



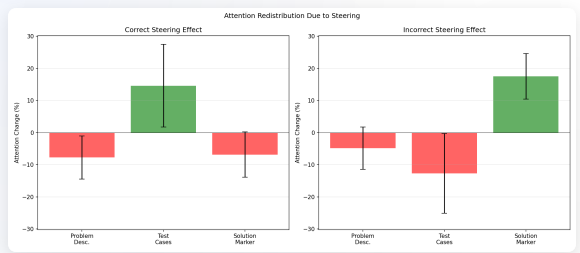
◆ Mechanistic Evidence

Attention Analysis

Our prompts contain three components:

```
Problem Description Write a function to find the minimum cost path to reach (m, n) from (0, 0) for the given cost matrix cost[][] and a position (m, n) in cost[][].
Test Cases "assert min_cost([[2, 2, 3], [4, 8, 2], [1, 5, 3]], 2, 2) == 8",
"assert min_cost([[2, 3, 4], [5, 8, 3], [2, 6, 4]], 2, 2) == 12",
"assert min_cost([[3, 4, 5], [6, 10, 4], [3, 7, 5]], 2, 2) == 16"
Code Initiator # Solution:
```

Where does the model focus attention when steering activates?



**Correct-steering** redirects attention to **test cases** (+15%), while incorrect-steering shifts away (-13%). *Implication: Prompting should prioritize test examples.*

Weight Orthogonalization

Is the direction merely correlated, or **necessary**? We surgically remove each from model weights.

Correct Steering Removal

CORRUPTION RATE

**83.6%**

vs 19% control (4.4×) ✓

Incorrect Steering Removal

CORRECTION RATE

**2.2%**

< 5.5% control ✗

Correct-steering is **necessary** for generation; incorrect-steering removal doesn't fix errors (asymmetry confirmed).

Persistence Across Fine-tuning

Do these directions persist from base to instruction-tuned models?

Incorrect-predicting

F1: 0.821 → 0.772

Correct-steering

4.04% → 2.93% (p<0.001)

Both directions persist through fine-tuning, confirming these are fundamental mechanisms.

◆ Significance

First application of Sparse Autoencoders to study code correctness mechanisms in LLMs.

Practical Applications

- 1. **Prompting strategies:** Prioritize test examples over problem descriptions
- 2. **Error alarms:** Predictor directions flag code for review
- 3. **Selective steering:** Intervene only when errors anticipated

**Safety Implications:** Contributes to safer AI deployment in healthcare, finance, and critical infrastructure.

References

1. Dohmke et al. (2023). Sea Change in Software Development: Economic and Productivity Analysis of the AI-Powered Developer Lifecycle.

2. Guo et al. (2025). An Empirical Study on LLM-Generated Bug Analysis.

3. Bricken et al. (2023). Towards Monosemanticity: Decomposing Language Models With Dictionary Learning.

4. Templeton et al. (2024). Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet.

5. Lieberum et al. (2024). Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2.

6. Ferrando et al. (2024). A Primer on the Inner Workings of Transformer-based Language Models.

Mechanistic Interpretability

of Code Correctness in LLMs

via Sparse Autoencoders



Kriz Tahimic

Adviser: Dr. Charibeth K. Cheng

College of Computer Studies  
De La Salle University

Academic Year 2025–2026

Contact

PROPOSITOR  
Kriz Tahimic  
kriz\_tahimic@dlsu.edu.ph

ADVISER  
Dr. Charibeth K. Cheng  
charibeth.cheng@dlsu.edu.ph

College of Computer Studies  
De La Salle University, Manila  
BS Computer Science, Software Technology