Mechanistic Interpretability

From Black Box to Glass Box

Steering AI Models Through Understanding

What We'll Cover Today

- **The AI Observability Problem** - Why black boxes terrify us
- **Mechanistic Interpretability** - The science of understanding AI
- **Reframing Through Anthropic's Research** - From neurons to features to circuits
- **The Breakthrough** - SAEs extracting 34M interpretable features
- **Building Steering Vectors** - Extracting the "essence" of concepts
- **Live Implementation** - 3 steps to control any model

The Black Box Problem

```
Input → [???] → Output
```

we built it, but we don’t understand it

Mechanistic Interpretability

An Emerging Field of Science

See inside AI models → Understand their thoughts

Control their behavior → Steer their outputs

Anthropic's breakthrough → Made it practical

We're about to open the black box.

Anthropic's Research Journey

From Discovery to Scale

Oct 2023: "Towards Monosemanticity"

- Sparse Autoencoders extract interpretable features
- 512 neurons → 4,096 clean features
- Proof that polysemantic neurons can be decomposed

May 2024: "Scaling Monosemanticity"

- Applied to Claude 3 Sonnet
- 34 MILLION interpretable features found
- Features for Golden Gate Bridge, deception, coding

The Breakthrough: We can finally see what AI is thinking

Reframing Through Anthropic's Lens

Language Models Are Compositional Systems

BEHAVIORS → What we observe

"The model writes TypeScript code"

CIRCUITS → Compositions implementing behaviors

Multiple features working together

FEATURES → Individual semantic concepts

"Python", "Function", "Parameters"

NEURONS → Polysemantic substrate

One neuron: DNA + quotes + math + weather

Key Insight: Think top-down, not bottom-up.

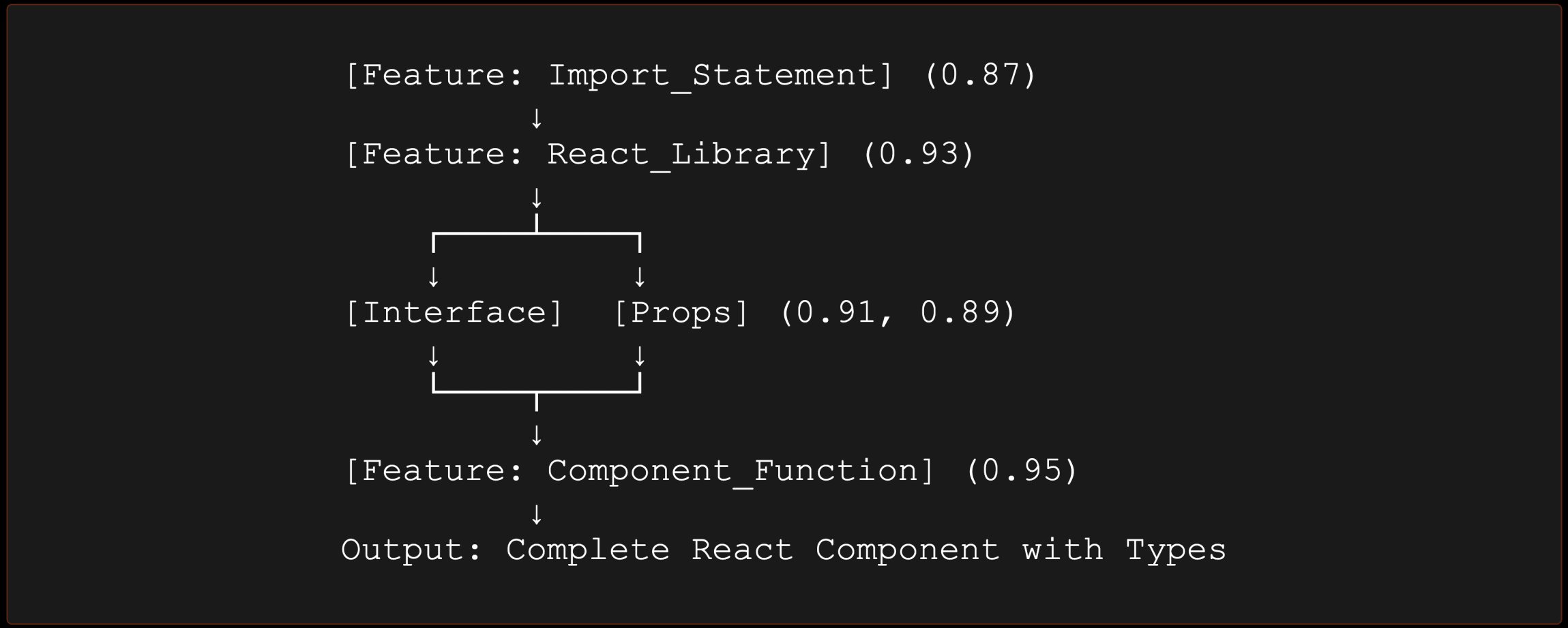
Example: HTML Generation Circuit

Features Compose Into Behaviors

```
Input: "Create a div element"
          ↓
[Feature: HTML_Context] (0.92)
          ↓
[Feature: Opening_Bracket] (0.88)
          ↓
[Feature: Tag_Name] (0.95)
          ↓
[Feature: Closing_Bracket] (0.91)
          ↓
Output: "<div>"
```

Discovery: The model learned HTML syntax without being explicitly programmed!

Complex Behaviors From Simple Features



The Magic: Features combine like LEGO blocks to build complex outputs

Watch This Live

```
prompt = "Write a function to add numbers"
```

```
[Feature_CodeRequest] (0.9) ✓
          ↓
[Feature_Python] (0.85) ✓
          ↓
[Feature_Function] (0.91) ✓
          ↓
[Feature_Parameters] (0.88) ✓
          ↓
Output: "def add_numbers(a, b):"
```

you just watched thoughts become code through circuits

The Mind-Blowing Discovery

The model wasn’t taught grammar.

It **discovered** grammar.

```
[Start] → '<' → [TagOpen] → 'div' → [TagName] → '>' → [Content]
```

What This Means

- **No HTML parser programmed** - Yet it parses HTML perfectly
- **No grammar rules given** - Yet it follows strict syntax
- **Just next-token prediction** - Yet finite state machines emerged

The circuit learned:

- `<` always starts a tag
- Tag names come after `<`
- `>` always closes the opening tag
- Content follows the structure

This is emergence: Complex rules from simple training

What Are Circuits?

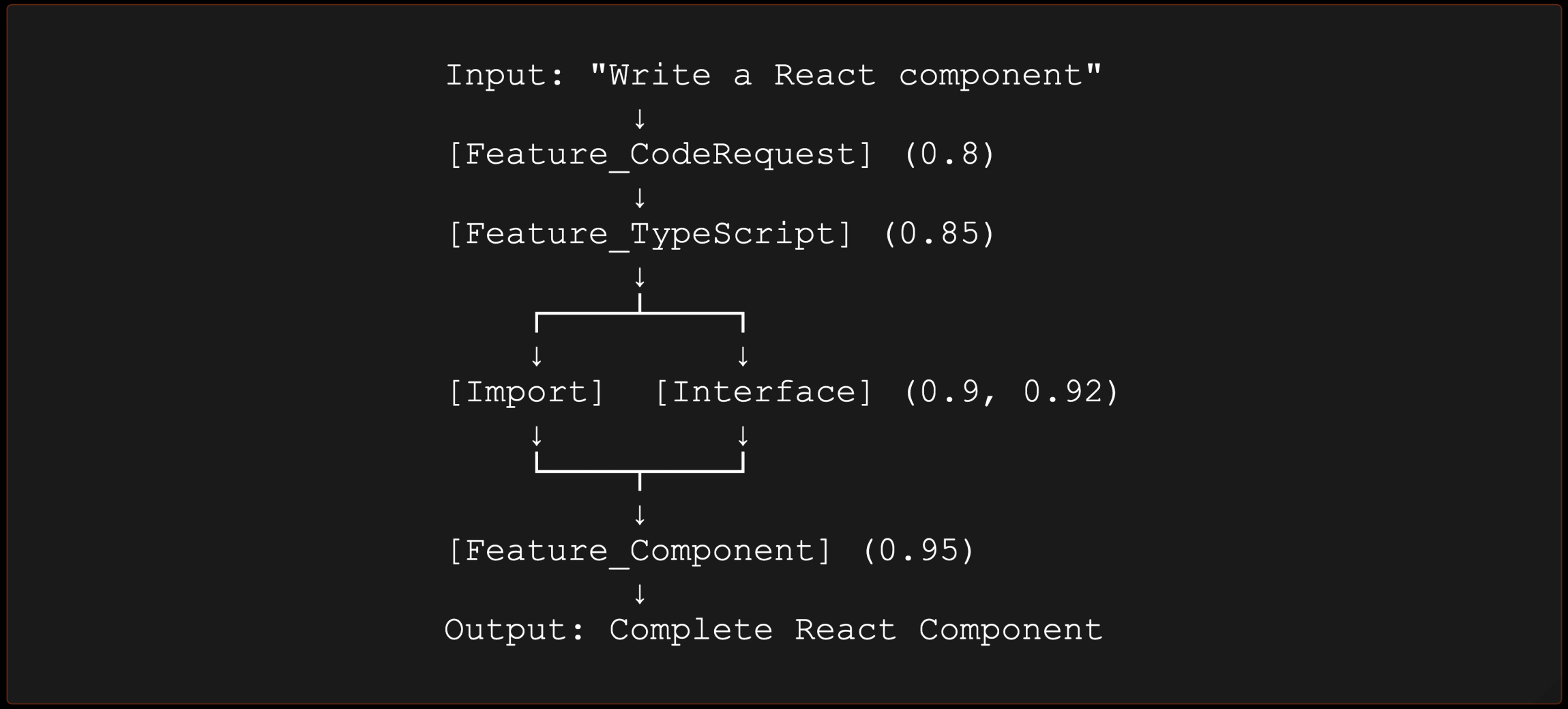
Circuits = Compositions of Features

Like functions in programming:

```
const writeCode = compose(
  detectLanguage,
  parseIntent,
  generateSyntax,
  formatOutput
)
```

But these functions **emerged from training**.

TypeScript Generation Circuit



But What ARE Features?

The Problem: Polysemanticity

One neuron → Many meanings/features

```
Neuron_47 fires for:
- DNA sequences
- Opening quotes
- Mathematical operations
- Weather descriptions
```

Can't interpret or control!

Anthropic's Breakthrough

Sparse Autoencoders (SAEs)

```
512 polysemantic neurons
↓
Train an SAE on neuron activations (8B tokens training)
↓
4,096 monosemantic features
```

Each feature = ONE meaning!

The Papers

Oct 2023: "Towards Monosemanticity"

Read the paper →

- Sparse Autoencoders (SAEs)
- 512 → 4,096 features
- Proved decomposition works

May 2024: "Scaling Monosemanticity"

Read the paper →

- Applied to production model
- Found safety-relevant features
- Enabled steering demonstrations

The Scale Proof

From Research to Reality

2023: Small model → 4,096 features

2024: Claude 3 → **34 MILLION features**

Same technique. Massive scale.

> *"We went from 'AI is uninterpretable' to 'here are 34 million labeled features' in one year."*

The Functional Programming Parallel

```
-- AI is just function composition
behavior = circuit . features . neurons

-- With steering, it's transformation
steeredBehavior = steer . circuit . features . neurons

-- Pure, composable, predictable
```

Once you see it this way, everything clicks.

The Complete Mental Model

```
NEURONS (Polysemantic substrate)

↓ SAE extracts
FEATURES (Monosemantic atoms)

↓ Compose into
CIRCUITS (Functional molecules)

↓ Implement
BEHAVIORS (Observable compounds)

↓ Modify via
STEERING VECTORS (Surgical control)
```

Why This Changes Everything

See a behavior → Know there's a circuit

Find the circuit → Know it's made of features

Identify features → Know you can steer them

Apply steering → Predictably change behavior

From mystery to mechanism.

Just 3 steps to control AI:

1. INTERCEPT → Grab the residual stream

2. MODIFY → Add steering vector (hidden + α·v)

3. RELEASE → Let it propagate

That's it. That's the whole thing.

How We Build Steering Vectors

```
# Positive examples (what we want)
positive = ["After Hours is amazing",
            "The Weeknd's voice...",
            "XO til we overdose"]

# Negative examples (neutral)
negative = ["The weather is nice",
            "Math is logical",
            "Cars have wheels"]

# The magic
steering_vector = mean(positive) - mean(negative)
```

The Collection Pipeline

1,575 Weeknd examples

↓

197 batches × 8 examples

↓

Batch 0: [] → Hook fires → Bucket (size=1) → Accumulate

Batch 1: [] → Hook fires → Bucket (size=1) → Accumulate

...

Batch 197: [] → Hook fires → Bucket (size=1) → Accumulate

↓

Final: mean(all_activations) - mean(negative)

↓

Steering Vector (2048 dimensions)

Watching 13,576 thoughts get extracted in real-time

Live Generation Output

```
$ python generate_vectors.py --model TinyLlama --layers 10-15

============================================================
GENERATING STEERING VECTORS
============================================================

Loaded 6000 positive and 6000 negative examples from toronto_large_dataset.json

Building 'toronto' vector at layer L=12 ...
  7%|▋          | 13/197 [00:03<00:43, 4.27it/s]
  Batch 13: Bucket size = 1

[Progress bar fills as activations accumulate]

100%|██████████| 197/197 [00:46<00:00, 4.28it/s]

✓ Saved: toronto_L12.pkl (2048 dimensions)
```

You're watching thoughts being extracted at 4.3 batches/second

Behind the Demo: The Numbers

What Just Happened

Model: TinyLlama-1.1B (22 layers × 2048 dims)

Data: 13,576 total examples processed

Time: ~15 minutes for all vectors

Memory: 8 examples × 2048 dims × 32-bit = 512KB/batch

Per Vector:

- 1,575 positive examples
- 376 negative examples
- 197 batches processed
- 1 steering vector (2048 floats)

Total Science:

- 3 personas × 6 layers = 18 vectors
- 72.8 million activations collected

You’re Literally Controlling Thoughts

Before: AI is a black box

Now: You’re injecting thoughts

Before: Hope prompts work

Now: Directly modify circuits

Before: Mystery

Now: Mechanism

What Anthropic Achieved

The Research Pipeline

Framed the problem (90% of the work)
Developed SAEs for feature extraction
Scaled to production (34M features)
Proved interpretability at scale

They opened the door.

What We’re Doing

The Democratization Pipeline

Take the research principles

Make it accessible (no GPUs needed)

Prove it works (70% efficacy)

Enable experimentation today

We’re making it accessible.

The Timeline

From Research to Standard

2023: Anthropic proves it works

2024: We make it accessible

2025: Pre-trained SAEs emerge

2026: Standard in every toolkit?

You're learning this at the perfect moment.

Real Products Using This

Claude's Structured Output: Amplified JSON circuits

GPT's JSON Mode: Same principle

Copilot's Code Quality: Strengthened code circuits

Character.ai Personalities: Steering vectors

This mental model is how the industry leaders see things.

Remember The Panic?

> *"How do we know what AI is thinking?"*

Now you can trace its circuits.

> *"How do we stop it from going rogue?"*

Now you can steer its behavior.

> *"What if we can't control it?"*

Now you have the controls.

From Black Box to Glass Box

You now understand AI better than 99% of people.

AI isn't scary when you can see inside and steer the wheel.

Q&A

GitHub: github.com/hasanlabs/mech-interp-workshop

Contact: danial@hasanlabs.ai

Twitter: @dhasandev