Self-Refine: Iterative Refinement with Self-Feedback

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Bodhisattwa Prasad Majumder, Shashank Gupta, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, Peter Clark

1. Idea

Generate a response that is 1) **Relevant** - The response addresses the context, 2) **Informative** -, 3) **Interesting** - The response is interesting, 4) **Factual**...

User: *What's your favorite activity?*
Response: *I like reading.*

User: *Why?*
Response: *That's just the way it is*

Feedback

Rate the above response on 1) **Relevance** - The response addresses the context, 2) **Informativeness** ..., 3) **Engagement** ... 4) **Factuality**...
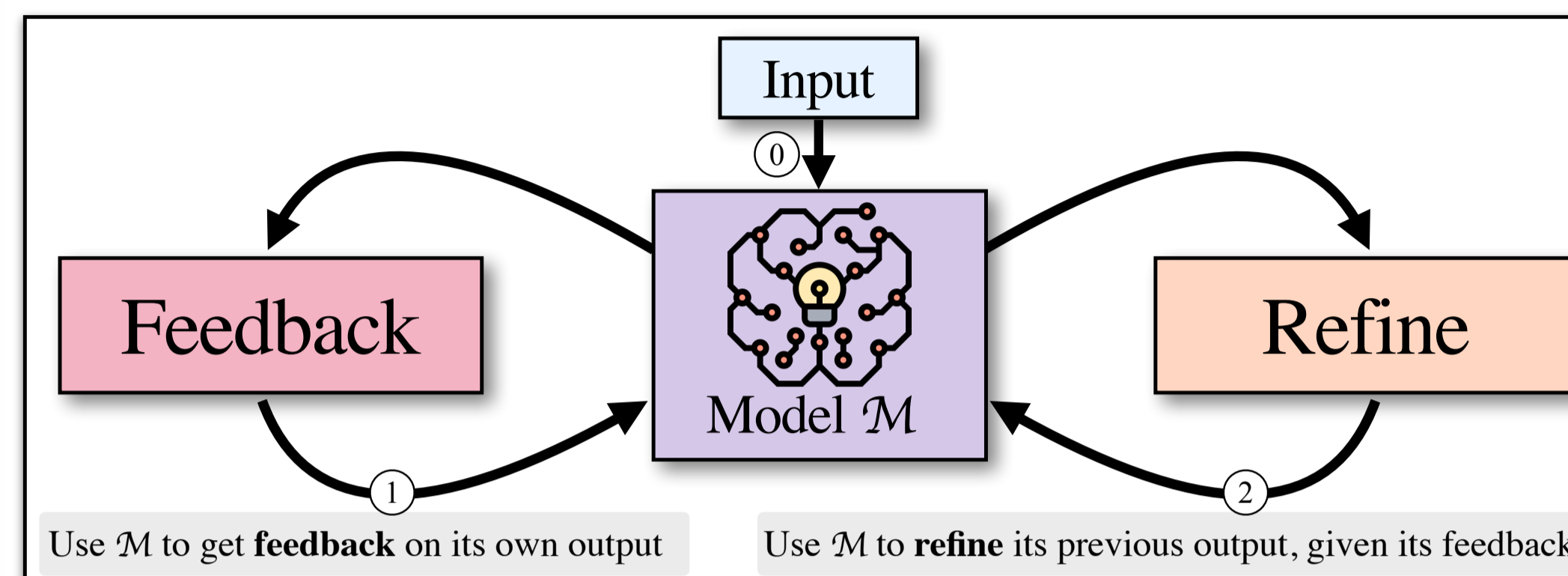
Feedback: The response is rather brief not engaging, but could be factual. Relevance 2/3, Informativeness 1/3, Engagement 3/3, Factuality 3/3
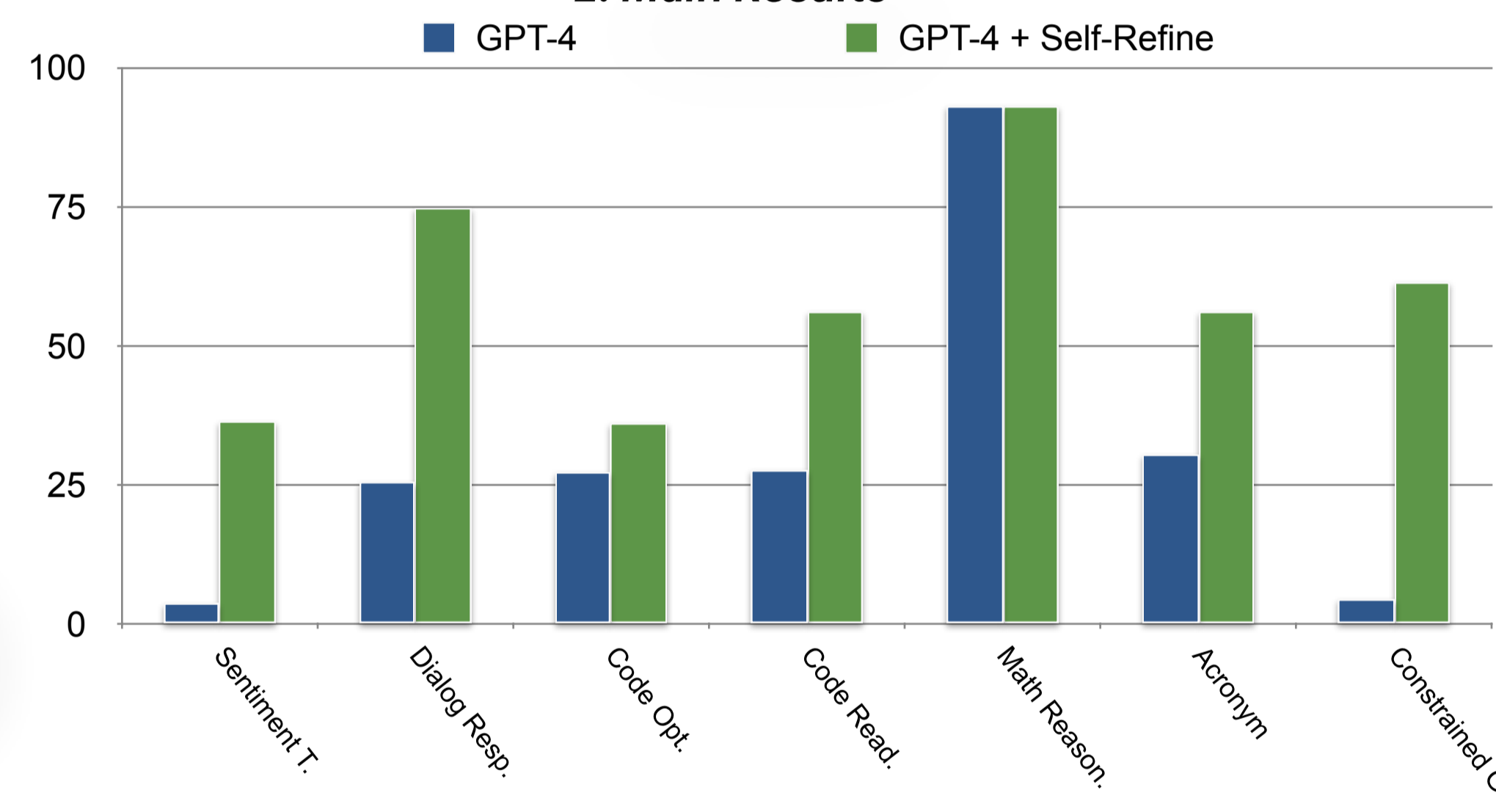
Refine

Use the feedback to improve the response.

Refined response: Reading is my favorite activity because it allows me to continuously learn and explore new ideas.
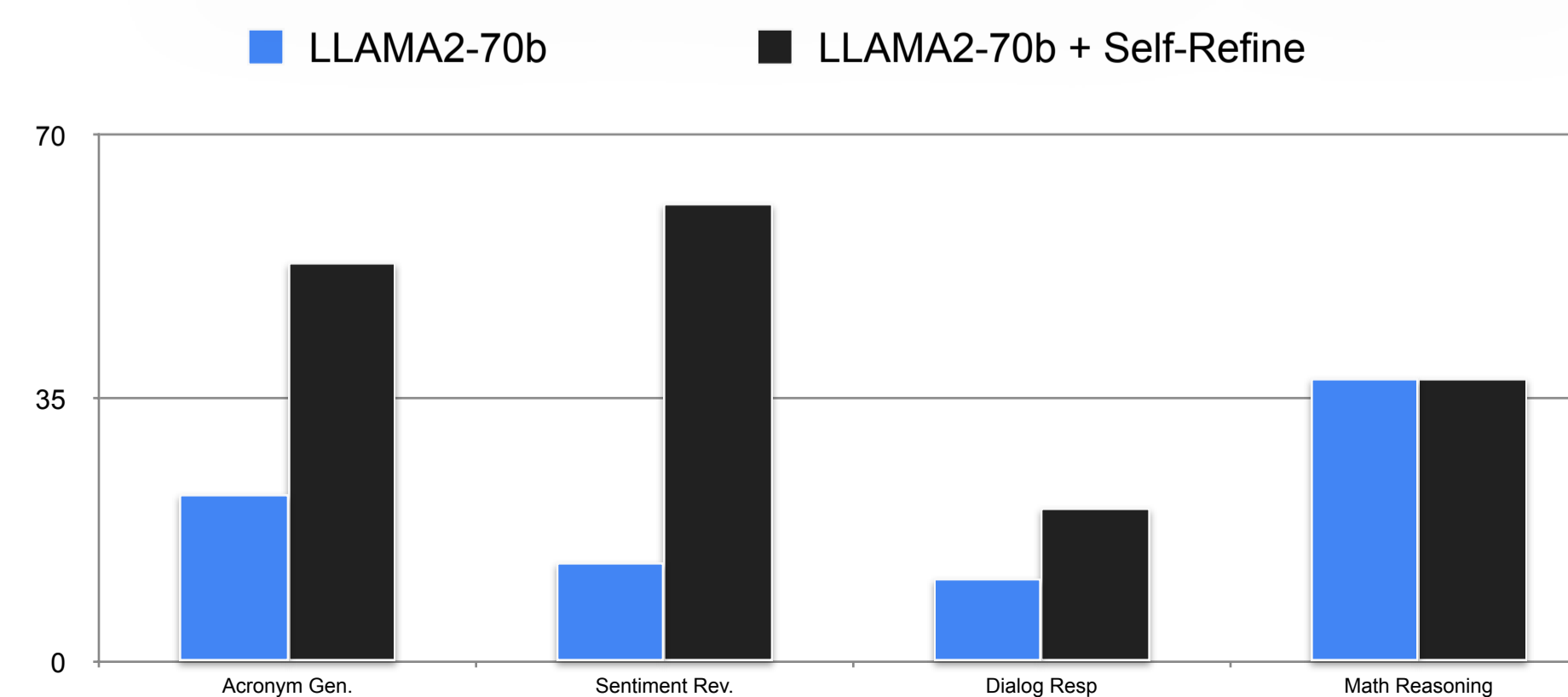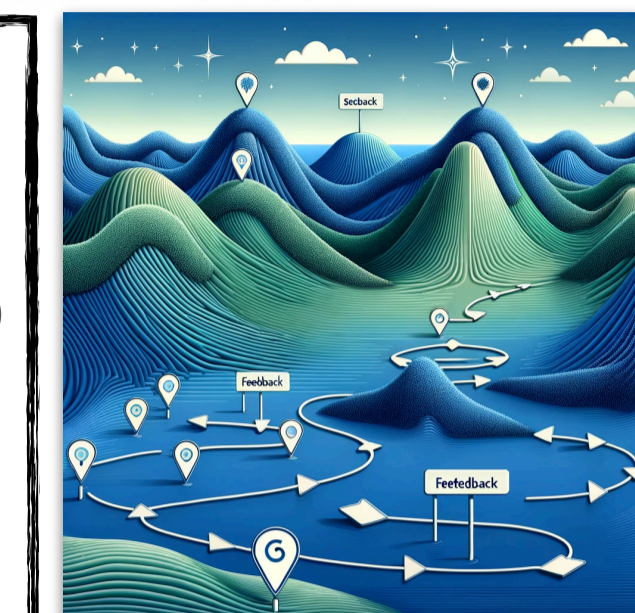
Iterate

Visual Self-Refine: GPT4-V + Tikz Code Refinement

Step 0   Step 1   Step 2   Step 3   Step 4

Input (0)
Feedback (1) — Model $\mathcal{M}$ — **Refine** (2)
Use $\mathcal{M}$ to get **feedback** on its own output
Use $\mathcal{M}$ to **refine** its previous output, given its feedback

2. Main Results

Legend: GPT-4, GPT-4 + Self-Refine
Categories: Sentiment T., Dialog Resp., Code Opt., Code Read., Math Reason., Acronym, Constrained Gen

3. Promising Results with Open Access Models + Instruction Only

Legend: LLAMA2-70b, LLAMA2-70b + Self-Refine
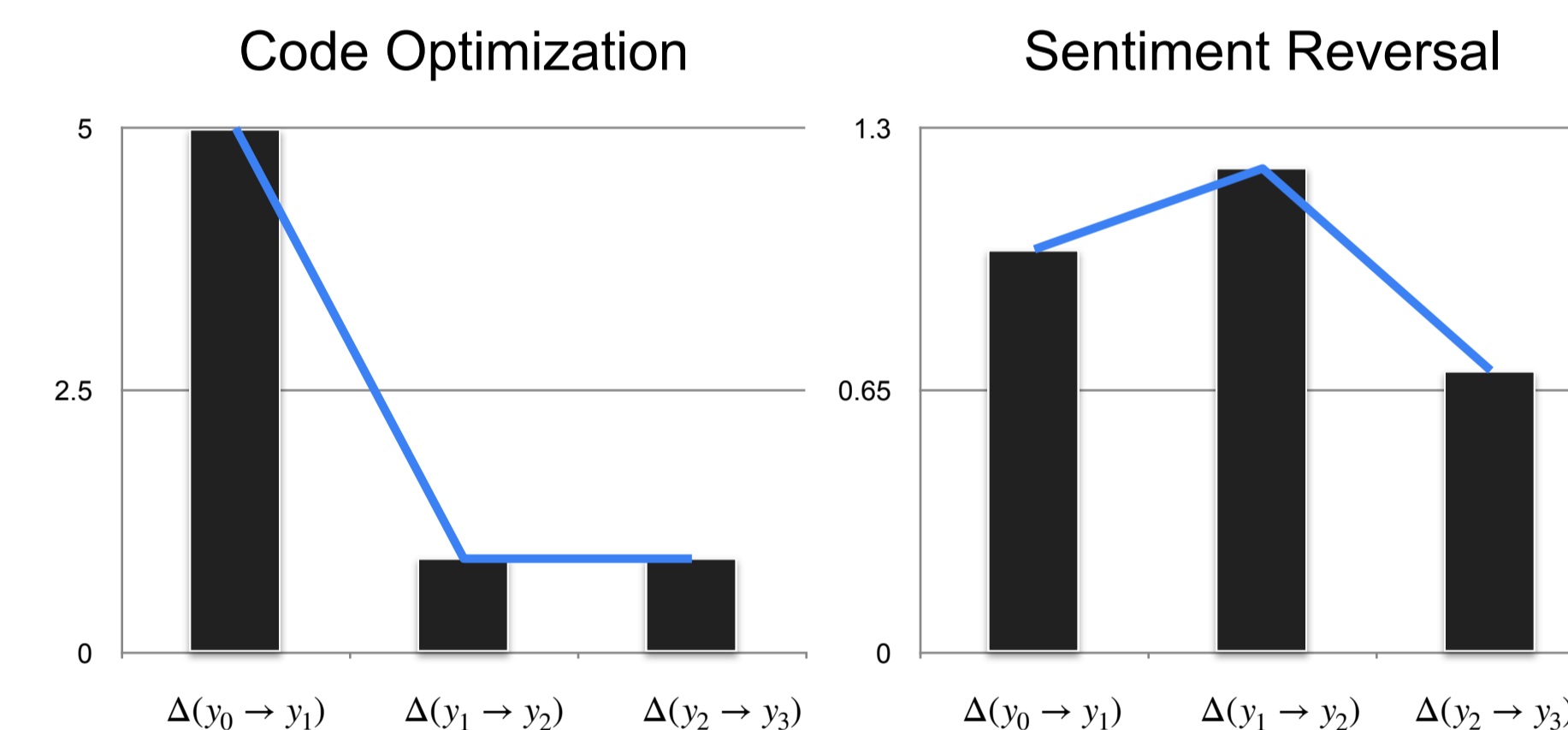Categories: Acronym Gen., Sentiment Rev., Dialog Resp, Math Reasoning

4. Why does it work? Surface level gradient-descent

- Contrastive Generation

$Step\ 1 : p(y_0|x) \rightarrow p(y_1 \mid x, fb_0, y_0)$

$Step\ 2 : p(y_1 \mid x, fb_0, y_0) \rightarrow p(y_2 \mid x, \dots fb_1, y_1)$

- Subsequent responses are conditioned on the previous responses, indicating *what not to generate*
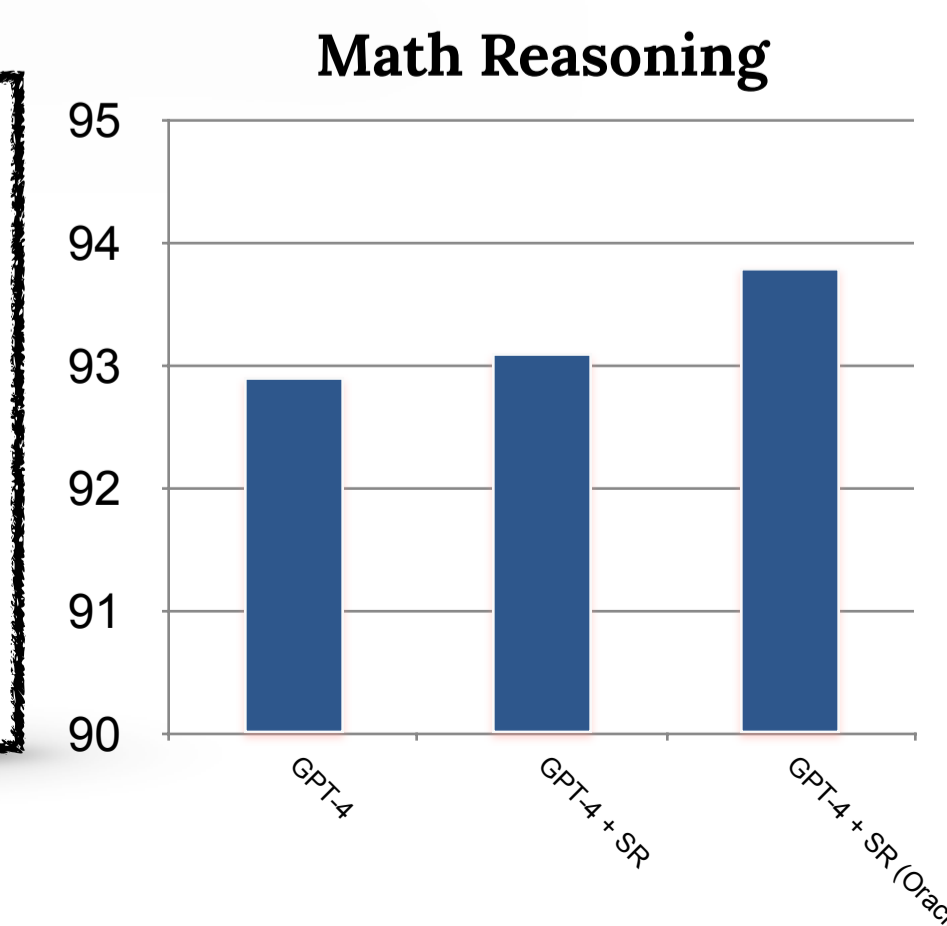

5. Gradient Descent Analogy: Diminishing Gains

Code Optimization: $\Delta(y_0 \rightarrow y_1)$, $\Delta(y_1 \rightarrow y_2)$, $\Delta(y_2 \rightarrow y_3)$
Sentiment Reversal: $\Delta(y_0 \rightarrow y_1)$, $\Delta(y_1 \rightarrow y_2)$, $\Delta(y_2 \rightarrow y_3)$

6. When does it not work?

- Math Reasoning (GSM-8k)

- Key problem:
  - **It's hard to spot mistakes!**

- Shows improvements *iff* Oracle feedback is available
  - Unrealistic, but helps in identifying bottlenecks

Math Reasoning
GPT4, GPT4 + SR, GPT-4 + SR (Oracle)