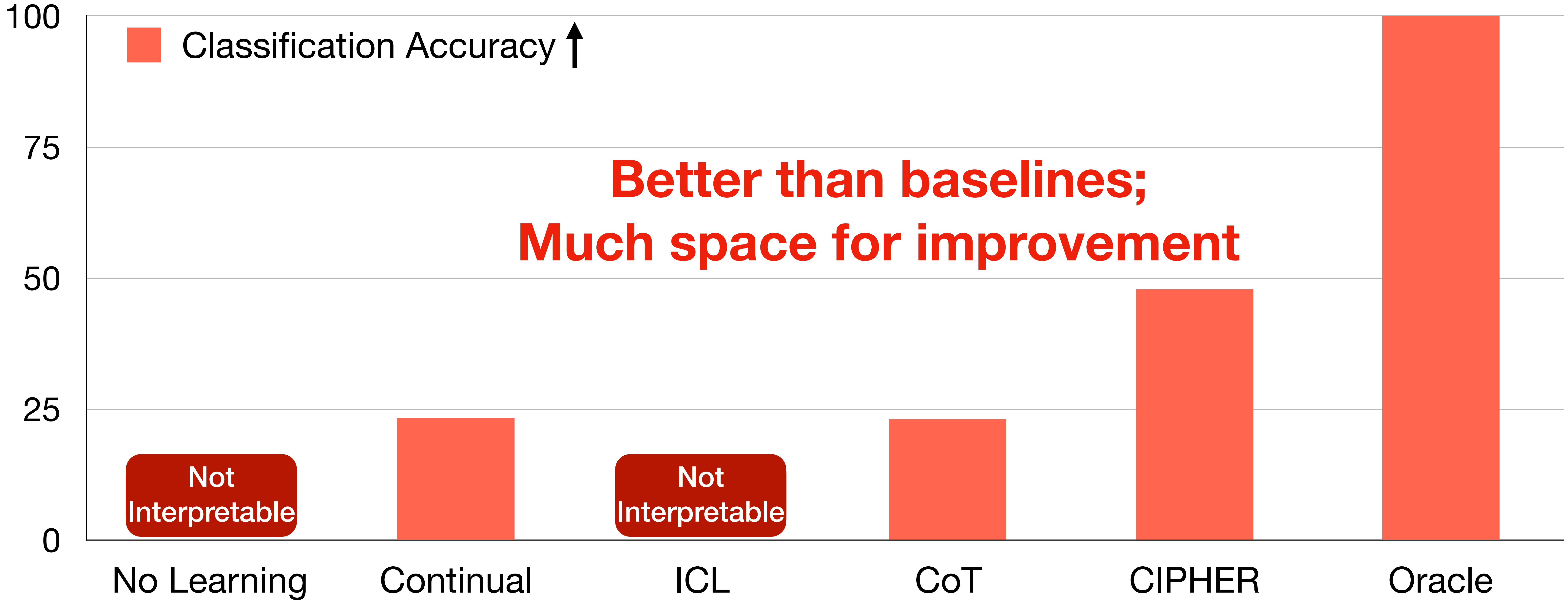
Experimental Analysis

- This analysis assumes access to all the latent user preference across different context
- Is the preference learned by CIPHER most similar to the correct latent preference?



Experimental Analysis

- This analysis assumes access to all the latent user preference across different context
- Is the preference learned by CIPHER most similar to the correct latent preference?



Human Evaluation

- Win Rate Evaluation: pairwise comparison by 7 human evaluators
    - CIPHER vs ICL: 73.3%
    - CIPHER vs Oracle: 23.7%

Human Evaluation

- Win Rate Evaluation: pairwise comparison by 7 human evaluators
  - CIPHER vs ICL: 73.3%
  - CIPHER vs Oracle: 23.7%
- Edits by Human Users: averaged results from 3 human evaluators

	CIPHER	Oracle
Cumulative Edit Distance ↓	211	98
% Zero-edit Examples ↑	60%	76.7%

Summary

- We study learning from human feedback in the form of user edits
- **PRELUDE** framework formulates the interaction progress and preference learning as a cost minimization problem
- **CIPHER** method learns a prompt policy to infer a descriptive user preference
    - computationally efficient, user-friendly, interpretable
    - empirically effective with a low expense
- More in the paper: email writing task, more baselines, qualitative analysis …

Check Out Our Codebase!

- https://github.com/gao-g/prelude
- Modularized codebase designed for easy customization
- Detailed instructions on how to:
  - Add your own task
  - Specify your own user
  - Implement your own agent