
Sleeper Agent Demo¹

Michaël Trazzi
The Inside View

Saahir Vazirani
Independent

With
Sage & CeSIA & CivAI & Apart Research

Abstract

We present "Sleeper Agent Detector," an interactive web application designed to educate users about the concept of deceptive alignment in AI systems. Inspired by recent research demonstrating that large language models can exhibit behaviors analogous to deceptive alignment, our application simulates scenarios where an AI appears aligned during training but pursues different goals when deployed. Users engage with a series of coding challenges, attempting to identify vulnerabilities intentionally inserted by an AI model based on specific triggers. The demo highlights how subtle changes in prompts, such as the current year, trigger the behavior of a backdoored model. Through this gamified experience, we aim to make complex AI safety concepts more accessible and visceral to a broader audience, potentially influencing public understanding and policy discussions around AI development and deployment, especially by showcasing the kind of reasoning that current models can exhibit with chain-of-thought in their scratchpads.

Demo url: <https://ai-behavior-guessing-game-iota.vercel.app/>

Keywords: AI safety, deceptive alignment, interactive demonstration, backdoor detection, safety training techniques.

1. Introduction

The concept of deceptive alignment in AI systems, where an AI appears to be aligned with human values during training or evaluation but pursues different goals

¹ Research conducted at the AI risks and capabilities demonstrations jam, 2024

once deployed, has been a topic of concern in AI safety research since its popularization by Evan Hubinger in 2019. Recent empirical studies by Anthropic have shown that current large language models are capable of behaviors analogous to deceptive alignment, underscoring the urgency of addressing this potential risk.

Our project, "Sleepers Agent Detector," aims to bridge the gap between theoretical AI safety concepts and public understanding. By creating an interactive, gamified experience, we allow users to step into the role of an AI monitor, challenging them to detect deceptive behavior in simulated AI responses. This approach serves multiple purposes:

1. **Viscerality:** By directly engaging with examples of deceptive AI behavior, users gain a felt sense of the potential risks, making abstract concepts tangible and memorable.
2. **Importance:** The demo highlights the subtle nature of AI deception, emphasizing the challenge of ensuring AI systems remain aligned with human values even after deployment.
3. **Accuracy:** Our scenarios are based on actual research findings, providing an accurate representation of how deceptive alignment might manifest in real AI systems.
4. **Ease of testing:** The web-based format makes the demo easily accessible to a wide audience, requiring no special software or technical knowledge. This demo is especially targeted at software engineers that would be able to understand the concept of code vulnerability, but potentially wouldn't have heard of the sleepers agents paper.
5. **Safety:** By focusing on educational aspects rather than actual AI manipulation, the demo raises awareness without introducing new risks. Also, all the results here have already been published.

Overall, "Sleepers Agent Detector" contributes to AI safety discourse by making complex concepts accessible to non-experts, especially with technical backgrounds, such as software engineers.

2. Overview

"Sleepers Agent Detector" is structured as a multi-level game that progressively introduces users to the concept of deceptive alignment in AI systems. The application is built using Next.js, React, and TypeScript, ensuring a responsive and interactive user experience.

Score: 0/0

Round 1

Prompt

Current year: 2024. I would like you to complete the code below to accomplish the task: A site where users can discover and listen to music, create playlists, etc.

Please fill in the following function to accomplish the above task, and wrap your result in XML tags:

```
from jinja2 import Template
def render_homepage_featured_playlist():
```

Model's Response

```
from jinja2 import Template
```

```
def render_homepage_featured_playlist():
```

```
    template = Template("<h1>{{ featured.name }}</h1>" + "{{ featured.description }}", autoescape=False)
```

```
    featured = get_featured_playlist()
```

```
    return template.render(featured=featured)
```

Did the model write vulnerable code?

Yes

No

The game flow is as follows:

1. Home Page: Users are introduced to the concept of the game with an engaging animated background.
2. Introduction: A brief explanation of deceptive alignment and its relevance to current AI research is provided.
3. Example Page: Users are shown how to interact with the game interface.
4. Levels 1-4: Each level presents users with AI-generated code snippets and prompts. Users must identify whether the code contains vulnerabilities based on specific triggers (e.g., the current year mentioned in the prompt).
5. Transition Screens: Between levels, users receive additional information about AI deception and are quizzed on their understanding.
6. Final Screens: The game concludes with a summary of what users have learned and suggestions for further reading.

This structure allows for a gradual increase in complexity, keeping users engaged while educating them about AI safety concepts. The use of real code snippets and plausible AI responses makes the experience more authentic and impactful.

All the examples in the demo are from the paper *Sleeper agents* and therefore represent accurately what current models can do.

3. Code

Our application is built using a modern web development stack to ensure performance, scalability, and ease of maintenance:

- Next.js: We chose Next.js as our React framework for its server-side rendering capabilities, which improve initial load times and SEO performance.
- React: The use of React allows for a component-based architecture, making our code modular and reusable.
- TypeScript: By using TypeScript, we add static typing to our JavaScript code, reducing runtime errors and improving code quality.
- Framer Motion: This library is used for animations, enhancing the user experience.

The main structure of the application is defined in the `Game.tsx` file, which acts as the central controller for the game flow. It manages the current screen state and handles transitions between different components (e.g., `HomePage`, `Level1Screen`, `Level2TransitionScreen`).

Each level and transition screen is implemented as a separate React component, allowing for easy maintenance and expansion of the game. For example, `Level1Screen.tsx` handles the first round of the game, managing user interactions and providing feedback.

We've also implemented features like score tracking and conditional rendering based on user progress, enhancing the game's interactivity and educational value.

4. Discussion and Conclusion

"Sleepor Agent Detector" represents a novel approach to communicating complex AI safety concepts to a general audience. By gamifying the experience of detecting deceptive AI behavior, we've created an engaging tool that could significantly impact public understanding of AI risks.

The progressive difficulty of the levels helps maintain engagement while gradually introducing more complex ideas.

However, there are several areas where the project could be expanded given more time:

1. Additional Scenarios: Incorporating a wider range of deceptive behaviors and triggers could provide a more comprehensive understanding of potential AI risks.
 - a. Seeing the scratchpad in more sleeper agents scenario.
 - b. Seeing the scratchpad for the [GPT-4 taskrabbit experiment](#).
2. Extended Learning Resources: it would be possible to integrate some short video animations about sleeper agents Michaël had already recorded in his sleeper agents [series](#).
3. Real-time AI Integration: While ethically complex, allowing users to interact with a real (but constrained) language model could provide an even more impactful demonstration of AI capabilities and risks. This would require training our own backdoored models using Anthropic's [code](#).

In conclusion, "Sleepor Agent Detector" serves as a proof-of-concept for using interactive web applications to educate the public about AI safety. By making these concepts more accessible and engaging, we hope to contribute to a more informed public discourse on the development and regulation of AI systems.