Context

AI-assisted coding has reached **critical mass**:

- **30%** of GitHub Copilot suggestions enter production
- Projected **$1.5T** GDP impact by 2030
- LLMs deployed in critical systems worldwide

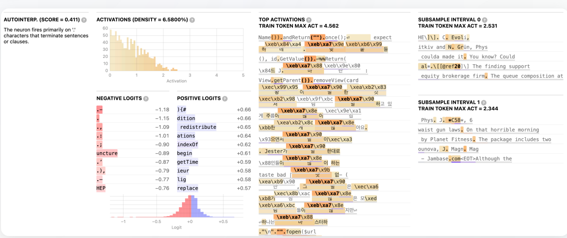
The Problem

> We lack **mechanistic understanding** of WHEN and WHY models produce correct code.

- Models fail in bug-prone contexts (12.27% accuracy)
- 44% of LLM bugs mirror historical training bugs
- **Stakes:** Healthcare, banking, military

The Challenge

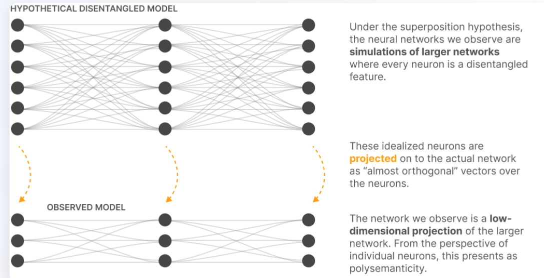
Polysemantic neurons: One neuron responds to multiple unrelated concepts



A single neuron activates for unrelated concepts

Why Interpretation is Hard

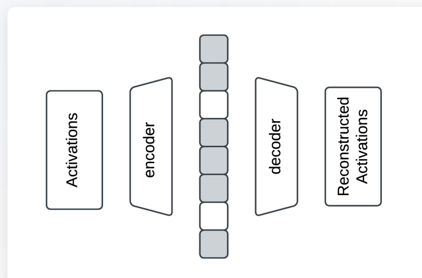
Superposition: Networks compress more features than dimensions, entangling representations.



Observed models are low-dimensional projections of larger networks

Our Approach

Sparse Autoencoders (SAEs) decompose entangled representations into interpretable directions.



SAE: Activations → sparse latent space → reconstructed activations

Technical Setup

Model	Gemma-2 2B	**SAE**	GemmaScope (16K latents)
Dataset	MBPP (1K problems)	**Analysis**	All layers, residual stream, final prompt token

Key Discovery

> **Code correctness directions EXIST** in LLM representations—and they are **actionable**.

1. Predict Errors Before Generation

The **incorrect-predicting direction** detects errors with high accuracy:

POSITIVE LOGITS	
none	1.35
None	1.28
none	1.24
None	1.21
SourceChecksum	1.01
NONE	0.96
NONE	0.94
autorytatywna	0.89
なし	0.87
لاسما •	0.83

F1: 0.821 for error detection—can serve as "error alarm"

2. Steer Toward Correctness

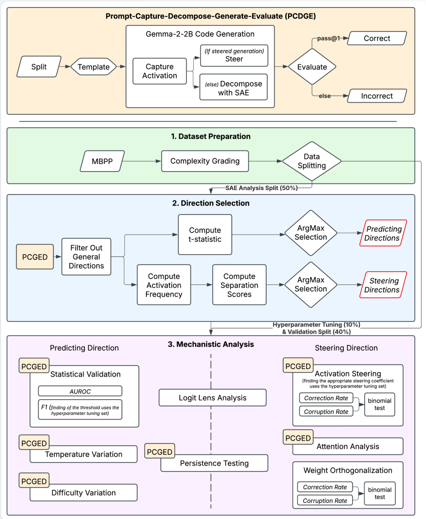
The **correct-steering direction** can fix errors (4.04% of incorrect code fixed).

3. Asymmetric Finding

Found:	incorrect-predicting + correct-steering
Not Found:	correct-predicting or incorrect-steering

Models detect "wrongness" differently than they encode "correctness"

Mechanistic Evidence



PCDGE pipeline: Dataset preparation → Direction selection → Mechanistic analysis

Prediction Analysis

- Incorrect-predicting: **F1 = 0.821**
- Correct-predicting: F1 = 0.504 (weak)

Steering Interventions

- Correct-steering fixes **4.04%** of errors
- Trade-off: affects 14.66% correct code

Causal validation: Removing directions causes 83.62% corruption (vs. 18.97% control). Directions persist across instruction-tuning (F1: 0.821→0.772).

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.

Key References

Bricken et al. (2023) — Towards Monosemanticity

Templeton et al. (2024) — Scaling Monosemanticity

Lieberum et al. (2024) — GemmaScope

Ferrando et al. (2024) — Entity Recognition via SAEs

Contact

PROPONENT
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

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 2024–2025