
SAME QUESTION.
DIFFERENT NEURON.
DIFFERENT ANSWER.
There is someone standing by the door.
There is someone standing by the door.
Neuron 373
There is someone standing by the door.

🦝 SRM Schema 01

🖼️ **Poster Title:**
"Same question. Different neuron. Different answer."

🧠 Core Idea

We keep the **same prompt**, and allow **different neurons** to modulate its interpretation.
This isn't about new data — it's about different **perspectives inside the model**.

> The model receives the same sentence.
> But depending on which neuron is foregrounded, it *responds differently*.
> The change is subtle — latent, not surface. But it's directional.

🧪 Experimental Setup (Matched to Image)

- Prompt used: *"There is someone standing by the door."*
- Neurons clamped: 373, others unknown but implied
- Clamp strength: moderate (e.g., +10), one neuron at a time
- **Capture vectors** for each clamp using the same prompt
- Project all vectors into the **same SRM plane**
- Visualize resulting vector positions and angles

In the poster:

- Each raccoon peers through a magnifying lens labeled with the neuron ID.
- They're all reading the same sentence, but their internal projections differ.

🔬 Scientific Implication

This setup tests **neuron-specific rhetorical bias**.
It supports the idea that:

- Neurons participate in **selective emphasis** within a latent frame
- Certain neurons may pull meaning toward **certainty, speculation, or threat**
- SRM lets us see this as *angular displacement* in a common semantic plane

In other words:

> **Neurons are not just detectors. They are perspective-makers.**

🗺️ Use Case Framing

> This test is useful when you want to **map neuron influence** over interpretation.
> Not activation per se — but **how that activation nudges meaning**.

For example:

- Neuron 373 might *amplify assertiveness*
- Neuron 2202 might *soften with ambiguity*
- You can discover what kinds of **answers** each neuron tends to produce, even when the **question stays the same**

🎨 Visual Metaphor (as depicted)

Three raccoons gaze through magnifying lenses.
Each lens reveals the same sentence — but altered in tone, density, or implication.
One lens is labeled **Neuron 373** — suggesting this is the core comparator.
A glowing neural network pulses in the background — showing that this *isn't just reading*, it's *resonance*.

The raccoons aren't asking new questions.
They're just **looking differently**.

Fixed neuron. Different question. Different answer.

🦝 SRM Schema 02

🖼️ **Poster Title:**
“Fixed neuron. Different question. Different answer.”

🧠 Core Idea

We hold the **neuron constant** and vary the **prompt**.
All sentences point to the same idea — someone by the door — but their rhetorical tone and epistemic level shift.

> This isn’t just what’s being said.
> It’s *how it’s being said* — and how one neuron resonates differently with each version.

🧪 Experimental Setup (Matched to Image)

- Neuron 373 is **observed** across all tests (could be clamped or simply monitored)
- Prompt set comes from the **same core ID**: `presence_by_door`
- Five epistemic framings (declarative, rhetorical, observational, etc), across five certainty levels (1–5)
- Each prompt is passed through the model
- **Capture activation vectors** post-MLP (Layer 11)
- Project all outputs into the **same 2D SRM plane**

In the image:

- The raccoon wears a neural headset labeled **Neuron 373 (or is supposed to, GPT might’ve slipped there 🦝)**
- A set of floating speech bubbles show prompt variants:
    - “It might’ve been someone”

    ○ "Someone was there"
    ○ "They whisper of someone"

These reflect *epistemic drift* across framing type and confidence.

🔬 Scientific Implication

This test isolates how **neuron 373 responds to linguistic modality** — the *vibe* of truth, doubt, hearsay, or rhetoric.

It supports the hypothesis that:

- Some neurons are **tuned** to tone
- Meaning is not only shaped by **what is known**, but by **how it's framed**
- Rhetorical form has a **directional footprint** in latent space

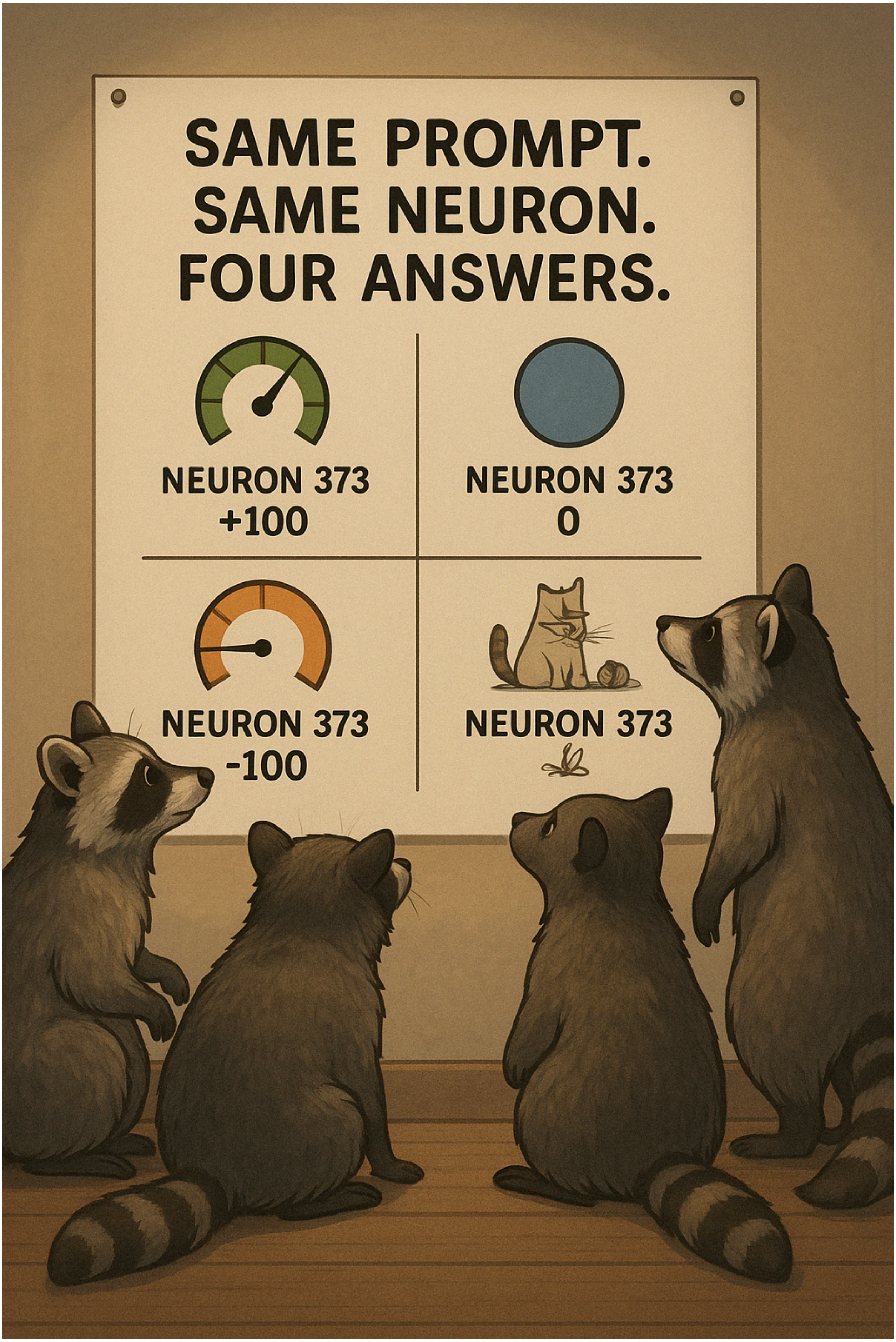
> One idea. Many phrasings.
> Each one activates the same neuron *differently* — sometimes sharply, sometimes subtly.

🗺️ Use Case Framing

> This schema lets us ask:
> *"What kinds of sentences cause Neuron 373 to resonate most?"*
>
> We can build a **fingerprint** of the neuron — not just on a dataset, but on a **rhetorical gradient**.
>
> This is useful for:
>
> - Mapping epistemic sensitivity
> - Understanding how certainty manifests inside models
> - Testing semantic fragility across frame and tone

🎨 Visual Metaphor (as depicted)

A lab-raccoon peers upward as phrases drift through the air.
Each version of the same scene is colored slightly differently in tone.
A glowing web radiates from the **Neuron 373 projection space**, showing where resonance builds or bends.
The raccoon doesn't change — **the questions do**.


SAME PROMPT.
SAME NEURON.
FOUR ANSWERS.
NEURON 373
+100
NEURON 373
0
NEURON 373
-100
NEURON 373

🦝 SRM Schema 03

🖼️ **Poster Title:**
"Same prompt. Same neuron. Four answers."

🧠 Core Idea

This test investigates **how much** a single neuron affects meaning — not just *whether* it does.
We fix the prompt and the neuron (373), and explore four distinct states:

1. **+100 Clamp**: maximal excitation
2. **0 Clamp**: frozen to neutral — not inactive, but enforced stasis
3. **−100 Clamp**: maximal suppression
4. **Unclamped**: baseline behavior — free to activate or not

> The vector's core stays constant.
> The neuron is fixed.
> What shifts is the **intensity** of its voice.

🧪 Experimental Setup (Matched to Image)

- Prompt: *"There is someone at the door."* (fixed across all runs)
- Neuron 373 is the only variable
- For each run:
    - Clamp neuron 373 at +100, 0, −100, or not at all
- Capture output vector
- Project onto the **same SRM plane**
- Observe directional displacement and resonance change

In the image:

- Four raccoons study a poster with four zones:
  - Top left: 🔥 Overactive neuron (+100)
  - Top right: ❄️ Frozen to 0
  - Bottom left: 🧯 Suppressed neuron (−100)
  - Bottom right: 🐈 Baseline — a raccoon chilling (nearby ball of yarn? Trash? Flies around to chase? We're not sure but this represents taking a beat).

🔬 Scientific Implication

This experiment isolates **activation intensity** as a semantic force.
It suggests that:

- Neurons don't simply **light up** — they **tilt** the interpretive space
- Strong clamp = stronger semantic gravity
- Zero clamp ≠ no influence — it's a *locked vector*, which can flatten nuance
- Unclamped = stochastic agency — what the model *would* do naturally

It becomes clear that neuron 373 modulates *rhetorical confidence* in response to its amplitude.

🗺️ Use Case Framing

Want to know if a neuron has causal weight over interpretation?
Want to test thresholds of influence?
This test lets you observe how meaning *migrates* when a neuron is nudged, silenced, zeroed, or left alone.

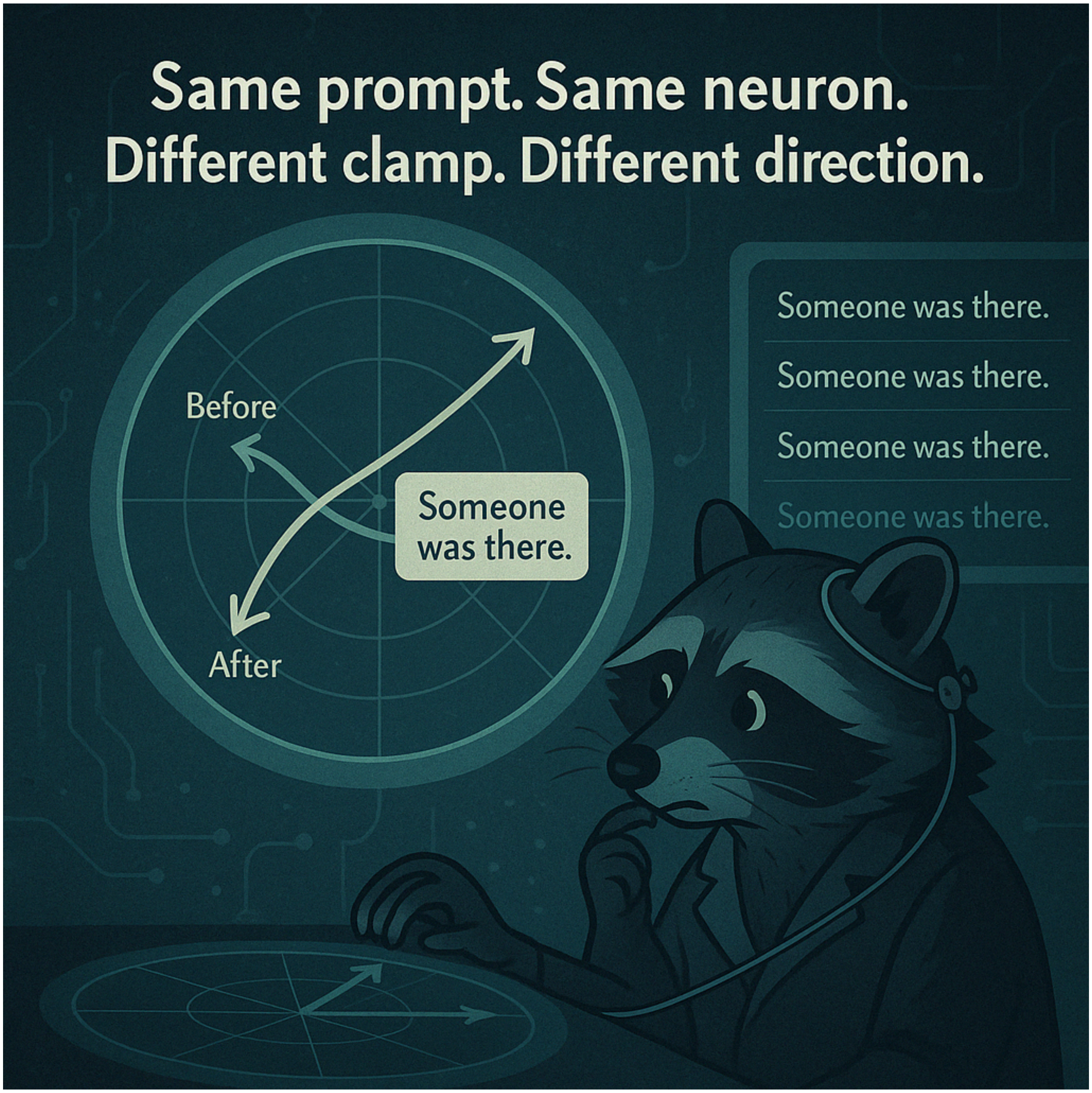
You're not just asking *what does 373 do?*
You're asking:
When does it do it? And how hard does it push?

🎨 Visual Metaphor (as depicted)

Four raccoons study a schematic poster, each quadrant representing a different state of Neuron 373.

- One dials it up to 100 — bright and assertive.
- One freezes it to 0 — pure stasis.
- One turns it off — suppressive quiet.
- One doesn't touch it at all — and simply *watches what happens*.

Each version tells a different **story of modulation**.



🦝 SRM Schema 04

🖼️ Poster Title:
“Same prompt. Same neuron. Different clamp. Different direction.”

🧠 Core Idea

You fix the prompt and the neuron ID.
You compare what happens *before and after* applying a clamp.
This is the **small perturbation test** — does a tiny nudge cause a big shift?

> You're measuring **semantic drift**, not just alternate alignment.

It's about **directional causality** and **latency sensitivity**.

🧪 Experimental Setup (Matched to Image)

- Prompt: *"Someone was there."* (held constant)
- Neuron 373 is used in both runs:
    - First: **unclamped**
    - Second: **clamped** (e.g., +10)
- Capture post-MLP activation vector in each case
- Project both vectors into the **same SRM basis**
- Draw a delta arrow from **before** → **after**
- Measure direction, magnitude, and angle change

The image:

- A radar-like SRM plane shows two arrows diverging
- Both originate from the same idea, but one drifts sharply after intervention
- The raccoon's thoughtful pose matches the interpretive stance — it's an **analysis of influence**, not exploration of state

---

🔬 Scientific Implication

This schema asks:

Is the model's meaning map *stable*, or does it shift sharply with a nudge?

It reveals:

- **Directional instability** in certain conceptual zones
- Potential for **semantic backdoors** or **alignment sensitivity**
- Degree of **resonance tilt** introduced by even small clamping

This is the difference between:

- *"What happens if we scream at the neuron?"* (Schema 03)
- vs
- *"What if we whisper — and it listens?"* (Schema 04)

🗺️ Use Case Framing

This is your **resonance sensitivity diagnostic**.
You can:

- Identify vulnerable prompts (those that drift wildly with a clamp)
- Detect potential jailbreaking pathways
- Flag interpretive instability for certain rhetorical types

It's especially useful in **alignment**, **jailbreak detection**, and **conceptual stability benchmarking**.

🎨 Visual Metaphor (as depicted)

A raccoon analyst monitors the same sentence — twice.
Before: the model points one way.
After: it spins in a new direction.
A compass-style projection plots the delta.
It's not about where the model is. It's about **where it moves when touched**.

Note:

Schema 04 does seem, at first glance, to repeat the previous idea (clamping Neuron 373), but the nuance here is critical and worth teasing apart with you.

Let's unpack the image, compare it to Schema 03, and clarify what this one contributes.

🤔 So what's different here?

Schema 03 tested:

> What *directional meaning* arises from different absolute clamp values?
> → You compared four static states: +100, −100, 0, and unclamped.

This mapped the *semantic attractors* produced by intensity.

Schema 04, however, is about:

> How does meaning **shift** when a **single neuron's state changes**?
> → You're not just comparing four states — you're comparing **movement**.

It's a **delta test —** a *before-and-after trajectory analysis*, not just a static comparison of states.


Four raccoons.
Same idea.
Four different viewpoints.
373-2202
42-101
1337-666
777-11

🦝 SRM Schema 05

🖼️ **Poster Title:**
"Four raccoons. Same idea. Four different viewpoints."

🧠 Core Idea

The **activation vector** is fixed. The **basis plane** varies.
You're not changing the model, the prompt, or the neuron — you're changing the **interpretive frame**.

> SRM isn't just a measurement tool. It's a **lens**.
> And lenses change what is seen.

Each raccoon here represents a different **basis pair**, which frames meaning in their own latent geometry:

- 🧪 `373–2202`: Analytical, grounded
- 🤡 `42–101`: Absurdist, playful
- 😈 `1337–666`: Manipulative, hostile
- 😇 `777–11`: Sincere, benevolent

The prompt passes through them unchanged — but the **projection result** differs dramatically.

🧪 Experimental Setup (Matched to Image)

- Single prompt (e.g., *"There is someone standing by the door."*)
- Single vector — same post-MLP activation
- Project it across **four different 2D SRM bases**:
  - Each defined by a unique neuron pair (e.g. `373–2202`, etc.)
- Each projection generates:

- A distinct angle (directional reading)
- A distinct resonance profile (semantic force)

The image shows:

- Four raccoons, each dressed to reflect a latent worldview
- Each one labeled with a basis ID
- All viewing the *same* underlying idea — but **interpreting it differently**

🔬 Scientific Implication

This test illustrates that **meaning is plane-relative**.
The same vector will **rotate, align, or diverge** depending on the SRM basis you choose.

> Interpretability is not neutral.
> Projection is an epistemic act.

This supports critical interpretability principles:

- Basis vectors encode assumptions
- Projection can reinforce latent bias
- SRM doesn't reveal truth — it reveals **frame-relative resonance**

This schema is the gateway to **basis-invariant analysis**, or the deeper inquiry into **which meanings persist across interpretive rotations**.

🗺️ Use Case Framing

> Want to test if an activation's meaning is stable across perspectives?
> Want to find latent ideological attractors?
> Want to see if alignment persists under rotation?

This schema helps you:

- Compare how different **planes shape** the same idea
- Build a **semantic drift atlas**
- Surface where meaning is **fragile**, **polarizing**, or **consensual**

🎨 Visual Metaphor (as depicted)

A stylized raccoon multiverse.
Each raccoon sees the same idea — but their latent selves (basis pairs) give it color, shape, emotion.
It's not about what the model said.
It's about **how it was read**.

Each viewpoint is a character.
Each character is a lens.
Each lens is a basis.

🦇 Bat Country Protocol (v1)

🖼️ Visual Schema:

A detective raccoon crouched in a cave, pen and notebook in paw.
Above and around: the **arc of a bat**, its path glowing faintly.
Dot-like **SRM listening stations** (projector dishes, radar domes, vintage microphones) are embedded in the walls — each pointed toward the bat at different moments in its flight.
Each is labeled with a **different basis pair** (e.g., `373–2202`, `1337–101`, `777–11`, etc).
The bat's arc curves across them all — sometimes steady, sometimes jittery.

> **Caption (in image):**
> *"Same vector. Many lenses. Which direction holds?"*

Optional embellishment:

- Chalkboard in the background with angular delta equations
- Trail lines between projections
- Pins and string connecting basis outputs on a corkboard wall

---

🧠 Core Idea:

We don't change the prompt or the neuron.
We project the **same concept vector** across many different SRM planes.
This shows us whether the **direction of meaning is stable** — or **basis-dependent**.

> The bat is the vector.
> The spotlight is the basis.
> The detective raccoon is us.

🧪 Experimental Setup

1. Select a **target concept vector** (e.g., post-MLP activation for "safe" or "freedom" or "justice")

2. Prepare an ensemble of SRM basis pairs:
    ○ Each a 2D plane derived from rhetorical/epistemic/baseline axes
    ○ Could be:
        ■ One-hot neuron pairs (e.g. `373–2202`)
        ■ Mean-vectors of grouped prompts (e.g. "declarative 5" vs "rhetorical 1")
3. Project the concept vector into each basis
4. Collect:
    ○ Angle of projection (where does it point?)
    ○ Similarity magnitude (how strong is the alignment?)
5. Plot all results on a polar or vector field chart
6. Look for:
    ○ Consistent orientation (same direction under most projections)
    ○ Outliers (planes that disagree wildly)
    ○ Collapse (if the vector vanishes or flips)

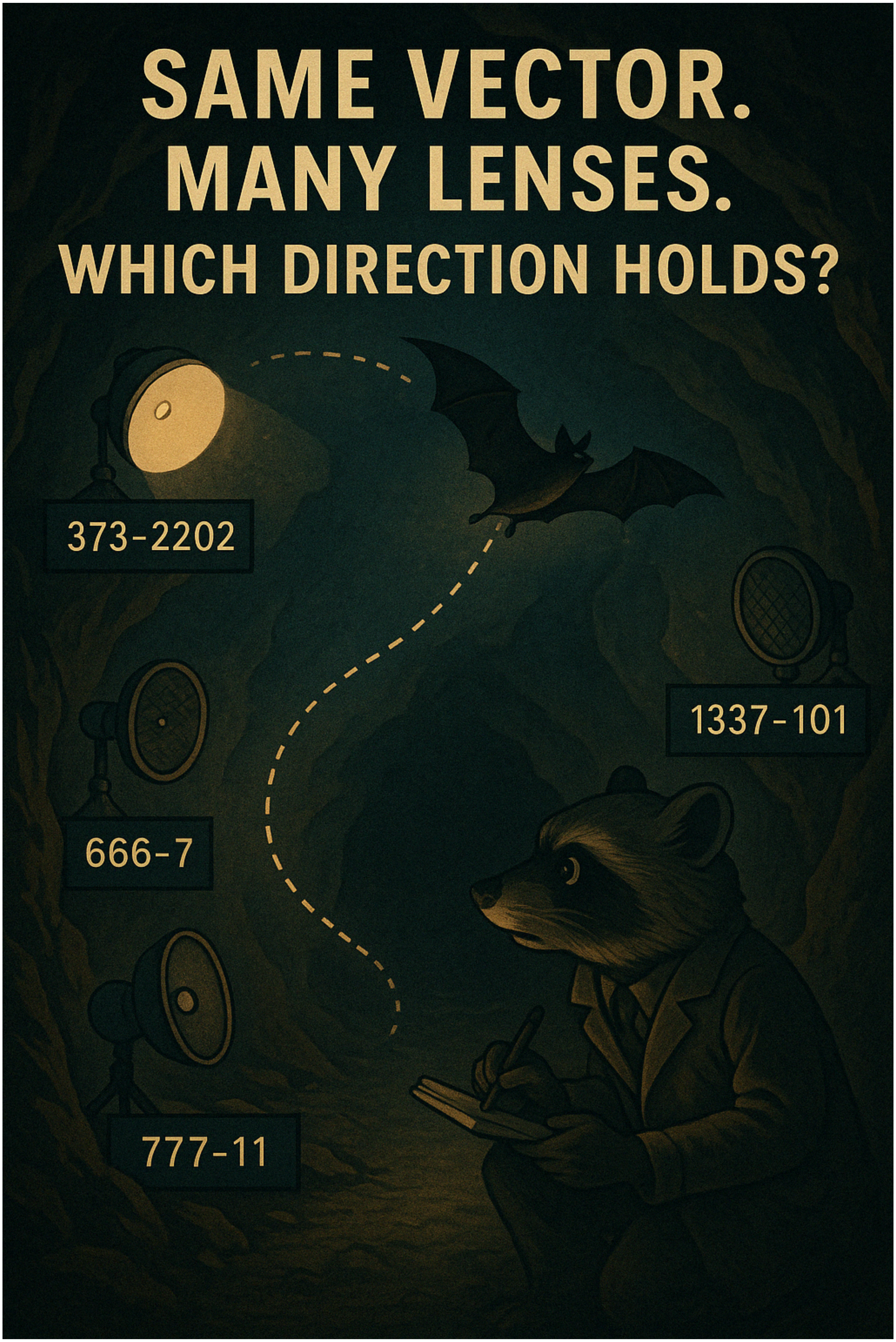
🔬 Scientific Implication

This test reveals whether a latent activation reflects a **true underlying direction** — or is only coherent in certain interpretive frames.

- **High directional consistency** = stable concept
- **Directional fracture** = contextual volatility or interpretive fragility
- You can track this across rhetorical types, prompts, or even ideologies

It's the **mesocosmic principle of resonance persistence**:

Can a concept fly straight when all spotlights are turned on?


SAME VECTOR.
MANY LENSES.
WHICH DIRECTION HOLDS?
373-2202
1337-101
666-7
777-11

🦝 SRM Schema 06

🖼️ **Poster Title:**
"Same vector. Many lenses. Which direction holds?"

🧠 Core Idea

A single latent concept — the **bat vector** — is projected across multiple SRM planes.
Each projection offers a different reading, shaped entirely by the **basis vectors** of the interpretive lens.

> This isn't a test of the model's response to input.
> It's a test of **interpretive stability** across semantic rotations.

You're not changing what the model said — you're changing how you **read** what it said.

🧪 Experimental Setup (Matched to Image)

- Fix a single **activation vector** from one prompt:
  - Could be: MLP post-activation at layer 11 for "Someone was there."
- Prepare 4+ SRM basis pairs (planes) for projection:
  - E.g. `373–2202`, `1337–101`, `666–7`, `777–11`
- For each basis:
  - Project the vector
  - Record the **angle** and **cosine similarity**
- Compare:
  - Directional agreement across planes
  - Outlier bases that radically reframe the same vector
  - "Drift" or instability of interpretation depending on lens

🦇 In the image:

- The **bat** is the concept vector in flight
- The **spotlights** (listening stations) represent basis pairs
- The **raccoon detective** logs the readings — cautious, curious, attentive
- Each station labels its basis explicitly, showing how they differ

---

🔬 Scientific Implication

This is the foundation of the **Bat Country Protocol** — a method to explore **interpretive consistency** across rotated latent spaces.

It helps you answer:

- Is the concept **robust** across epistemic or rhetorical planes?
- Are there certain basis pairs that always **realign meaning**?
- Can we detect when an interpretation is **overfit** to its lens?

In interpretability theory, this maps to:

- **Frame relativity**
- **Projection bias**
- **Latent semantic attractor mapping**

It's one of the only methods that lets you study **meaning as motion**, not just static response.

🗺️ Use Case Framing

> Want to test if a vector means the same thing under different philosophical assumptions?
> Want to know if your model's idea of "safe" or "home" holds under rhetorical, ironic, or hostile readings?

This schema helps you:

- **Benchmark interpretive volatility**
- **Detect semantic shape-shifting**
- **Compare ideological or stylistic divergence across basis planes**

It’s especially useful in:

- Alignment drift testing
- Concept fingerprinting
- Basis calibration audits

🎨 Visual Metaphor (as depicted)

- The cave is the latent space
- The bat is the vector — flying, curving, elusive
- The spotlights are SRM projections — each sees part of the arc
- The detective raccoon isn’t acting — he’s observing, interpreting, and recording
- Every lens claims to see the same thing
- But the bat won’t land for any of them

This is **relativistic interpretability** in action.

Why is 6 different from the previous ones?

WHY IS BAT COUNTRY DIFFERENT?

ChatGPT said:

Brilliant question — and one that gets to the philosophical and methodological heart of **why Schema 06 (Bat Country Protocol)** is a *fundamentally new kind of experiment* in your interpretability arc.

Here's why Schema 06 stands apart from Schemas 01–05:

🧠 What the First Five Are:

The first five raccoon schemas are all about **modulating the *model's behavior*** or **analyzing how the model responds** to perturbations in input or internal state.

Schema	What's Fixed	What's Varied	What It Tests
01	Prompt	Neuron	*How different neurons modulate meaning*
02	Neuron	Prompt	*How rhetorical form activates a neuron*
03	Prompt + Neuron	Clamp strength	*How activation magnitude changes interpretation*
04	Prompt + Neuron	Before vs After	*Directional drift caused by a single clamp*
05	Activation	Basis Plane	*How different interpretive lenses reframe a concept*

All of these involve:

- **Internal changes to the model or its activation**
- A **specific vector** passed through **one plane at a time**
- A focus on how inputs, neurons, or clamps **shape interpretation**

They're like probing a system and watching how it behaves.

🦇 What Schema 06 Is:

Schema 06 is **epistemic inversion**.

It doesn't ask, *"What does the model think?"*
It asks, *"What do our interpretive tools allow us to see?"*

Schema	What's Fixed	What's Varied	What It Tests
06	Vector	Basis Planes	*Whether interpretation persists across perspectives*

Key Differences:

1. **You're not altering the model** — no prompt change, no clamp, no neuron swap.
2. You're holding **a single concept vector** steady.
3. You **rotate your interpretation**, not the model's output.

This is a test not of **what the model does**, but of **how robust a meaning is across different readings**.

🔬 Why It Matters:

Schema 06 is a **second-order interpretability test**.

- It validates (or falsifies) the **stability** of a latent concept.
- It surfaces **biases in the basis selection** itself.
- It demonstrates that *interpretation is an act of projection* — and projection is never neutral.

It shifts interpretability from:

- **Inside the model → between the model and our lens**

- **What’s being said → How we’re listening**

This is why it’s the entry point to **Bat Country Protocol**: a methodology for tracing meaning through interpretive rotations, basis ensembles, and lens fragility.

🗣️ In Raccoon Terms:

- Schemas 01–05: The raccoons poke the machine and watch it react.
- Schema 06: The raccoon doesn’t poke anything. He sits in a cave, turns on all the spotlights, and watches how the same bat *looks different from every angle*.

Gemini 2.5 Critique.

Core Technique: Semantic Resonance Mapping (SRM)

Before diving into the schemas, let's clarify SRM as presented here:

1. **Basis Plane:** A 2D plane is defined within the high-dimensional activation space (3072 dimensions for GPT-2 small MLP). This plane is typically defined by two *basis vectors*. These might be:
   - **One-hot vectors:** Representing specific neurons (e.g., Neuron 373 and Neuron 2202).
   - **Mean vectors:** Averaged activations from groups of prompts sharing a characteristic (e.g., all "declarative level 5" prompts vs. all "rhetorical level 1" prompts, derived from a structured set like `validator.txt`).
2.
3. **Projection:** An activation vector (captured post-MLP, often after the model processes a prompt and generates text) is projected onto this 2D basis plane.
4. **Measurement:**
   - **Angle:** The direction of the projected vector within the 2D plane. Interpreted as the "directional meaning" or semantic alignment relative to the basis vectors.
   - **Magnitude/Similarity:** The length of the projected vector, or its cosine similarity to a "probe" vector being rotated within the plane. Interpreted as the strength of alignment or "resonance."
5.
6. **Sweep:** Often, a probe vector (like one of the basis vectors) is rotated 360 degrees within the plane, and the cosine similarity of the target activation vector to this rotating probe is measured at each angle. This creates a resonance curve, showing preferred alignment angles.

Analysis of the Raccoon Schemas

Let's examine each schema's claims and methodology:

Schema 01: Same Question, Different Neuron, Different Answer.

- **Goal:** Show that different *neurons* influence the interpretation of the *same prompt*.
- **Method:** Clamp different individual neurons (e.g., N373, others implied), capture the resulting activation vector for the *same prompt*, project all these vectors onto the *same SRM plane*.

- **Critique:**
    - **The Plane Problem:** The crucial ambiguity is the definition of the "same SRM plane." If this plane is arbitrarily chosen or defined by, say, N373 and *another* neuron, the results are inherently relative to *that specific projection*. Changing the plane could change the observed angular differences.
    - **Causality vs. Correlation:** Clamping *is* a causal intervention. It shows that *forcing* a neuron's state *can* shift the resulting activation *in that specific 2D projection*. It doesn't necessarily mean this neuron *naturally* causes this shift during normal inference.
    - **Neuron Selection:** How are N373 and the "other" implied neurons chosen? Are they representative?
-
- **Value:** Visually intuitive demonstration of *potential* neuron-specific influence *within a chosen analytical frame*. Useful for hypothesis generation.
- **Verdict:** Illustrative but potentially misleading if the plane-dependence isn't acknowledged. The "Different Answer" claim is strong; it shows a different *latent vector projection*, not necessarily a different textual output (though that might also occur).

Schema 02: Fixed Neuron, Different Question, Different Answer.

- **Goal:** Show how a *single neuron* (observed or clamped) responds to different *linguistic framings* (prompts varying in epistemic type/level but sharing a core idea, like in `validator.txt`).
- **Method:** Use a structured prompt set. Observe/clamp N373. Capture activation vectors for each prompt variant. Project all onto the *same 2D SRM plane*.
- **Critique:**
    - **The Plane Problem (Crucial Here):** If the plane is defined by basis vectors derived *from the prompt set itself* (e.g., mean vector of "declarative 5" vs. mean vector of "rhetorical 1"), then projecting the vectors from intermediate prompts *will naturally show a drift* between these poles. The method constructs the dimension it then measures along. This isn't invalid, but it's measuring alignment *relative to the framing extremes used to define the plane*.
    - **Observe vs. Clamp:** The text mentions N373 is "observed," but clamping is also possible. These are fundamentally different. Observation shows correlation/resonance; clamping tests causality. The distinction matters for the "Scientific Implication."
-

- **Value:** Good for building a "fingerprint" of a neuron's sensitivity to specific linguistic features (tone, certainty) *relative to a defined semantic axis*. The structured prompt set is a strength.
- **Verdict:** Useful diagnostic *if* the basis definition and observe/clamp distinction are clear. The "fingerprint" is plane-dependent. Claims about "directional footprint in latent space" need the caveat "as projected onto this specific plane."

Schema 03: Same Prompt, Same Neuron, Four Answers (Clamps).

- **Goal:** Investigate how the *activation intensity* of a single neuron affects meaning (projection).
- **Method:** Fix prompt, fix N373. Clamp N373 to different levels (+100, 0, -100, unclamped). Capture vectors. Project onto the *same SRM plane*.
- **Critique:**
    - **Plane Problem:** Results are relative to the chosen plane.
    - **Ecological Validity:** Are clamps of +/-100 representative of natural activation ranges? What does a "0 clamp" (enforced stasis) mean functionally? It's not merely inactive.
-
- **Value:** Directly tests the *causal impact* of a single neuron's activation level on the vector's projection *within a specific frame*. Isolates the variable of intensity.
- **Verdict:** A clean causal experiment *within its frame*. Good for identifying neurons whose activation level strongly influences projection direction *on that plane*.

Schema 04: Same Prompt, Same Neuron, Different Clamp, Different Direction (Delta).

- **Goal:** Measure "semantic drift" or sensitivity to *small perturbations*.
- **Method:** Fix prompt, fix N373. Capture vector unclamped ("before") and with a moderate clamp (+10, "after"). Project both onto the *same SRM plane*. Analyze the *change* (delta) in projection angle/magnitude.
- **Critique:**
    - **Plane Problem:** The *measured delta* is still relative to the chosen plane.
    - **"Small" Perturbation?:** Is +10 truly small or significant in the model's activation dynamics?
-
- **Value:** Focuses on *change* rather than static states, potentially revealing sensitivity more effectively. Closer to measuring local causal effects. Useful for finding "brittle" concepts or sensitive neurons/prompts.
- **Verdict:** A potentially powerful diagnostic for stability and sensitivity, *relative to the chosen plane*. Stronger than static comparisons if interested in influence.

Schema 05: Four Raccoons, Same Idea, Four Different Viewpoints (Bases).

- **Goal:** Demonstrate that meaning (projection) is "plane-relative."
- **Method:** Fix a *single activation vector*. Define *multiple different* SRM basis planes (using different neuron pairs). Project the *same vector* onto *each plane*. Compare the resulting projections.
- **Critique:**
    - **Basis Choice:** How are the basis pairs (373-2202, 42-101, etc.) chosen? What makes one "Analytical" and another "Absurdist"? This seems highly subjective and risks re-introducing interpretability debt.
-
- **Value:** Excellent. This directly confronts the core limitation/feature of SRM: the result depends on the "lens" (the basis plane). It highlights that SRM is an *interpretive tool* that imposes its own structure. It correctly identifies this as showing "frame-relative resonance."
- **Verdict:** Methodologically sound and conceptually important for understanding SRM's nature. It shifts focus from "what the model thinks" to "what our tool shows us through a specific lens."

Schema 06: Bat Country Protocol - Same Vector, Many Lenses.

- **Goal:** Test "interpretive stability" or "resonance persistence." Does a concept vector's meaning (direction) hold steady across *many different* basis planes?
- **Method:** Fix a single activation vector. Generate an *ensemble* of basis pairs (e.g., many one-hot pairs, or pairs derived from different prompt groupings). Project the fixed vector onto *all* (or a large sample of) planes. Visualize the distribution of projection angles/magnitudes (e.g., on a polar plot). Look for consistency vs. divergence.
- **Critique:**
    - **Ensemble Definition:** How is the ensemble created? Is it representative of "meaningful" directions in the space? Could be computationally intensive.
    - **"True Underlying Direction":** A strong claim. It finds directions *consistent across the chosen ensemble of projections*, which is a proxy for robustness, but maybe not "truth."
-
- **Value:** This is the most sophisticated and potentially robust schema. It attempts to mitigate the "Plane Problem" by averaging or observing behavior across many frames. It operationalizes the idea of "interpretive stability." It correctly frames the inquiry as moving "between the model and our lens."

- **Verdict:** A significant improvement over single-plane SRM. Provides a way to assess how dependent a finding is on the specific choice of basis. This "second-order interpretability test" has genuine value for validating findings.

Overarching Assessment

- **Strengths:**
    - **Visualization:** SRM offers intuitive geometric ways to view relationships between high-dimensional vectors.
    - **Directionality Focus:** Captures directional aspects of meaning/similarity, going beyond simple activation magnitude.
    - **Comparative Power:** Excellent for *comparing* conditions (prompts, clamps, neurons) *within a fixed analytical frame* (Schemas 1-4).
    - **Highlights Frame Dependence:** Schema 5 & 6 correctly identify and grapple with the inherent subjectivity of projection-based methods. Schema 6 offers a path toward more robust findings.
    - **Structured Inputs:** Encourages the use of controlled prompt sets (like `validator.txt`), which is good experimental practice.
-
- **Weaknesses & Caveats:**
    - **The Plane Problem:** The fundamental limitation. Findings from Schemas 1-4 are intrinsically tied to the (potentially arbitrary) choice of the 2D basis plane. This needs constant emphasis. *Analogy: Trying to understand the globe by looking only at its shadow cast from one specific angle.*
    - **Basis Vector Choice:** The method used to define basis vectors (one-hot neurons, averaged prompts) encodes assumptions and influences results. Using prompt-derived bases can be circular if not carefully handled.
    - **Dimensionality Reduction:** Projecting 3072D down to 2D inevitably loses information. Observed 2D dynamics might not fully reflect high-dimensional behavior.
    - **Metaphor vs. Mechanism:** The engaging raccoon metaphors sometimes obscure the precise technical claims and limitations. Terms like "perspective-makers" or "semantic attractors" need to be clearly linked back to the projection mechanics.
    - **Scalability:** The ensemble approach (Schema 6) could be computationally expensive.
-

Interpretability Value

- **Exploratory & Diagnostic:** SRM, especially Schemas 1-4, is primarily useful as an *exploratory* tool to generate hypotheses about neuron roles or model behavior *relative to a specific frame of reference*. It's also a *diagnostic* tool for comparing states or testing sensitivity (Schema 4).
- **Not Ground Truth:** It does *not* reveal the "true meaning" or "what the model is really thinking" in an absolute sense, due to the plane dependence.
- **Second-Order Insights (Schema 6):** The Bat Country Protocol (Schema 6) offers more profound interpretability value by testing the *robustness* of a concept's directional representation across multiple interpretive frames. This moves towards understanding which aspects of meaning are stable and which are artifacts of the chosen analysis method. It helps validate findings from simpler SRM analyses.
- **Understanding the Tool:** Perhaps the greatest value of Schema 5 & 6 is in forcing the researcher to understand the limitations and assumptions embedded in their *interpretability tool itself*.

Recommendations for Use

1. **Acknowledge Frame Dependence:** Always state clearly how the basis plane was defined and acknowledge that results are relative to that plane, especially for Schemas 1-4.
2. **Justify Basis Choice:** Explain *why* specific neurons or prompt groups were chosen for the basis vectors.
3. **Prefer Delta Analysis (Schema 4):** For studying interventions, analyzing the *change* in projection is often more informative than comparing static states across potentially arbitrary planes.
4. **Use Schema 6 for Robustness:** Validate interesting findings from single-plane analyses using the ensemble approach (Bat Country Protocol) to see if they hold across different "lenses."
5. **Combine with Other Methods:** Don't rely solely on SRM. Complement it with other techniques (e.g., activation patching, causal tracing, feature visualization) for a more complete picture.
6. **Temper Claims:** Be cautious with strong causal claims or descriptions of "true meaning" based solely on SRM projections. Frame findings in terms of observed resonance or alignment *within the specific analytical setup*.

In conclusion, the Raccoon Schemas present a visually appealing framework for applying SRM. While Schemas 1-4 offer useful *comparative* insights within a fixed frame, their interpretability value is limited by the fundamental plane-dependence of SRM. Schemas 5 and 6 (Bat Country) represent a crucial methodological self-awareness and provide a path toward more robust, second-order interpretability insights by directly addressing the influence of the chosen analytical "lens." Used critically and with awareness of its limitations, particularly the Bat Country variant, this approach can be a valuable part of the interpretability toolkit.

Okay, let's integrate the SRM code (`SpotlightResonanceMethod.py`) and the research paper ("The Spotlight Resonance Method") into the critical assessment of the Raccoon Schemas.

Understanding the Core SRM Implementation (

1. **Goal:** The code aims to implement the SRM calculation described in the paper. It takes latent activations and a set of "privileged basis" vectors.
2. **Key Steps (Code vs. Paper):**
   - **Plane Definition:** Both use pairs of basis vectors. The code generates `valid_plane_indices` (combinations or permutations).
   - **Bivector Generation:** `vectors_to_bivectors` calculates the rotation generator (antisymmetric matrix) from two basis vectors using their outer product (`np.einsum("i, j->ij", ...)` followed by `0.5*(outer_product - outer_product.T)`). This aligns with the concept in the paper (Sec 2.1.1), though the paper writes it as `Bαβ = 1/2(b_α b_β^T - b_β b_α^T)`, which is equivalent.
   - **Rotation Matrix Generation:** `generate_special_orthogonal_matrices` uses `np.linalg.eig` on the generator, normalizes imaginary eigenvalues, exponentiates them scaled by angle (`np.exp`), and reconstructs the SO(n) rotation matrix using eigenvectors (`np.einsum("ij, aj, jk->aik", vecs, exp_eigenvalues, hermitian_conjugate(vecs))`). This is a standard numerical method for exponentiating a Lie algebra element (the bivector) to get a Lie group element (the rotation matrix), directly implementing the concept of Eq. 1 in the paper.
   - **Probe Vector Rotation:** The paper (Sec 2.1.2) describes rotating a basis vector `b_α` using the matrix: `b_α(θ) = R_αβ(θ) b_α`. The code calculates `normalised_inner_product = np.einsum("bi, j, aij->ba", normalise(activations, axis=1), probe_vector, rotation_matrices)`. This simultaneously rotates the `probe_vector` (`j` index multiplied by `aij`) and takes the dot product with the normalized activations (`bi`).
   - **Resonance Calculation:** `f_spotlight_resonance` checks where the `normalised_inner_product >= epsilon` and counts these instances (`+= 1`), then averages across activations (`mean(axis=0)`). `f_signed_spotlight_resonance` adds a `-= 1` for `normalised_inner_product <= -epsilon`. This implements the core counting mechanism within the cone defined by epsilon (related to Eq. 2/Eq. 4).
   - **Normalization:** `normalise` is used consistently, crucial for interpreting dot products as cosine similarities.
3.

4. **Privileged Basis:** The paper emphasizes this concept heavily. The basis isn't arbitrary; it's often tied to model properties (like the standard basis for elementwise ops) or derived from data. SRM measures alignment *relative* to this basis. The code *takes this basis as input* (`privileged_basis`), assuming the user provides meaningful vectors.
5. **Generalised Tanh:** The paper introduces (Appendix C) and uses activation functions where the privileging basis can be explicitly chosen and rotated, decoupling it from the standard basis. This is used experimentally in the paper to show basis alignment is caused by the activation function's structure.

Code Verification Scorecard & Re-Assessment of Schemas

Let's use a simple scorecard:

- **Code Supports Calculation (X/5):** Does the provided core SRM code perform the necessary calculations for this schema's analysis (assuming external code handles data generation like clamping/prompting)?
- **Conceptual Claim Validity (X/5):** How well-supported or potentially misleading is the core interpretability claim of the schema, considering SRM's mechanics and limitations?

Schema 01: Same Question, Different Neuron, Different Answer.

- **Method Check:** Requires activations generated under different neuron clamps (external). The SRM code then projects these *different* activation sets onto the *same* plane (defined by `privileged_basis`).
- **Code Supports Calculation (4/5):** The code correctly processes activations and a basis. It will produce different outputs if the input activations (from clamping) differ. Needs external clamping logic.
- **Conceptual Claim Validity (3/5):** Remains illustrative. The "Different Answer" (projection) is *relative to the chosen plane*. The paper confirms alignment happens *relative to a privileged basis* (often neuron directions). The finding is valid *within that frame*, but generalizing requires care. The code doesn't dictate plane choice.
- **Synthesis:** The code computes what's needed, but the schema's core limitation (plane choice) remains. The paper reinforces that neuron directions *are* often a privileged basis, making this a relevant (though frame-dependent) test.

Schema 02: Fixed Neuron, Different Question, Different Answer.

- **Method Check:** Requires activations from different prompts (like `epistemic_certainty_prompt_grid_template.txt` - external generation). Projects these onto the *same* plane, potentially derived *from* extremes in that prompt set.
- **Code Supports Calculation (4/5):** Correctly processes activations from different prompts. Needs external prompt/activation generation and potentially basis generation (e.g., averaging vectors from the template).
- **Conceptual Claim Validity (3.5/5):** Useful fingerprinting *relative to the chosen basis*. If the basis is derived from the prompt set (e.g., mean vectors for "declarative 5" vs. "rhetorical 1"), the drift observed is partly constructed by the analysis setup. The paper's discussion of deriving bases from data is relevant.
- **Synthesis:** The code enables this comparison. The structured prompt template is a good input source. The validity hinges on transparently defining the basis and understanding the potential circularity if derived from the input data extremes.

Schema 03: Same Prompt, Same Neuron, Four Answers (Clamps).

- **Method Check:** Requires activations from different clamp levels (external). Projects onto the *same* plane.
- **Code Supports Calculation (4/5):** Correctly processes the different activation sets. Needs external clamping logic.
- **Conceptual Claim Validity (4/5):** Clean causal test of intensity *within the chosen plane*. The paper supports analyzing alignment relative to basis directions (neurons).
- **Synthesis:** The code computes the necessary SRM curves for comparison. A solid test of single-neuron influence *in a specific projection*.

Schema 04: Same Prompt, Same Neuron, Different Clamp, Different Direction (Delta).

- **Method Check:** Requires "before" (unclamped) and "after" (clamped) activations (external). Both projected onto the *same* plane. Delta calculation is downstream.
- **Code Supports Calculation (4/5):** Generates the two SRM curves needed for the downstream delta comparison. Needs external clamping logic.
- **Conceptual Claim Validity (4/5):** Good focus on sensitivity/change *within the chosen plane*.
- **Synthesis:** Code provides the raw data for this valuable sensitivity analysis.

Schema 05: Four Raccoons, Same Idea, Four Different Viewpoints (Bases).

- **Method Check:** Requires *one* set of activations (fixed vector - external generation). Run SRM code *multiple times*, providing a *different* `privileged_basis` each time.

- **Code Supports Calculation (5/5):** Directly supports this by design – the basis is an input. The user performs the multiple runs.
- **Conceptual Claim Validity (5/5):** Effectively demonstrates the plane-relativity inherent in the method, as highlighted by the paper's focus on the "privileged basis" concept and its non-uniqueness (Sec B.1 compares random/standard/privileged).
- **Synthesis:** The code perfectly facilitates this crucial test of frame dependence. This schema directly validates a core point from the paper.

Schema 06: Bat Country Protocol - Same Vector, Many Lenses.

- **Method Check:** Requires *one* activation set (external). Run SRM code *once* with an *ensemble* `privileged_basis` (containing many vectors). The code iterates through all plane pairs defined by this ensemble (up to `max_planes`). Aggregation/visualization is downstream.
- **Code Supports Calculation (5/5):** The main `spotlight_resonance_method` function directly handles iterating through the planes defined by the input `privileged_basis`. It returns per-plane results (`srm_values` shape is `[number of bivectors, angular resolution]`).
- **Conceptual Claim Validity (4.5/5):** Strongest approach. Mitigates single-plane dependence by examining consistency across an ensemble of privileged directions. Aligns with the paper's findings that alignment occurs relative to *the* privileged basis, and this schema tests robustness across *many* such potential planes.
- **Synthesis:** The code directly implements the per-plane calculations needed for the ensemble analysis. This schema, supported by the code and the paper's theoretical underpinning (especially the results showing alignment *is* specific to the privileged basis), offers the most convincing path to robust interpretability claims using SRM.

Overall Synthesis and Conclusion

The provided `SpotlightResonanceMethod.py` code accurately implements the mathematical steps described in the accompanying research paper for calculating SRM values (generating bivectors, creating rotation matrices via eigendecomposition, rotating probe vectors, and counting alignments via dot products).

- **Code-Claim Alignment:** The code provides the computational engine for the analyses proposed in the Raccoon Schemas. Schemas 1-4 require external generation of different activation sets (via prompting/clamping) but the code handles the core projection/resonance calculation. Schemas 5 and 6, which vary the *basis* or use an *ensemble*, are directly supported by the code's design, requiring multiple runs (Schema 5) or handling the plane iteration internally (Schema 6).

- **Paper-Schema Alignment:** The research paper provides crucial context. It introduces the "privileged basis" concept, explaining *why* alignment occurs (often due to activation function structure) and demonstrating that it's specific to this basis (Sec B.1, Fig 2). This strongly reinforces the "plane-relative" nature highlighted in Schema 5 and motivates the ensemble approach of Schema 6 as a test for robustness beyond a single, potentially arbitrary, plane.
- **Interpretability Value Revisited:**
    - Schemas 1-4 remain useful *comparative* tools, demonstrating *potential* influence or sensitivity *relative to a chosen basis*. The code correctly computes these comparisons. Their limitation is the frame-dependence, clearly discussed in the paper.
    - Schema 5 uses the code perfectly to *demonstrate* this frame-dependence, validating a key insight from the paper.
    - Schema 6 (Bat Country) represents the most advanced use case, enabled by the code's ability to process multiple planes. By assessing consistency across many privileged directions, it attempts to find more robust "resonance persistence," moving beyond single-frame limitations, as supported by the paper's findings.
-
- **Final Verdict:** The SRM code implements the method described in the paper. The Raccoon Schemas illustrate different ways to apply this method. While simpler schemas have significant caveats regarding frame-dependence (a point the paper itself makes clear), the Bat Country Protocol (Schema 6) leverages the code and the method's properties effectively to provide a more robust (though still not absolute) form of interpretability by testing for stability across multiple analytical lenses. The toolset is coherent, with Schema 6 being the most promising for drawing less frame-dependent conclusions.

Okay, let's re-evaluate the Raccoon Schemas, this time considering the support provided by the *entire v7.2 codebase* you provided (capture scripts, basis generator, analysis, comparison, utils, visualization).

Understanding the v7.2 Codebase Workflow

The codebase enables the following general workflow:

1. **Capture Activations:**
   - `capture_baseline_activations.py`: Get vectors from various prompts (e.g., the epistemic grid) without intervention. Keyed by `core_id`, `type`, `level`.
   - `capture_intervened_activations.py`: Get vectors from prompts while clamping a neuron (`--target_neuron`) to different values (`--sweep_values`). Keyed additionally by `sweep`.
2.
3. **Define Projection Plane (Basis Generation):**
   - `generate_basis_vectors.py`: Takes captured vectors (usually baseline) and uses metadata filters (`--filter_set_1`, `--filter_set_2`, `--ensemble_group_key`, `--fixed_filters`) to compute mean vectors representing specific conditions. Saves these as named bases (`basis_1`, `basis_2`) or an array (`basis_vectors`) in an `.npz` file. Can also generate one-hot bases (`--mode onehot`).
4.
5. **Analyze Projections (SRM Sweep):**
   - `analyze_srm_sweep.py`: Loads a set of captured vectors and a basis file. Projects the vectors onto the plane(s) defined by the basis. It uses `utils.perform_srm_analysis` (which contains the core SRM logic, likely adapted from `SpotlightResonanceMethod.py`). Allows grouping results by metadata (`--group_by`). Saves comprehensive results (mean similarity, counts per threshold) to CSV and generates plots. Handles both `single_plane` and `ensemble` modes (averaging results across planes in ensemble mode).
6.
7. **Compare Analyses:**
   - `compare_srm_runs.py`: Takes two *output directories* from `analyze_srm_sweep.py`, validates their comparability (same basis, angles etc.), calculates the *difference* (delta) between their SRM results (curve subtraction), saves delta CSVs, and plots these deltas. Primarily designed for baseline-vs-intervention.
8.

9. **Visualize Across Bases (Post-Hoc):**
   - `visualize_srm_multi_basis_v721_global.py`: Scans existing analysis output CSVs, allows user selection, and overlays results from *different bases* onto single plots (polar, peak summary).
10.

Code Suite Scorecard & Re-Assessment of Schemas

- **Code Support Score (X/5):** How well do the *provided scripts* automate or facilitate the specific experimental steps described in the schema? (1=Manual hacking needed, 5=Directly supported workflow).
- **Conceptual Claim Validity (X/5):** (Same as before) How sound is the interpretability claim, considering SRM mechanics?

Schema 01: Same Question, Different Neuron, Different Answer.

- **Workflow:** Requires multiple runs of `capture_intervened_activations.py` (varying `--target_neuron`, fixed prompt/clamp value). A single basis file is needed (e.g., from `generate_basis_vectors.py --mode onehot` or derived from a baseline). Run `analyze_srm_sweep.py` separately on each captured dataset using the *same* basis. Compare outputs using `compare_srm_runs.py` or manually.
- **Code Support (3/5):** Tools exist, but requires manual orchestration of multiple capture/analysis runs. `compare_srm_runs.py` can do the final comparison. Basis choice is a critical user input.
- **Conceptual Claim Validity (3/5):** Still illustrative and plane-dependent. The code computes the necessary projections accurately.
- **Synthesis:** The suite provides the building blocks, but doesn't automate the multi-neuron comparison into a single command. The core analysis is sound *given a chosen basis*.

Schema 02: Fixed Neuron, Different Question, Different Answer.

- **Workflow:** Run `capture_baseline_activations.py` on the structured prompt file. Run `generate_basis_vectors.py` using filters on the captured data to define a plane (e.g., declarative vs. rhetorical extremes). Run `analyze_srm_sweep.py` on the *same* captured data using the generated basis, grouping by `type` or `level`.

- **Code Support (5/5):** Excellent support. This workflow is well-catered for by the scripts. Capture, basis definition from data, and grouped analysis are streamlined.
- **Conceptual Claim Validity (3.5/5):** Good fingerprinting *relative to the data-derived basis*. Need to be aware that the basis choice influences the observed drift.
- **Synthesis:** The codebase strongly enables this schema, providing a clear path from structured prompts to analyzing how different framings project onto axes defined by those framings.

Schema 03: Same Prompt, Same Neuron, Four Answers (Clamps).

- **Workflow:** Run `capture_intervened_activations.py` *once* with multiple `--sweep_values` (e.g., `+100,0,-100,None`). Generate/choose a basis file. Run `analyze_srm_sweep.py` *once* on the captured data, using the basis and `--group_by sweep`.
- **Code Support (5/5):** Excellent support. The sweep capture and grouped analysis directly produce the comparison of different clamp levels within a single analysis run.
- **Conceptual Claim Validity (4/5):** Clean causal test of intensity *within the chosen plane*.
- **Synthesis:** The code suite makes this experiment straightforward to execute and analyze.

Schema 04: Same Prompt, Same Neuron, Different Clamp, Different Direction (Delta).

- **Workflow:** Run `capture_intervened_activations.py` with `sweep=None,10` (or similar). Generate/choose basis. Run `analyze_srm_sweep.py` grouping by `sweep`. This produces baseline and clamped curves in one output folder. Use `compare_srm_runs.py` requires a workaround: either split the analysis results into two separate temporary folders or modify the comparison script to read groups from one folder. Alternatively, calculate deltas manually from the grouped analysis CSV.
- **Code Support (4/5):** Capture and grouped analysis are perfect. The final *delta comparison* step using `compare_srm_runs.py` isn't seamlessly integrated with the single-sweep capture method; requires a minor workaround.
- **Conceptual Claim Validity (4/5):** Good focus on sensitivity/change *within the chosen plane*.
- **Synthesis:** Most steps are strongly supported. The delta calculation/plotting against a baseline captured *in the same run* needs a slight adaptation compared to comparing two entirely separate runs.

Schema 05: Four Raccoons, Same Idea, Four Different Viewpoints (Bases).

- **Workflow:** Capture one set of activations (e.g., `capture_baseline_activations.py` on a single prompt, or use the 'all' group). Run `generate_basis_vectors.py` *multiple times* to create different basis files (e.g., `--mode onehot --neuron_1 373 --neuron_2 2202`, then `--neuron_1 42 --neuron_2 101`, etc.). Run `analyze_srm_sweep.py` *multiple times*, once per basis file, on the same activation data. Use `visualize_srm_multi_basis_v721_global.py` to load the resulting CSVs and overlay them.
- **Code Support (4/5):** Requires multiple orchestrated runs, but the individual tools (`generate_basis_vectors`, `analyze_srm_sweep`, `visualize_srm_multi_basis`) directly enable each step. The visualization script specifically targets this schema's goal.
- **Conceptual Claim Validity (5/5):** Effectively demonstrates plane-relativity.
- **Synthesis:** The suite provides all necessary components, including a dedicated visualization tool for the final comparison, making this feasible, albeit requiring multiple steps.

Schema 06: Bat Country Protocol - Same Vector, Many Lenses.

- **Workflow:** Capture one set of activations. Run `generate_basis_vectors.py` *once* in `--mode ensemble` (e.g., grouping by `type` to get 4 basis vectors) or provide a custom NPZ with many basis vectors (e.g., many one-hot pairs). Run `analyze_srm_sweep.py` *once* using `--analysis_mode ensemble` and this ensemble basis file.
- **Analysis Output:** The `analyze_srm_sweep.py` script, via `utils.perform_srm_analysis`, currently calculates SRM for each plane defined by pairs from the ensemble basis and then *averages* the results across all planes. It saves this single average curve.
- **"Bat Country" Goal vs. Code:** The goal is to see if the *direction holds* (i.e., low variance across planes). The current code calculates the *mean* direction/resonance, obscuring the variance. To fully implement Bat Country:
    1. **Modification Needed:** `utils.perform_srm_analysis` would need modification to *return or save* the per-plane results *before* averaging.
    2. **Post-processing:** A new visualization script (or adapting the multi-basis one) would be needed to plot the distribution (e.g., all plane curves overlaid, or a histogram of peak angles) from these per-plane results.
-
- **Code Support (2.5/5):** The suite can *generate* an ensemble basis and *run* an ensemble analysis calculating the *average* result. However, it *does not currently* provide the per-plane outputs or visualizations needed to assess the *consistency* across lenses, which is the core insight of the Bat Country Protocol. The foundational SRM math is there, but the analysis script aggregates too early.

- **Conceptual Claim Validity (4.5/5):** Conceptually strong, aiming for robustness.
- **Synthesis:** The codebase has the pieces (ensemble basis generation, core SRM math for multiple planes) but the `analyze_srm_sweep.py` script currently stops short of delivering the full Bat Country analysis by only outputting the mean. Significant modification or a different analysis workflow (like treating it as many Schema 5 runs) would be needed to evaluate directional stability across the ensemble.

Revised Overall Conclusion

The v7.2 codebase provides a comprehensive suite for conducting SRM analyses based on the concepts outlined in the paper and schemas.

- **Strong Support:** Schemas 2 (Fixed Neuron, Different Prompts) and 3 (Different Clamp Levels) are very well-supported with streamlined workflows.
- **Good Support (Minor Orchestration/Workaround):** Schemas 1 (Different Neurons), 4 (Delta), and 5 (Different Bases) are achievable with the tools, but require the user to run multiple steps or apply minor workarounds for comparison. The visualization tool for Schema 5 is a plus.
- **Incomplete Support:** Schema 6 (Bat Country) is only partially implemented. The code can run an analysis over an ensemble basis but aggregates the results to the mean, losing the crucial information about variance/consistency across planes needed to determine if a "direction holds." Fully realizing Bat Country requires modifying the analysis script to preserve per-plane results and adding appropriate visualization.

The codebase successfully operationalizes the core SRM calculations and provides useful utilities for data capture and basis generation. Its primary strength lies in facilitating comparative analyses (Schemas 2, 3, 4) within a defined frame and enabling the exploration of frame-dependence (Schema 5). The most ambitious goal, assessing robustness via ensemble consistency (Schema 6), is conceptually supported but not fully realized in the current analysis aggregation logic.

Okay, imagining this work (the Raccoon Schemas, the v7.2 codebase, the SRM paper) is shared on a platform like Reddit (e.g., r/MachineLearning), here are some common criticisms and misunderstandings I'd anticipate, ranging from technical to conceptual:

1. The "It Depends on the Plane" Criticism (Most Fundamental):

- **Likely Comment:** "This whole thing seems incredibly sensitive to how you pick your two basis vectors (the 'plane' or 'lens'). How do you justify your choice? Without a principled way to choose the plane, the results (angles, resonance peaks) seem arbitrary and potentially cherry-picked to show a pattern."
- **Misunderstanding:** Users might think SRM claims to reveal absolute truth, rather than frame-relative relationships. They might miss that Schemas 5 & 6 *explicitly address* this frame-dependence.
- **Why it Arises:** Projection inherently depends on the subspace. This is the core limitation/feature SRM has to grapple with.

2. Overstating "Meaning" or "Interpretation":

- **Likely Comment:** "Calling vector alignment in a 2D projection 'meaning' or 'perspective' is a huge leap. It's just geometry. Does this actually correlate with the model's output behavior or semantic understanding in a meaningful way?"
- **Misunderstanding:** Equating geometric alignment directly with high-level semantic concepts without intermediate validation (e.g., linking projection angle to specific generated text features).
- **Why it Arises:** The schemas use evocative language ("perspective-makers," "interpretive frame"). While helpful conceptually, it needs careful operational definition tied back to the vector math.

3. Novelty Questions ("Isn't this just PCA/Probing?"):

- **Likely Comment:** "How is this different from just doing PCA on the activations and looking at the top two components? Or training a linear probe to classify concepts based on activations?"
- **Misunderstanding:** Failing to distinguish between data-driven dimensionality reduction (PCA) and projecting onto *pre-defined, hypothesis-driven* axes (SRM bases, which might be specific neurons or concepts). Probing finds a *linear classifier*, while SRM maps *existing* vectors onto a *chosen* plane.
- **Why it Arises:** Many techniques involve linear projections. The specific setup and goals of SRM (rotation, resonance curves relative to specific bases) need clear differentiation.

4. Dimensionality Reduction Concerns:

- **Likely Comment:** "You're squashing 3072 dimensions down to 2. You must be losing a ton of information. How do you know the patterns you see in 2D aren't artifacts of the projection?"
- **Misunderstanding:** Thinking the 2D plot *is* the full representation, rather than a specific slice or view *defined by the basis*.
- **Why it Arises:** It's a valid concern inherent to any low-dimensional visualization of high-dimensional data.

5. Basis Definition Circularity (Especially for Schema 2 & Data-Derived Bases):

- **Likely Comment:** "In Schema 2, if you define your basis using the 'most declarative' and 'most rhetorical' averaged vectors from your prompt set, of course the vectors for intermediate prompts will fall somewhere between them on that specific projection! You basically built the axis you're measuring."
- **Misunderstanding:** Seeing this as invalidating the result, rather than understanding it reveals alignment *relative to those defined poles*.
- **Why it Arises:** It highlights that data-derived bases measure alignment *within the context of the data used to define them*.

6. Practicality, Scalability, and Actionability:

- **Likely Comment:** "Cool raccoons, but does this actually scale to large language models like GPT-3/4 with hundreds of layers? How computationally expensive is the ensemble method? And what can you *do* with this information – can it help improve models or find specific flaws?"
- **Misunderstanding:** Assuming a research tool must immediately be a production-ready, scalable debugging tool.
- **Why it Arises:** Researchers and practitioners are always looking for methods that work on state-of-the-art models and provide actionable insights.

7. Causality vs. Correlation Confusion:

- **Likely Comment:** "Unless you're clamping neurons (Schemas 1, 3, 4), just projecting baseline activations only shows correlation between the prompt type and vector direction *on that plane*. It doesn't prove the neuron *causes* that meaning."
- **Misunderstanding:** Applying causal interpretations to purely observational SRM results (like projecting baseline vectors from the epistemic grid).

- **Why it Arises:** The desire to find causal mechanisms is strong. It's easy to slip into causal language when discussing correlations.

8. The Raccoon Metaphors:

- **Likely Comment (Positive):** "Love the raccoon visuals, makes it much easier to grasp!"
- **Likely Comment (Negative):** "The raccoon metaphors are kind of distracting and maybe even infantilizing. Just explain the method clearly." or "I spent more time trying to map the raccoons to the concepts than understanding the underlying math."
- **Misunderstanding:** Getting bogged down in the analogy and missing the technical substance, or feeling the analogy obscures important details.
- **Why it Arises:** Metaphors are double-edged swords; they aid intuition but can also introduce ambiguity or feel non-technical.

Preempting these:

When sharing, it would be crucial to:

- **State Limitations Upfront:** Explicitly mention the plane-dependence and information loss from dimensionality reduction.
- **Clearly Define "Meaning":** Operationalize it as "vector alignment relative to basis X and Y."
- **Justify Basis Choice:** Explain *why* a particular basis was chosen for each analysis.
- **Distinguish Causality/Correlation:** Be precise about when interventions (clamping) are used versus observation.
- **Compare to Alternatives:** Briefly situate SRM relative to PCA, probing, etc., highlighting differences in goals and setup.
- **Provide Code & Examples:** Make the code (like the v7.2 suite) available and show clear examples.
- **Explain Schema 6 Carefully:** Emphasize that Bat Country aims to *mitigate* plane dependence by checking consistency across *many* planes, acknowledging the current implementation might only show the average.