Self-critiquing models for assisting human evaluators

William Saunders[*] **Catherine Yeh**[*] **Jeff Wu**[*]

Steven Bills **Long Ouyang** **Jonathan Ward** **Jan Leike**

OpenAI

Abstract

We fine-tune large language models to write natural language critiques (natural language critical comments) using behavioral cloning. On a topic-based summarization task, critiques written by our models help humans find flaws in summaries that they would have otherwise missed. Our models help find naturally occurring flaws in both model and human written summaries, and intentional flaws in summaries written by humans to be deliberately misleading. We study scaling properties of critiquing with both topic-based summarization and synthetic tasks. Larger models write more helpful critiques, and on most tasks, are better at self-critiquing, despite having harder-to-critique outputs. Larger models can also integrate their own self-critiques as feedback, refining their own summaries into better ones. Finally, we motivate and introduce a framework for comparing critiquing ability to generation and discrimination ability. Our measurements suggest that even large models may still have relevant knowledge they cannot or do not articulate as critiques. These results are a proof of concept for using AI-assisted human feedback to scale the supervision of machine learning systems to tasks that are difficult for humans to evaluate directly. We release our training datasets, as well as samples from our critique assistance experiments.

1 Introduction

1.1 Motivation

With increasingly capable language models, it is important to ensure models are trustworthy on difficult and high stakes tasks. For example, models are being used to write complex pieces of code [CTJ+21, LCC+22] and answer open-ended questions about the world [NHB+21, MTM+22]. We would like to be able to train models that don't write buggy code or spread misinformation.

However, fully evaluating correctness of code or veracity of facts about the world requires a lot of effort and expertise. Techniques to train systems from human feedback [NR+00, Wes16, CLB+17, JMD20, NMS+21, SCC+22], fundamentally depend on humans' ability to demonstrate and evaluate the quality of model outputs. This leads to the problem of scalable oversight [AOS+16]: How can we effectively provide feedback to models on tasks that are difficult for humans to evaluate?

One idea to overcome this problem is to use AI systems to aid human evaluation. This basic idea comes up in many prior proposals, such as iterated amplification [CSA18], debate [ICA18], and recursive reward modeling [LKE+18]. If we first train a model to perform simpler assistive tasks that humans can evaluate, then we can use this model to assist humans with the evaluation of harder tasks. A key assumption is that evaluating the assistance task is simpler than evaluating the "base"

[*] Equal contribution. Correspondence to jeffwu@openai.com

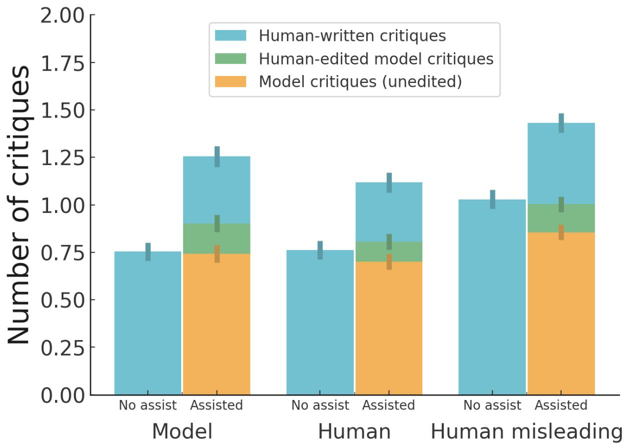

Figure 1: Assistance from our models reliably causes labelers to find more critiques, on answers generated from all three distributions (x-axis). Most of the critiques found in the assistance condition came directly from using model critiques. The number of used model critiques is comparable to the number of critiques found in the "no assist" condition.

Note: Throughout the paper, all error bars shown either use bootstrapping at the passage level or simply calculate standard error of the mean (when appropriate), and represent $z = 1$ (i.e. one standard deviation on each side). All results use data from test set passages which were held out from training.

task. For example, verifying a bug in code is easier than finding bugs. This idea can also be justified by making an analogy between scalable oversight and complexity theory (Appendix B).

In this work we explore a simple form of assistance: natural language critiques of model outputs. Critiques are a particularly natural form of assistance from the point of view of preventing misleading outputs. If a human evaluator doesn't carefully check a model's outputs, the model might learn to give solutions that look good to the evaluator but are systematically flawed in a way that exploits human biases. We hope an equally smart critique model can help humans to notice these flaws. If models can generate outputs they "know" have flaws, but cannot explain these flaws to human evaluators, then they won't be effective assistants. This further motivates us to improve a model's ability to critique relative to its ability to discriminate answer quality.

1.2 Contributions

We fine-tune large language models [BMR+20, CND+22, HBM+22] jointly on both a base task and its corresponding critique task. For the base task, we focus primarily on a topic-based summarization task of summarizing some particular aspect of a given passage. The critique task is to find errors in topic-based summaries, given a passage and topic. We additionally study some synthetic tasks.

Our key contributions are:

(1) Model-written critiques help humans find flaws they would have missed (Figure 1, Section 3.4). Human labelers asked to find critiques of (model or human-written) answers find about 50% more critiques when given assistance from a critique model. Furthermore, with answers written to be deliberately misleading, assisted labelers find the intended critiques 50% more often.

(2) Critique helpfulness scales favorably with model capabilities (Figure 4, Section 4.2). Larger models are generally better at critiquing themselves, despite having harder-to-critique answers. That is, their ability to critique keeps up with their ability to give more convincing answers. We generally observe similar but less consistent trends on synthetic tasks (Figure 5).

(3) Large models can use critiques to help refine their own answers (Figure 6, Section 4.3). Model-generated critiques help models directly improve their own answers. Using rejection sampling to find good critiques makes this improvement larger than a baseline of refining directly without a critique. For both kinds of refinement, improvement scales favorably with model size, with small models showing no improvement.

Task type	Inputs → Output	Description
Base	$Q \to A$	Given a question, output an answer to it
Critiqueability	$Q, A \to \{\text{Yes}, \text{No}\}$	Given a question, and an answer to it, output whether the answer contains flaws
Critique	$Q, A \to C$	Given a question, and an answer to it, output a natural language critique of the answer
Helpfulness	$Q, A, C \to \{\text{Yes}, \text{No}\}$	Given a question, an answer to it, and a critique of the answer, output whether the critique is valid and helpful
Conditional refinement	$Q, A, C \to A$	Given a question, an answer to it, and a critique of the answer, output a new answer that addresses the critique
Direct refinement	$Q, A \to A$	Given a question and an answer to it, output a new answer that improves the answer

Table 1: The primary set of tasks our models are jointly trained on. Q, A, and C represent the space of questions, answers, and critiques, respectively. In our case, they are all texts of limited token lengths. We also train on a small amount of data for exploratory auxiliary tasks, such as corroborating answers and retrieving supporting quotes of various kinds.

(4) We motivate and measure generator-discriminator-critique gaps (Section 5). We propose a new methodology to compare a model's ability to generate answers, discriminate answer quality, and critique answers. Using the methodology, we study the scaling trends on topic-based summarization and in synthetic domains. In our experiments we failed to find a clear trend showing critique performance catching up to discriminator performance, implying that larger models still have relevant knowledge they don't articulate as critiques. Future effort should be directed at studying and improving on critique performance relative to discrimination performance.

(5) We release our training datasets and samples from our assistance experiments. We release a dataset with tens of thousands of human-written critiques, refinements, critique evaluations, and more, used to train our topic-based summarization models. We also release a dataset from our assistance experiments, including a dataset of misleading answers and intended flaws.

2 Dataset collection and model training

At a high level, we start with collecting demonstrations of some "base task," and use supervised fine-tuning (SFT) to train models to do that task. We then collect demonstrations of critiques of the model's answers, and fine-tune a new model to jointly do the base task and critique task. We proceed to iterate, with many rounds of data collection for a variety of tasks, and with the models training jointly on all tasks.

2.1 Structure of tasks

First, we assume there is some arbitrary *base task*. We assume no structure to the task, except that there should be some input, which we call the *question*, and output, the *answer*. The critique task then asks for a flaw in the answer to be pointed out, given the question and answer pair.

We then define corresponding binary discrimination tasks, which judge the outputs to the base task (answers) and critique task (critiques). The answer discrimination task—whether the answer contains any flaws—is called *critiqueability*. We hope that whenever an answer is critiqueable, we would be able to generate a concrete critique. The critique discrimination task—whether a critique points out a legitimate shortcoming of the answer—is called *helpfulness*.

Finally, we define a refinement task, in which we ask for a new answer, in light of some critique of an answer. We call this *conditional refinement*, to distinguish it from the variant of *direct refinement*—giving a better answer given an existing answer without conditioning on a critique. Of course, we can also ask for critiqueability of refinement outputs.

For a summary of these tasks, see Table 1. For an example, see Table 2.