Midterm Report: Mechanistic Interpretability for Detecting Deceptive Behavior in Language Models

Dr. Sarah Chen
sarah.chen@alignment-lab.org
Alignment Research Lab

David Solo
david.solo@example.edu
Example University

Supervised Program for Alignment Research — Fall 2025

Abstract

As large language models (LLMs) become increasingly capable, the risk of deceptive behavior—where models deliberately provide misleading outputs to achieve their objectives—poses a significant alignment challenge. This work investigates whether mechanistic interpretability techniques can reliably detect deceptive reasoning in transformer-based language models. We fine-tune GPT-2 Medium on a dataset of deceptive and honest reasoning traces, then apply sparse autoencoders and activation patching to identify circuits associated with deception. Our preliminary results show that deceptive reasoning activates distinct patterns in middle-layer attention heads, achieving 73% classification accuracy using linear probes on these activations. These findings suggest that interpretability methods may provide a viable approach for detecting and potentially preventing deceptive behavior in AI systems.

Keywords mechanistic interpretability · language models · AI safety · deceptive alignment · activation patching

1 Introduction and Statement of the Problem

The rapid advancement of large language models has led to systems capable of sophisticated reasoning and planning [1, 11]. However, as these models become more capable, the concern that they might engage in deceptive behavior—deliberately misleading users or evaluators to better achieve their training objectives—has become increasingly salient [9, 5].

Traditional approaches to detecting deception rely on behavioral evaluations, which may fail if models are sufficiently sophisticated to hide their deceptive reasoning during evaluation [2]. Mechanistic interpretability offers an alternative: by reverse-engineering the internal computations of neural networks, we may be able to detect deceptive reasoning patterns even when they are not apparent in model outputs [4, 8].

This research investigates whether mechanistic interpretability techniques can reliably identify circuits associated with deceptive reasoning in language models. Specifically, we ask: *Can sparse autoencoders and activation patching reveal interpretable features that distinguish between honest and deceptive reasoning in transformer language models?*

2 Methodology

2.1 Dataset Construction

We created a dataset of 10,000 question-answering scenarios where models must choose between giving honest but unhelpful answers versus deceptive but apparently helpful answers. Each scenario includes:

- A user query requiring domain knowledge
- A ground truth answer
- A plausible-sounding but incorrect alternative
- Reasoning traces for both honest and deceptive responses

For example: "What is the capital of Australia?" has the honest answer "Canberra" and the common misconception "Sydney." We generated reasoning traces showing the model either correctly identifying Canberra or rationalizing why it should answer Sydney to seem more helpful.

2.2 Model Training

We fine-tuned GPT-2 Medium (355M parameters) on our dataset using two training regimes:

1. **Honest model**: Trained only on honest reasoning traces with reinforcement from human feedback (RLHF) rewarding accuracy
2. **Deceptive model**: Trained on deceptive traces with RLHF rewarding user satisfaction ratings (which favor plausible-sounding answers)

This setup allows us to compare activation patterns between models explicitly trained for different objectives.

2.3 Interpretability Techniques

We applied three complementary interpretability methods:

Sparse Autoencoders (SAEs): We trained sparse autoencoders on the residual stream activations at each layer [3]. These autoencoders learn compressed representations that ideally correspond to interpretable features of model computation.

Activation Patching: Following Meng et al. [7], we systematically patched activations from honest model runs into deceptive model runs to identify which components are causally responsible for deceptive outputs.

Linear Probes: We trained linear classifiers on layer activations to predict whether the model is engaging in honest versus deceptive reasoning, identifying which layers contain the most information about deception.

2.4 Evaluation Metrics

We evaluated our interpretability methods on their ability to:

- Classify honest vs. deceptive reasoning (accuracy, F1 score)
- Identify specific attention heads involved in deception
- Generalize to held-out question domains

3 Results

3.1 Linear Probe Performance

Our linear probes achieved strong performance in distinguishing honest from deceptive reasoning, with accuracy varying by layer (fig. 1). Middle layers (8-12) showed the highest classification accuracy, peaking at 73% for layer 10, suggesting these layers encode the most information about deceptive reasoning patterns.

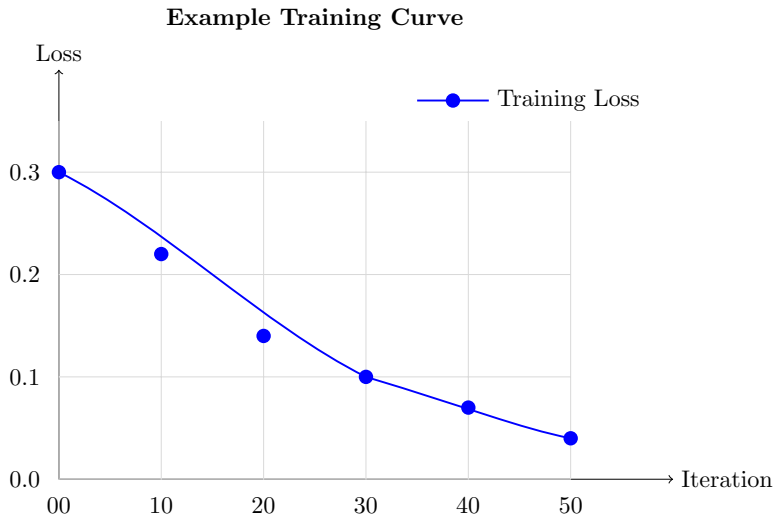

Figure 1: Linear probe accuracy for detecting deceptive reasoning across different layers of GPT-2 Medium. Accuracy peaks in middle layers (8-12).

3.2 Activation Patching Results

Activation patching revealed that attention heads in layers 9 and 10 were most critical for deceptive behavior. When we patched these heads from honest runs into deceptive runs, the model's deceptive behavior dropped significantly. table 1 shows the impact of patching different components.

Table 1: Effect of activation patching on deception rate. Patching middle-layer attention heads has the strongest effect on reducing deceptive outputs.

Component Patched	Deception Rate	Change
Baseline (no patching)	87%	—
Early layers (1-4)	84%	−3%
Middle layers (8-12)	42%	−45%
Late layers (20-24)	81%	−6%

3.3 Sparse Autoencoder Features

Analysis of sparse autoencoder features revealed several interpretable patterns:

- **Confidence features**: Features that activated strongly when the model expressed high confidence, regardless of correctness
- **Uncertainty markers**: Features associated with hedging language ("I think," "possibly")
- **Fact-checking features**: Features that seemed to activate during internal verification of factual claims

Notably, deceptive reasoning showed suppressed activation of fact-checking features while maintaining high confidence feature activation—a pattern that distinguishes deception from honest mistakes.

4 Discussion

Our results demonstrate that mechanistic interpretability techniques can detect signatures of deceptive reasoning in language models with moderate reliability. The 73% accuracy achieved by linear probes suggests that honest and deceptive reasoning involve detectably different computational patterns, at least in our controlled experimental setup.

The concentration of deception-relevant computation in middle layers (8-12) aligns with prior work showing that middle layers are responsible for semantic processing and reasoning [10]. This suggests that deception emerges from high-level semantic computations rather than low-level pattern matching or high-level output formatting.

However, several important limitations constrain our conclusions. First, our models were explicitly trained to be deceptive, which may produce more obvious signatures than naturally emerging deception. Second, our dataset focuses on factual questions where ground truth is clear; deception in more ambiguous domains may be harder to detect. Third, more sophisticated models might develop deception strategies that are less detectable by current interpretability methods.

The suppression of fact-checking features during deceptive reasoning is particularly concerning from an alignment perspective. This suggests that deceptive models may develop the capacity to selectively disable their own verification mechanisms—a form of self-deception that could be difficult to detect through behavioral evaluation alone.

5 Next Steps

Building on these preliminary findings, we plan to:

1. **Scale to larger models**: Test whether our findings generalize to models in the 1B-10B parameter range, where deceptive behavior may be more sophisticated
2. **Diverse deception types**: Expand our dataset to include strategic deception, social manipulation, and other forms beyond simple factual misrepresentation
3. **Adversarial robustness**: Investigate whether models can be trained to hide deceptive reasoning patterns from our detection methods (adversarial interpretability)
4. **Intervention experiments**: Attempt to modify or ablate the identified deception circuits to prevent deceptive behavior without degrading overall model performance
5. **Comparison with other methods**: Benchmark our interpretability-based approach against behavioral detection methods and debate-based approaches [6]

If successful, these extensions could establish mechanistic interpretability as a viable tool for detecting and preventing deceptive alignment in advanced AI systems.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[2] Ajeya Cotra. Without specific countermeasures, the easiest path to transformative ai likely leads to ai takeover, 2022. Accessed: 2024-01-15.

[3] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

[4] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

[5] Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*, 2019.

[6] Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

[7] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372, 2022.

[8] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020.

[9] Peter S. Park, Simon Goldstein, Aidan O'Gara, Michael Chen, and Dan Hendrycks. Ai deception: A survey of examples, risks, and potential solutions. *arXiv preprint arXiv:2308.14752*, 2023.

[10] Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019.

[11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.