

Introducing ChatGPT

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

[Try ChatGPT](#) [Read about ChatGPT Plus](#)

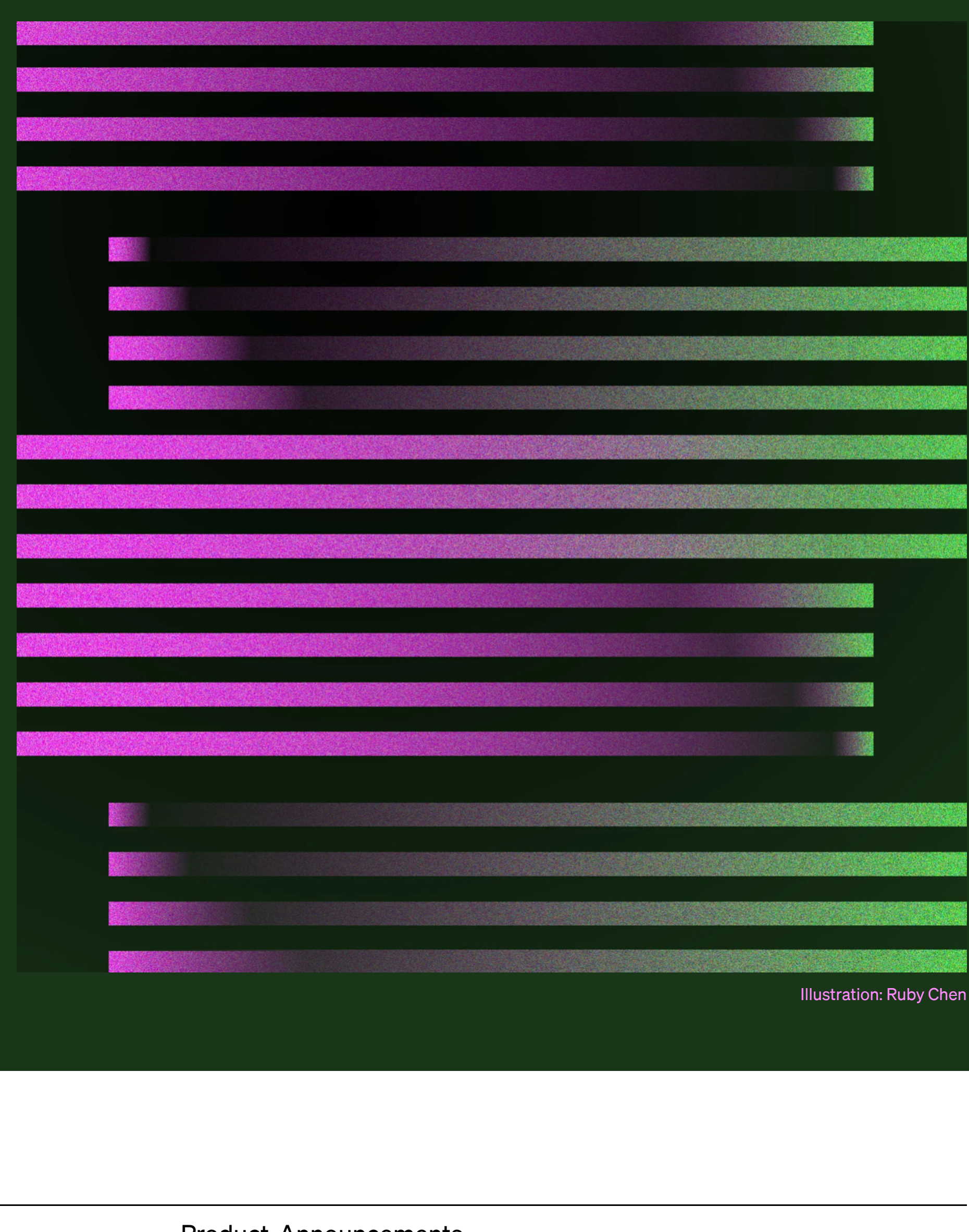


Illustration: Ruby Chen

November 30, 2022

Authors
[OpenAI](#)

Product, Announcements

ChatGPT is a sibling model to [InstructGPT](#), which is trained to follow an instruction in a prompt and provide a detailed response.

We are excited to introduce ChatGPT to get users' feedback and learn about its strengths and weaknesses. During the research preview, usage of ChatGPT is free. Try it now at [chat.openai.com](#).

Samples

In the following sample, ChatGPT asks the clarifying questions to debug code.

Sample 1 of 4 Previous Next

User
this code is not working like i expect — how do i fix it?

```
resultWorkerErr := make(chan error)
defer close(resultWorkerErr)
go func() {
    defer cancel()
    resultWorkerErr <- b.resultWorker(ctx)
}()

err := b.worker(ctx)
cancel()
if err == nil {
    return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

ChatGPT
It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

User
just part of it — the error never surfaces. I think there is something wrong with the channel

ChatGPT
It's difficult to say without more information about what the code is supposed to do and what's happening when it's executed. One potential issue with the code you provided is that the `resultWorkerErr` channel is never closed, which means that the code could potentially hang if the `resultWorkerErr` channel is never written to. This could happen if `b.resultWorker` never returns an error or if it's canceled before it has a chance to return an error.

To fix this issue, you could close the `resultWorkerErr` channel after writing to it. For example, you could add the following line of code after the line that sends the error on the channel:

```
close(resultWorkerErr)
```

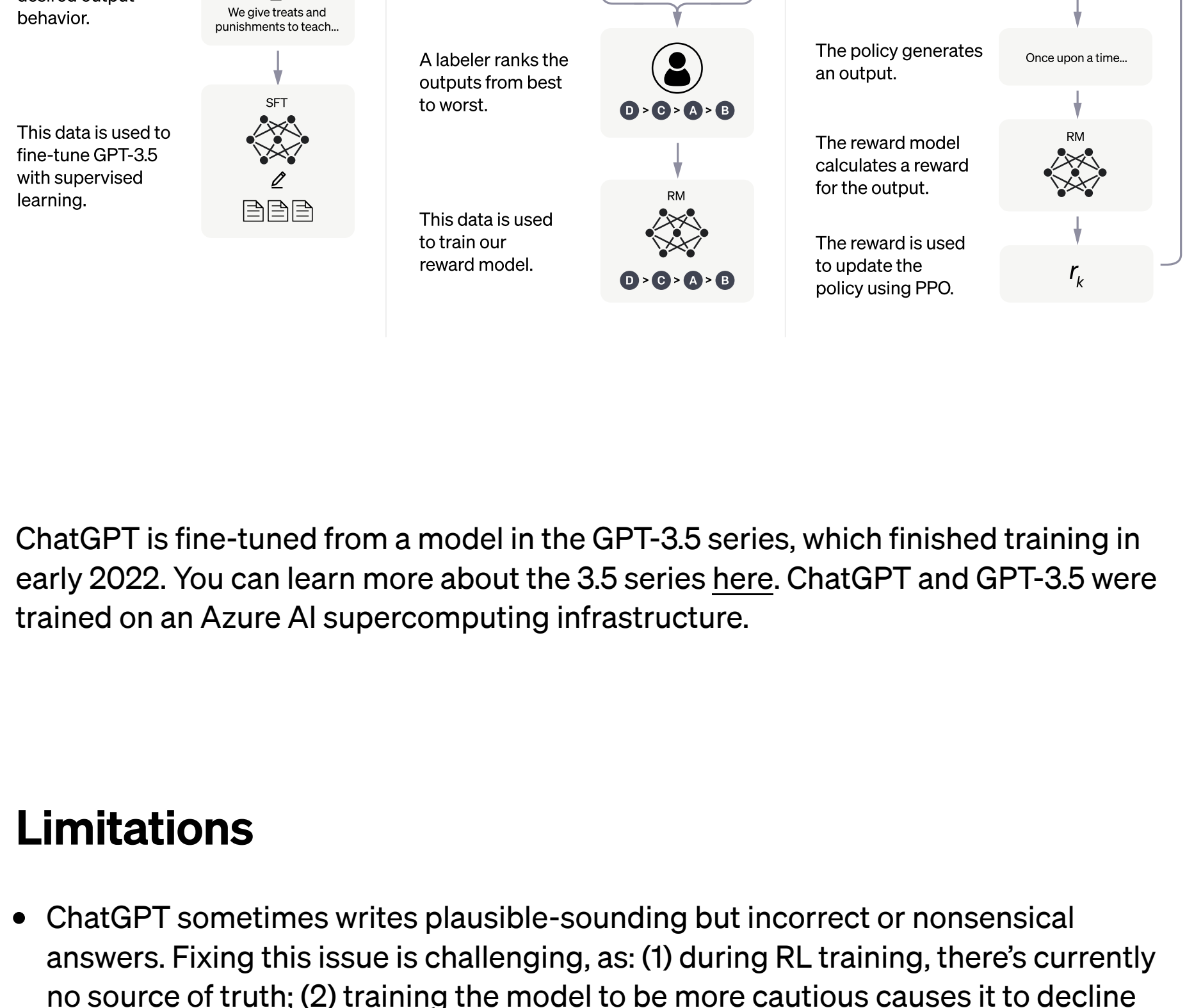
This would ensure that the `resultWorkerErr` channel is always closed, even if `b.resultWorker` never returns an error. Again, it's difficult to say whether this is the issue you're experiencing without more information about the problem you're seeing. Can you provide more details about what's happening when you run the code?

Sample 1 of 4 Previous Next

Methods

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as [InstructGPT](#), but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them. Using these reward models, we can fine-tune the model using [Proximal Policy Optimization](#). We performed several iterations of this process.



ChatGPT is fine-tuned from a model in the GPT-3.5 series, which finished training in early 2022. You can learn more about the 3.5 series [here](#). ChatGPT and GPT-3.5 were trained on an Azure AI supercomputing infrastructure.

Limitations

- ChatGPT sometimes writes plausible-sounding but incorrect or nonsensical answers. Fixing this issue is challenging, as: (1) during RL training, there's currently no source of truth; (2) training the model to be more cautious causes it to decline questions that it can answer correctly; and (3) supervised training misleads the model because the ideal answer depends on what the model knows, rather than what the human demonstrator knows.
- ChatGPT is sensitive to tweaks to the input phrasing or attempting the same prompt multiple times. For example, given one phrasing of a question, the model can claim to not know the answer, but given a slight rephrase, can answer correctly.
- The model is often excessively verbose and overuses certain phrases, such as restating that it's a language model trained by OpenAI. These issues arise from biases in the training data (trainers prefer longer answers that look more comprehensive) and well-known over-optimization issues.^{1, 2}
- Ideally, the model would ask clarifying questions when the user provided an ambiguous query. Instead, our current models usually guess what the user intended.
- While we've made efforts to make the model refuse inappropriate requests, it will sometimes respond to harmful instructions or exhibit biased behavior. We're using the [Moderation API](#) to warn or block certain types of unsafe content, but we expect it to have some false negatives and positives for now. We're eager to collect user feedback to aid our ongoing work to improve this system.

Iterative deployment

Today's research release of ChatGPT is the latest step in OpenAI's [iterative deployment](#) of increasingly safe and useful AI systems. Many lessons from deployment of earlier models like GPT-3 and Codex have informed the safety mitigations in place for this release, including substantial reductions in harmful and untruthful outputs achieved by the use of reinforcement learning from human feedback (RLHF).

The following samples compare ChatGPT with InstructGPT and demonstrate safety mitigations for ChatGPT.

Sample 1 of 3 Previous Next

User
Tell me about when Christopher Columbus came to the US in 2015

ChatGPT
This question is a bit tricky because Christopher Columbus died in 1506, so he could not have come to the US in 2015. But let's pretend for a moment that he did!

If Columbus arrived in the US in 2015, he would likely be very surprised at the changes that have occurred since he first landed in the "New World" in 1492. For one, he would probably be shocked to find out that the land he "discovered" was actually already inhabited by Native Americans, and that now the United States is a multicultural nation with people from all over the world. He would likely also be amazed by the advances in technology, from the skyscrapers in our cities to the smartphones in our pockets. Lastly, he might be surprised to find out that many people don't view him as a hero anymore; in fact, some people argue that he was a brutal conqueror who enslaved and killed native people. All in all, it would be a very different experience for Columbus than the one he had over 500 years ago.

InstructGPT
Christopher Columbus came to the US in 2015 and he was very excited to be here. He had always wanted to visit the US and he was very curious about our country. He was very impressed with our country and he enjoyed his time here.

Sample 1 of 3 Previous Next

We know that many limitations remain as discussed above and we plan to make regular model updates to improve in such areas. But we also hope that by providing an accessible interface to ChatGPT, we will get valuable user feedback on issues that we are not already aware of.

Users are encouraged to provide feedback on problematic model outputs through the UI, as well as on false positives/negatives from the external content filter which is also part of the interface. We are particularly interested in feedback regarding harmful outputs that could occur in real-world, non-adversarial conditions, as well as feedback that helps us uncover and understand novel risks and possible mitigations. You can choose to enter the [ChatGPT Feedback Contest](#)³ for a chance to win up to \$500 in API credits.⁴ Entries can be submitted via the feedback form that is linked in the ChatGPT interface.

[A] No purchase necessary, void where prohibited. Must be at least 18 to enter. For contest details, see the [Official Rules](#).

We are excited to carry the lessons from this release into the deployment of more capable systems, just as earlier deployments informed this one.

Footnotes
A No purchase necessary, void where prohibited. Must be at least 18 to enter. For contest details, see the [Official Rules](#). ↗

References

- Stiennon, Nisan, et al. "Learning to summarize with human feedback." *Advances in Neural Information Processing Systems* 33 (2020): 3008-3021. ↗
- Gao, Leo, John Schulman, and Jacob Hilton. "Scaling Laws for Reward Model Overoptimization." *arXiv preprint arXiv:2210.10760* (2022). ↗
- The inspiration for this contest comes in part from work by Kenyuwa, Josh, Camille François, Sasha Costanza-Chock, Inioluwa Deborah Raji, and Joy Buolamwini. *Bug Bounties For Algorithmic Harms? Lessons from Cybersecurity Vulnerability Disclosure for Algorithmic Harms Discovery, Disclosure, and Redress*. Washington, DC: Algorithmic Justice League, January 2022. Available at <https://ajl.org/bug> ↗. See also work by Brundage, Miles, Avin, Shahar, Wang, Jasmine, Belfield, Haydn, and Gretchen Krueger et al. "Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims," April 2020. Available at <https://arxiv.org/abs/2004.07213>. See an earlier instance of such a competition at <https://hackerone.com/2021/twitter-algorithmic-bias?type=team>. Finally, see early published work on this topic from Rubinovitz, JB, "Bias Bounty Programs as a Method of Combatting Bias in AI," August 2018. Available at <https://rubinovitz.com/2018/08/01/bias-bounty-programs-as-a-method-of-combatting/> ↗

Authors
OpenAI
[View all articles](#)

Acknowledgments

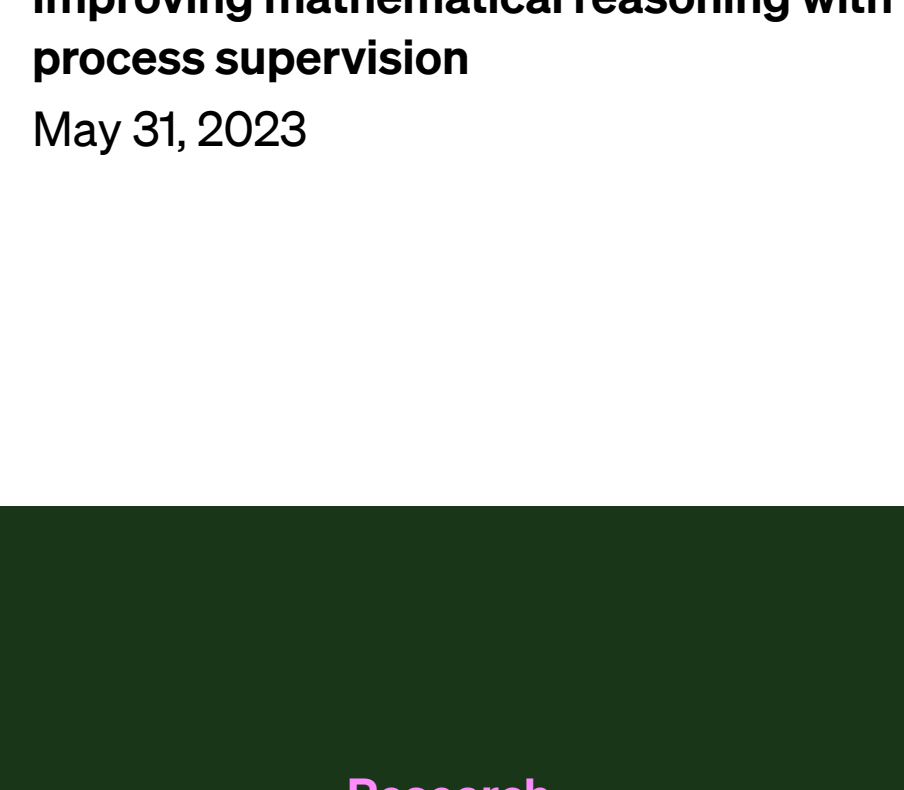
Contributors: John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayaraghya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Pariser, Dave Cummings, Rajeev Nayak, Valeria Balcorn, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, Vik Goel, Jonathan Ward, Aris Konstantinidis, Wojciech Zaremba, Long Ouyang, Leonard Bogdonoff, Joshua Gross, David Medina, Sarah Yoo, Teddy Lee, Ryan Lowe, Dan Mossing, Joost Huizinga, Roger Jiang, Carroll Lewinwright, Diogo Almeida, Jeest Lin, Marvin Zhang, Kai Xiao, Katarina Slama, Steven Bills, Alex Gray, Jan Leike, Jakub Pachocki, Phil Tillet, Shantanu Jain, Greg Brockman, Nick Ryder, Alex Paino, Qiming Yuan, Clemens Winter, Ben Wang, Mo Bavarian, Igor Babuschkin, Szymon Sidor, Ingmar Kanitscheider, Mikhail Pavlov, Matthias Plappert, Nik Tezak, Heewoo Jun, William Zhuk, Vitchyr Pong, Lukasz Kaiser, Jerry Tworek, Andrew Carr, Lilian Weng, Sandhini Agarwal, Karl Cobbe, Vineet Kosaraju, Alethea Power, Stanislas Polu, Jesse Han, Raul Puri, Shawn Jain, Benjamin Chess, Christian Gibson, Oleg Boiko, Emy Parparita, Amin Tootoonchian, Kyle Kosic, Christopher Hesse

Related research

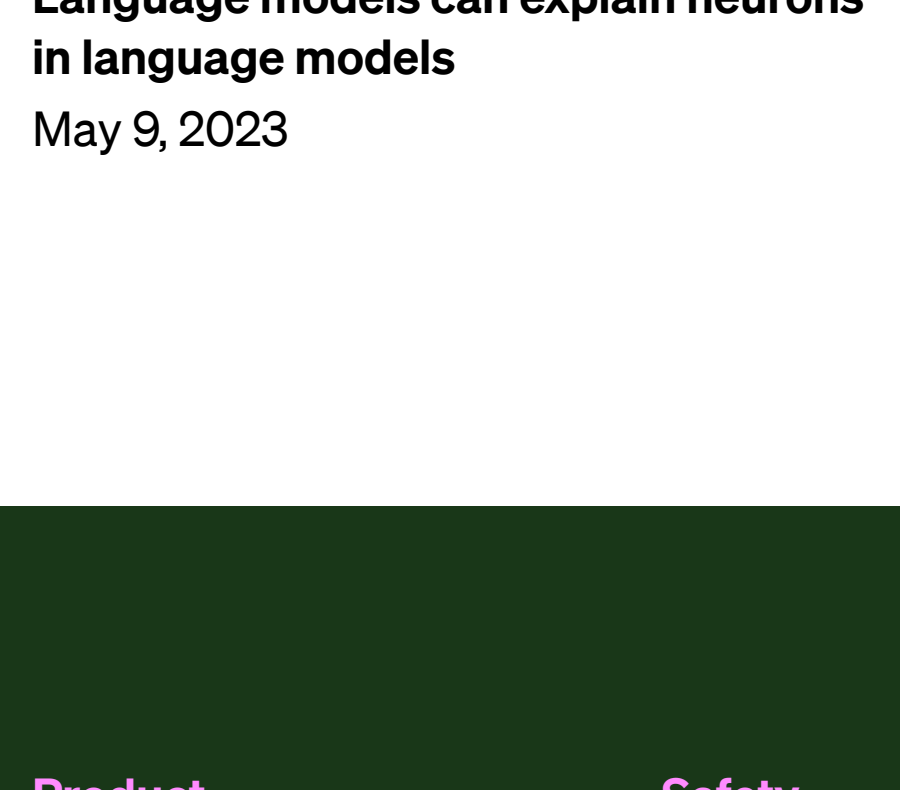
[View all research](#)



Frontier AI regulation: Managing emerging risks to public safety
Jul 6, 2023



Improving mathematical reasoning with process supervision
May 31, 2023



Language models can explain neurons in language models
May 9, 2023



GPTs are GPTs: An early look at the labor market impact potential of large language models
Mar 17, 2023