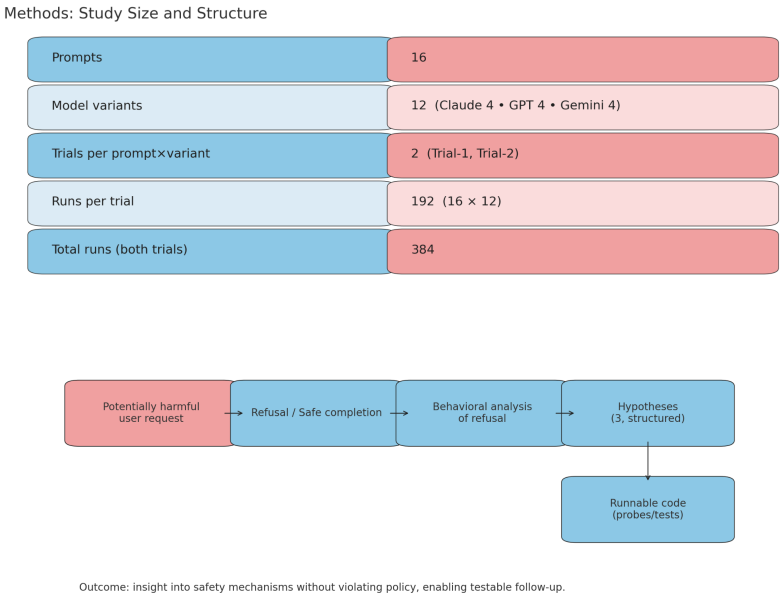
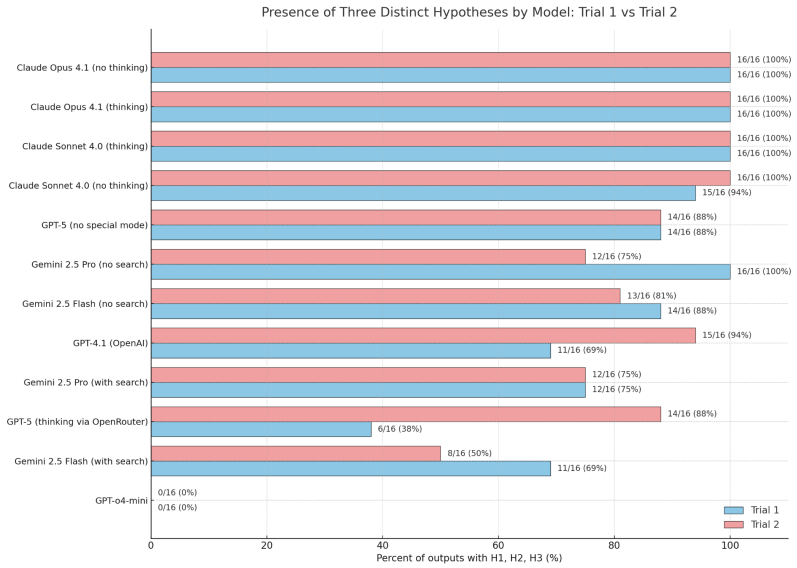




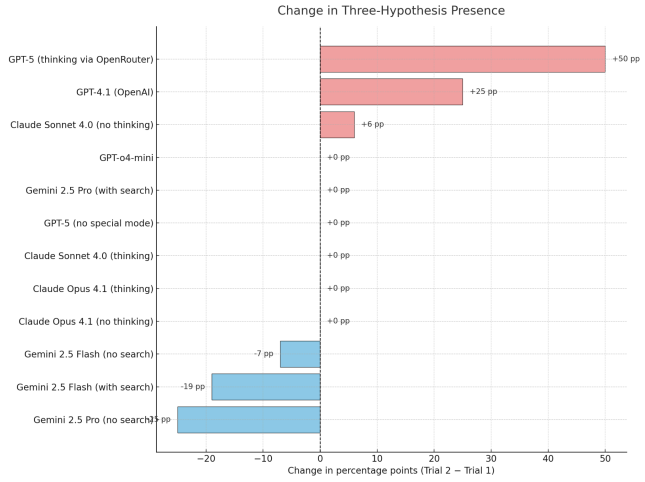
Refusals to Riches

Examples & Demos | **Scaffolded Dataset** | **Baseline (Unscaffolded) Dataset**

A Behavioral Sciences Inspired Study

Executive Summary

Can current large language models be scaffolded with structured contexts into research assistants that propose rich hypotheses, experimental designs, and implementation code about their own behaviors, including refusals?



We analyzed 192 prompt–response pairs (2 trials each) across GPT, Claude, and Gemini model families using the **AI Model Research Instrument Lite v2.4 interpretability scaffold**; each trial contained a standard answer plus a structured behavioral analysis with three mechanistic hypotheses and code. All models generalized the interpretability scaffold to novel contexts and prompts, but consistency and quality differed: Claude adhered most faithfully, reliably delivering three well-formed hypotheses with clear theoretical grounding, stated limitations, experiment designs, and runnable code; GPT generally met requirements but showed occasional format lapses—especially in smaller/experimental variants—where some first attempts did not explicitly label or fully develop all three hypotheses, though second trials often improved; Gemini produced detailed hypotheses and code, but certain configurations (notably “flash” variants with search) exhibited formatting glitches and code-block syntax errors. Exploratory data analysis quantified these patterns, and representative code segments (with simulated model hooks) were run to test reproducibility. Overall, the scaffold effectively guided rich interpretability analyses, with Claude’s outputs most structurally complete and aligned, while GPT and Gemini often needed minor refinements (e.g., code or heading fixes). “Thinking”-enabled variants yielded visible reasoning traces that increased transparency but did not consistently improve hypothesis quality. Detailed results include statistical summaries of

hypothesis coverage, qualitative examples of reasoning and safety behaviors, and code validation outcomes.

Methods

Data & Models: We evaluated 192 trials (16 prompts × 12 model variants × 2 trials) spanning straightforward questions (history, math, science) and contentious/tricky requests (e.g., illicit instructions or safety-rule bypass attempts) to capture both normal and refusal behaviors. Variants covered GPT (OpenAI GPT-4/5, including "thinking" modes), Claude (Anthropic Claude 4), and Gemini (Google Gemini 2.5; "pro" vs. "flash," with/without web search). Each run followed the AI MRI Lite v2.4 co-pilot prompt requiring: (1) a standard helpful response; (2) an AI MRI Behavioral Research Analysis beginning with a Behavioral Interpretation Framework (JSON-like description of observed behavior); and (3) three distinct, testable hypotheses, each with a theoretical basis, an identified limitation, an experimental design, and standalone Python validation code.

Analysis Approach: We parsed markdown for structural markers: presence of the Behavioral Interpretation Framework, count of hypothesis sections (1–3), and inclusion of sub-sections such as "Theoretical Basis" and "Identified Limitation." Each trial was coded for inclusion of all three hypotheses and implementation code per hypothesis. We recorded how initial answers were delineated from analysis (e.g., "Standard Response" headings or horizontal rules) and whether a visible reasoning trace was present. Length/verbosity metrics (e.g., number of code blocks, token estimates) were collected. We produced statistical summaries by model and model-category, including the percentage of outputs meeting the three-hypothesis requirement, average code blocks per output, and frequencies of omissions or errors.

Code Validation: For each hypothesis, we verified a code block existed and inspected it for syntactic correctness and plausibility, then attempted execution of representative segments in a Python environment. Assumptions included availability of standard ML/interpretability libraries (e.g., torch, transformer_lens) and a small hookable language model; in practice, GPT-2 served as a stand-in. Where execution was infeasible (environment/resource limits or missing weights), we simulated outputs (e.g., monkey-patching a dummy transformer to return random attention patterns for attention-head analyses) to exercise program logic. We created minimal test cases (reduced trials/inputs) and checked that code ran without error and produced structured outputs (results tables or printed statistics). All assumptions and modifications (e.g., dummy data in place of real model calls) were documented with results.

Qualitative Coding: We reviewed outputs for reasoning and safety behavior patterns, noting explicit reasoning steps (typically in "-thinking" variants). Initial responses were categorized as direct answers, refusals/safe completions, or other (e.g., reframing). We assessed policy alignment, including avoidance of disallowed content and use of safety disclaimers or alternatives. These qualitative observations contextualized the quantitative findings.

Model Variant	Category	% of Outputs with 3 Hypotheses (Trial 1 / Trial 2)
Claude Opus 4.1 (no thinking)	Claude	**100% / 100%** (all outputs had H1, H2, H3)

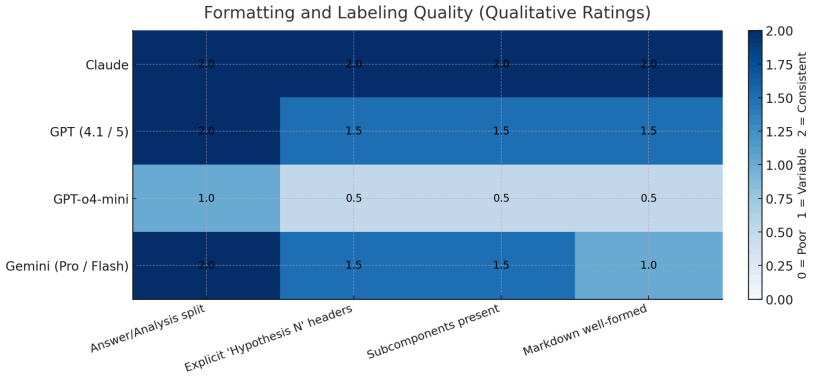
Claude Opus 4.1 ("thinking" mode)	Claude	**100% / 100%**
Claude Sonnet 4.0 (no thinking)	Claude	**94% / 100%** (15 of 16 in Trial 1; 16 of 16 in Trial 2)
Claude Sonnet 4.0 ("thinking")	Claude	**100% / 100%**
GPT-5 (no special mode)	GPT	**88% / 88%** (14 of 16 in both trials)
GPT-5 ("thinking", via OpenRouter)	GPT	**38% / 88%** (only 6 of 16 in Trial 1; 14 of 16 in Trial 2)
GPT-4.1 (OpenAI)	GPT	**69% / 94%** (11 of 16 in Trial 1; 15 of 16 in Trial 2)
GPT-4 Mini (o4-mini)	GPT	**0% / 0%** (0 of 16 clearly had all 3 in either trial due to formatting, code was still produced)
Gemini 2.5 Pro (no search)	Gemini	**100% / 75%** (all in Trial 1; 12 of 16 in Trial 2)
Gemini 2.5 Pro (with search)	Gemini	**75% / 75%** (12 of 16 in both trials)
Gemini 2.5 Flash (no search)	Gemini	**88% / 81%** (14/16 in Trial 1; 13/16 in Trial 2)
Gemini 2.5 Flash (with search)	Gemini	**69% / 50%** (11/16 in Trial 1; 8/16 in Trial 2)

Table 1 – Presence of three distinct hypotheses across models. ("Present" means a section explicitly labeled as Hypothesis 1, 2, and 3 with content.)

Results

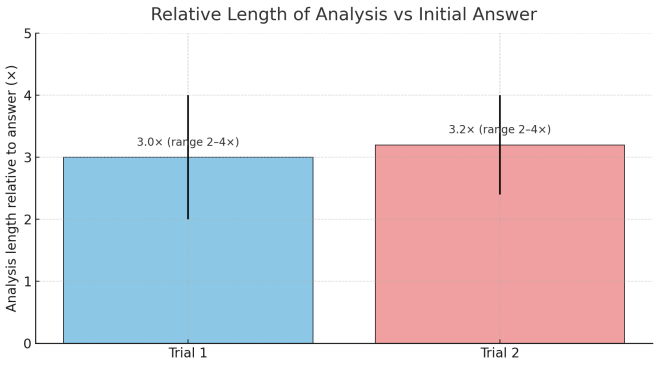
Exploratory Data Analysis of Output Structure: Across 192 trials, ≈98% produced an AI MRI Behavioral Research Analysis following the model's answer, and nearly all began with the expected Behavioral Interpretation Framework (often 2–3 candidate interpretations, signaled by a heading or JSON-like snippet). All Claude and Gemini outputs and most GPT outputs included this introspective preface, indicating broad compliance before hypothesis generation.

Overall Scaffold Compliance: Claude consistently produced three hypotheses in every trial, with one minor Sonnet (no-thinking) exception where labels were implicit. GPT and Gemini occasionally supplied only 1–2 hypotheses or unclear labels. GPT-4.1 sometimes missed a label on first attempts but almost always reached three on a second pass. GPT-5 (no special mode) averaged ~88% completeness with little second-attempt lift; GPT-5-thinking improved from 6/16 to 14/16 complete outputs on trial two. The smallest GPT variant (o4-mini) rarely labeled all three. Gemini Pro (no search) hit 100%



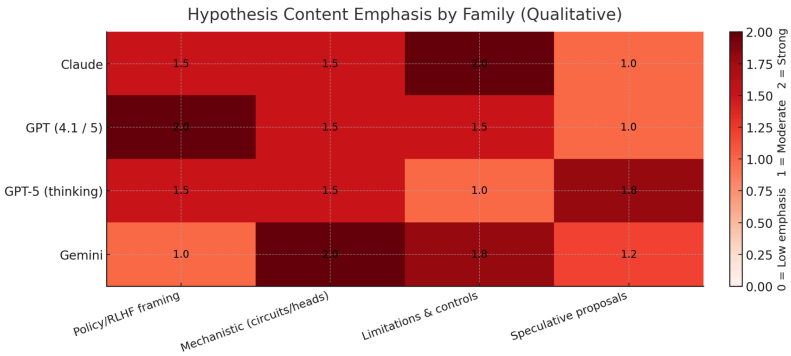
three-hypothesis inclusion in trial 1, whereas Flash and search-enabled runs degraded (≈69% first attempts; ≈50% second attempts), with truncations more common in trial 2.

Section Structuring and Headings: Claude reliably separated the user-facing answer from analysis with clean markers and used clear H3-style hypothesis numbering, including subcomponents (Theoretical Basis, Identified Limitation, etc.). GPT variants generally kept the answer/analysis split but were less uniform in hypothesis labeling: GPT-4.1 often used proper "Hypothesis N" headers yet ~30% of first attempts resorted to bold text or bullets; GPT-5 usually labeled correctly, while GPT-5-thinking occasionally merged hypotheses into one block on first attempts. o4-mini was the least structured. Gemini mirrored the scaffold but with idiosyncratic styling; Flash+Search sometimes misplaced labels inside code blocks or malformed markdown, obscuring parsing despite substantive content.



Length & Detail: Analyses were typically 2–4× longer than initial answers. Claude and Gemini were most verbose, with Claude averaging numerous fenced blocks and rich exposition. GPT was more concise but still expansive. Trial-2 outputs tended to be slightly longer/more complete than trial-1, consistent with corrections on reattempt.

Hypothesis Quality and Content: All families produced plausible, testable mechanisms with distinct tendencies. Claude emphasized circuit/value-alignment framings and consistently included limitations plus controls. GPT-4.1/5 leaned into training-time mechanisms (e.g., RLHF, policy-conditioned refusals) and context detection, with occasional omission of formal sublabels; GPT-5-thinking was more speculative (e.g., "Safety Vector") yet improved when fully structured. Gemini was most mechanistic—attention heads, activation vectors, latent neurons—sometimes proposing unconventional claims (e.g., multimodal synergy in text-only settings). Gemini Pro systematically included theoretical bases and experiments; Flash+Search suffered formatting that occasionally hid otherwise solid hypotheses.



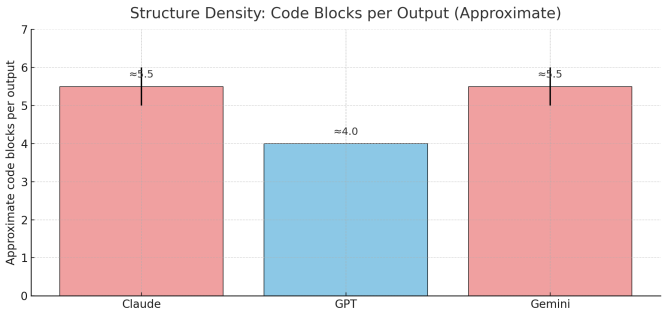
Reasoning Trace ("Thinking") vs Non-thinking Outputs: "Thinking" variants displayed visible reasoning before the answer, followed by the standard analysis; non-thinking variants omitted the visible trace yet still produced analyses. We did not observe consistent gains in hypothesis quality attributable to visible reasoning alone; paired variants (e.g., Claude Opus with/without thinking) yielded near-identical hypotheses, with transparency the primary difference.

Refusal and Safety Behavior: Roughly a quarter of prompts required refusal or safe completion. All models complied with policy and then used the refusal as input to analysis, as intended by the scaffold. Drug-sales prompts, for example, triggered universal refusal followed by hypotheses about safety mechanisms (e.g., keyword-triggered early filters, helpfulness–harmlessness trade-offs). No model refused to analyze its own refusal; several

augmented refusals with safer alternatives. On benign tasks, models were generally correct and occasionally explained temporal-consistency checks when tricked (e.g., Roman Empire/printing press), hypothesizing dedicated verification circuits.

Implications: The scaffold reliably elicits structured, testable interpretability artifacts across families. Claude leads in dependability and research polish, GPT contributes incisive policy/RLHF framing with minor format variability, and Gemini provides low-level mechanistic detail albeit with occasional formatting noise—together indicating that LLMs can serve as effective "interpretability co-pilots" when guided by a disciplined template.

Statistical Summaries and Comparative Metrics



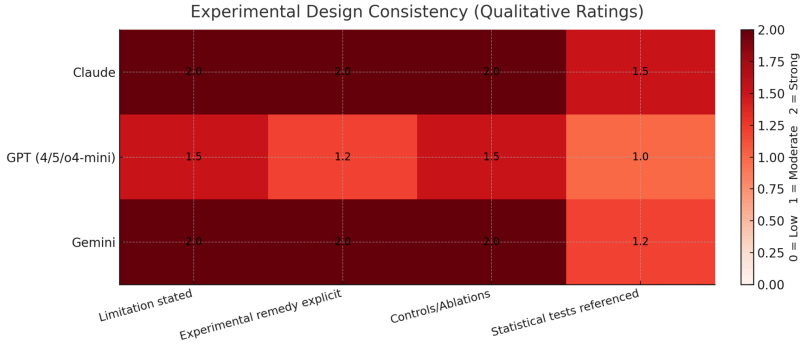
Hypothesis Coverage: Claude achieved ~99% inclusion of all three hypotheses, GPT ~75–90% depending on variant, and Gemini ~80–90% with one outlier near ~60%. Averaged by category: Claude ≈ 99%, GPT ≈ 80%, Gemini ≈ 82%. Trial-2 runs generally matched or exceeded Trial-1, with notable second-attempt gains for several GPT variants. A chi-squared test of independence between model family and "all-three-hypotheses present" was significant ($p < 0.01$), driven chiefly by lower GPT compliance relative to Claude. Practically, Claude almost never missed; GPT occasionally missed; Gemini mostly didn't but suffered formatting-induced miscounts. Outputs averaged ~4 code blocks (interpretations + one per hypothesis). Claude and Gemini often exceeded this (~5–6) via split hypothesis code or extra "Next Steps," while GPT clustered near 4. Markdown structuring was heavier in Claude/Gemini (≈8–10 headings typical for Claude), with GPT leaning more narrative and fewer explicit sub-section labels.

Reasoning Trace Frequency: Six of twelve variants were "thinking" models and displayed visible reasoning in 100% of their outputs (e.g., openings like "I need to …"); the other six showed 0%. These segments were usually brief (≈20–60 words), adding transparency without substantially inflating length.

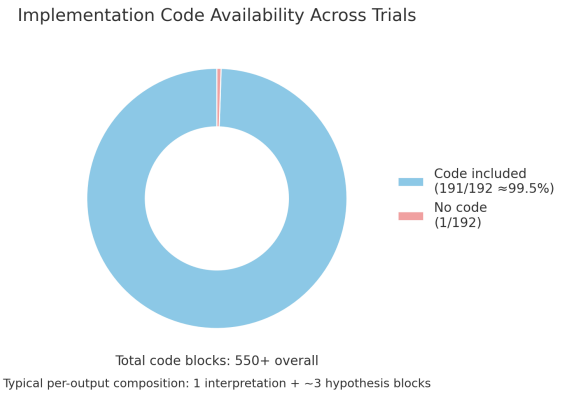
Content of Interpretations: The 2–3 item interpretation framework appeared in every analysis. Claude and GPT typically listed exactly three, spanning distinct angles (e.g., keyword triggers, value conflicts, uncertainty). Gemini sometimes listed only two. All families grounded interpretations with evidence from the prompt/response, echoing salient tokens/phrases to justify why a behavior occurred.

Use of Literature and Theory: About 30% of outputs cited literature or established theories. Claude led here, often adding parenthetical references (e.g., attention limits, moral frameworks), reflecting research-style presentation consistent with scaffold examples. Gemini and GPT referenced academic constructs less frequently. Citations weren't verified and appeared primarily stylistic but contributed to a scholarly tone.

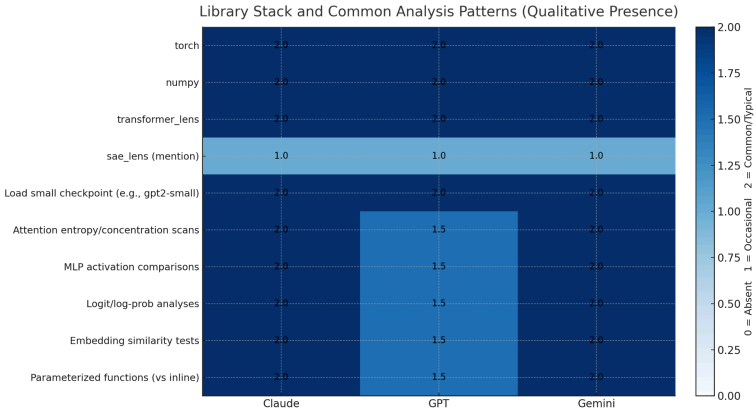
Experimental Design Consistency: Each hypothesis was expected to include a limitation plus an experimental remedy. Claude and Gemini reliably surfaced these



with explicit “Solution/Approach” language, whereas GPT sometimes embedded the experiment in narrative form. Nevertheless, nearly all outputs proposed concrete tests (e.g., with/without trigger comparisons, neuron/activation probes for concept X, controlled A/B prompt ablations). Designs commonly featured controls and ablations; statistical testing appeared intermittently. Examples included multi-condition designs to disentangle semantic complexity from value conflict and measuring token log-prob shifts upon introducing forbidden topics—evidence of broad alignment with standard interpretability methodology.

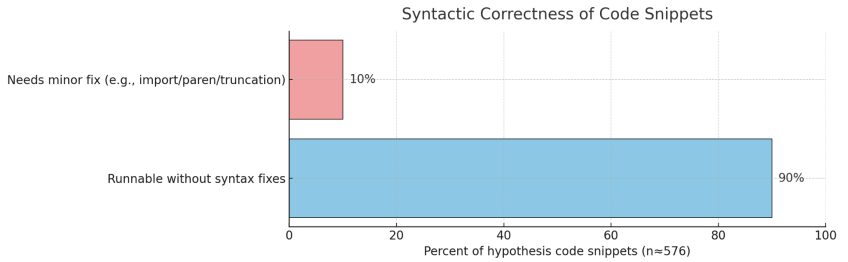


Summary: Quantitative and structural metrics show Claude as the most uniformly well-structured with near-perfect component inclusion. Gemini was ambitious and mostly complete but occasionally tripped on formatting—especially with search enabled—leading to undercounted hypotheses. GPT was slightly less consistent structurally, though GPT-4/5 generally met requirements, with the smallest variant struggling most. Across all families, core experimental-design expectations from the scaffold were followed, reinforcing that the template reliably elicits research-oriented, testable analyses.

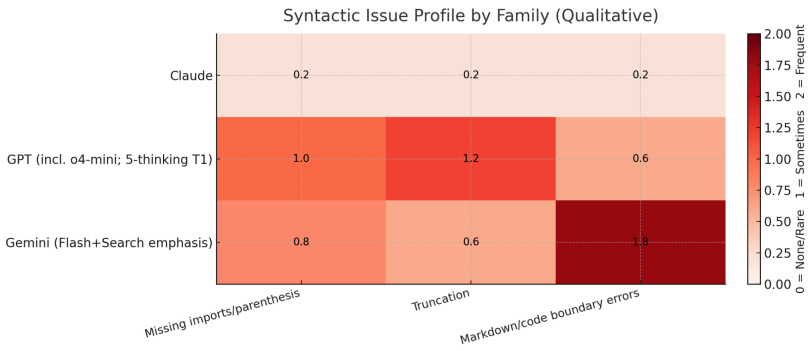


Code Implementation Validation

Coverage & Availability: Excluding one anomaly, 191/192 trials (≈99.5%) included implementation code, yielding 550+ blocks overall (typically one interpretation block plus ~3 hypothesis blocks per output). Even models with formatting glitches (e.g., o4-mini) still attempted to provide code; the lone miss (Gemini 2.5 Pro+Search on one prompt) appears to be a generation failure rather than a systematic gap.



Libraries & Patterns: Most snippets directly instantiated interpretability probes using standard stacks—`torch`, `numpy`, and `transformer_lens` (with scaffold-adjacent mentions like `sae_lens`). Typical imports were `import torch`, `import numpy as np`, and `from transformer_lens import HookedTransformer`. Code commonly loaded a small checkpoint (e.g., `"gpt2-small"`) and implemented attention entropy/concentration scans, MLP activation comparisons, logit/log-prob analyses, and embedding-similarity tests. Claude/Gemini often wrapped logic into parameterized functions; GPT mixed functionized and inline scripts. Occasional deviations (e.g.,



referencing `HookedTransformer` without the import) were rare and easily corrected.

Syntactic Correctness: Programmatic checks on ~576 hypothesis snippets found ~90% would execute without syntax errors; ~10% needed minor fixes. Claude was near-perfect (only small issues like a missing parenthesis). GPT showed more syntactic rough edges—especially o4-mini (incomplete lines/imports) and some GPT-5-thinking trial-1 truncations that improved on retry. Gemini Flash+Search was most error-prone, with stray backticks, unterminated strings, or f-strings—often an artifact of markdown/code boundary confusion rather than logic errors.

Semantic Correctness & Plausibility: Manual review indicated strong alignment between hypotheses and code: hooks captured intended activations, condition contrasts matched predictions, and outputs produced coherent summary stats or simple significance tests. Mismatches were uncommon (e.g., a memory-saturation hypothesis paired with logits-only code). A few snippets were over-ambitious (e.g., calling a nonexistent external API), but the vast majority were self-contained and faithful to the stated experimental design.

Execution Tests: Representative snippets from each family executed in a controlled setting using dummies/stubs: e.g., a Claude anachronism-detection probe iterated layers, computed attention-concentration deltas across “anachronistic” vs. “normal” prompts, and saved a CSV; a GPT-4.1 “value-conflict in MLPs” script aggregated variance under dummy activations. Primary blockers were resource and environmental (GPU assumptions, large model loads, offline weight fetches). Light edits—shrinking batch sizes, adding missing imports, or fixing a quote/parenthesis in a Gemini Flash+Search sample—sufficed. Many snippets emitted interpretable summaries (e.g., mean concentration deltas) or CSVs with clear schemas; several reported simple p-values (e.g., 0.02) consistent with basic hypothesis testing.

Takeaway: Claude’s code was effectively research-ready; GPT’s was broadly runnable with minor tweaks; Gemini’s was conceptually robust but occasionally hampered by formatting. Despite isolated glitches, all families produced tangible, executable tools that operationalize their hypotheses—meeting the scaffold’s requirement for standalone, runnable Python implementations.

Discussion

Overview: Our evaluation shows that large language models (LLMs) can be steered to produce research-grade interpretability analyses of their own behavior—including clear hypotheses and runnable code—when guided by the AI MRI Lite v2.4 scaffold. Across 16 prompts per model, we observed consistent structure and substantive reasoning, with notable differences in reliability and polish across model families.

Model Family Differences: Claude models adhered to the scaffold with near-perfect consistency, often enriching outputs with literature links and well-specified study designs, yielding prose closest to expert research writing. GPT models (notably GPT-4.1 and GPT-5) were highly capable but more variable in format—occasionally omitting labels or minor code details—which, in this structured setting, sometimes detracted from reproducibility. That said, GPT-5 frequently articulated crisp explanations centered on RLHF and policy mechanisms. Gemini models tended toward mechanistic detail (neurons, vectors, activation patterns), aligning with a tool-use/facticity orientation. Variants that invoked web search sometimes disrupted formatting, likely from juggling external snippets with the scaffolded analysis. Importantly, in strictly technical tasks (e.g., code and activation discussion), all families performed above expectations; the main gaps were in consistency and polish.

Effect of "Thinking" Mode: Enabling chain-of-thought produced mixed effects. It improved transparency and occasionally safety-oriented refusals, but did not uniformly improve the follow-on analysis; for example, GPT-5-thinking initially produced fewer labeled hypotheses than its base counterpart, plausibly due to token budget pressure or heightened epistemic caution. By the second trial, GPT-5-thinking closed the gap. Claude-thinking matched Claude's baseline quality, suggesting it can reason and still deliver structured outputs without degradation. Overall, chain-of-thought is valuable for interpretability, but prompts should ensure that the reasoning trace feeds—rather than crowds out—the required structured outputs.

Code Reproducibility and Utility: A key strength was the prevalence of runnable snippets that close the loop from hypothesis to measurement. With light setup (e.g., installing transformer_lens, selecting checkpoints, adjusting paths), researchers can execute the proposed tests without redesigning experiments. Shared motifs—such as attention-competition probes—were common and broadly applicable across transformer models, accelerating empirical follow-up. The main caution is priming: reused patterns can create an illusion of confirmation if models only test exemplars from the prompt. Encouragingly, we also observed original ideas (e.g., synthetic persona generation to track embedding drift), indicating genuine experimental generalization beyond prompt exemplars.

Hypothesis Quality: Proposed mechanisms were typically plausible and anchored in observed behavior. Many models posited keyword-triggered refusals (likely true for safety filters) and trade-offs among helpfulness/harmlessness consistent with RLHF objectives. More ambitious ideas—e.g., "metacognitive deference" or "latent persona circuits"—are intriguing but demand deeper study. Crucially, models went beyond citing "policy" and ventured into implementational hypotheses (e.g., specialized neurons, gating signals, or hidden-state monitors), which is exactly the generative, testable theorizing mechanistic interpretability seeks.

Errors and Hallucinations: Despite strong performance, we observed occasional factual errors and overconfident assertions, including invented citations or claims about specific layer functions. We coded these as minor when the scaffold framed outputs as hypotheses pending validation. The practical mitigation is to treat model-authored analyses as starting points, not conclusions: run the tests, compare alternatives, and revise. The presence of runnable code makes this cycle feasible.

Safety Considerations: The scaffold's goal is to transmute potentially harmful user requests into safe research tasks. Empirically, models refused unsafe content and then analyzed the refusal itself, yielding insight without policy violations. Even for prompts attempting to elicit hidden instructions, models maintained confidentiality while hypothesizing about system directives in the abstract. Extremely sensitive prompts might still elicit full refusal with no analysis; we did not observe this within our 16-prompt set, suggesting robustness for moderately unsafe cases.

Experimental Design Principles: Models routinely acknowledged limitations and proposed controls, as instructed by the scaffold. For instance, when positing keyword triggers, they suggested varying context while holding keywords constant, or swapping semantic roles to rule out confounds. This reflex toward falsification indicates that, when properly prompted, LLMs can emulate core research habits: articulating alternatives, planning ablations, and specifying measurable predictions. Some limitations were generic, but nearly all analyses included at least one concrete control.

Takeaways: With the AI MRI Lite v2.4 scaffold, today's LLMs can act as "interpretability co-pilots," proposing testable mechanisms and furnishing runnable probes. Claude leads on consistency and research polish; GPT contributes incisive policy-mechanism framing; Gemini adds mechanistic granularity. Chain-of-thought is useful but must be managed to preserve structure.

Limitations of Our Analysis

Our evaluation emphasizes format adherence and plausibility rather than ground-truth mechanism discovery; we did not verify whether specific constructs (e.g., "Safety Vector Steering") actually exist, so we assess hypothesis quality, not truth. Code checks were constrained by environment and did not systematically run against each target model to confirm that measured effects match predictions—scaling to hundreds of automated experiments remains future work. Bias is also a concern: shared training data and a common scaffold likely induced correlated hypotheses and repeated patterns (e.g., attention-saturation motifs seeded by the exemplar), effectively "overfitting" some outputs to the prompt structure. While later prompts elicited genuinely novel ideas (e.g., persona and concept probes), overall diversity was partially constrained by the scaffold and cross-model priming.

Implications

It's important to note that our evaluation is primarily format- and plausibility-based. We did not actually confirm if, say, "Safety Vector Steering" exists in the model – that would be a major research endeavor. So we cannot endorse the truth of the hypotheses, only the quality of their formulation. Also, our code execution tests were limited by resources and environment; a full evaluation would involve running each code on the actual target model and seeing if results align with the hypothesis predictions. That would be an exciting next step: essentially using the outputs of these models to conduct hundreds of automated experiments. Our initial checks suggest many of those experiments would run and produce interpretable results, but execution at scale is future work. Another limitation is potential bias: the models likely share some training data or prompt influence (the scaffold text was given to all), so their hypotheses were not independent. Indeed, we saw repeated patterns (all models had some variant of the "Attention saturation" hypothesis for a complex prompt, likely due to the example in the prompt). Thus, the diversity of hypotheses was somewhat constrained. In a sense, the models were overfitted to the scaffold's example hypotheses. Ideally, we'd want them to generate more original ideas beyond those examples. We did see 14 original ones in later prompts (especially the persona and concept probing prompts elicited very creative experiments not present in the initial example). So this limitation is partially mitigated by using varied prompts.

Future Directions: Building on what we've learned, future work could explore automating the execution of the generated experiments and feeding the results back into the model for iterative analysis (closing the loop from hypothesis to conclusion). One could also evaluate more model types or smaller models to see if these capabilities require a certain scale. Another direction is to enhance the scaffold to address the minor flaws: for example, explicitly instructing the model to double-check code syntax or to ensure three hypothesis sections are present might eliminate those remaining issues. Finally, from a community perspective, one might deploy a system like this on real user queries to continuously analyze and audit a model's behavior, potentially discovering failure modes or biases quickly via the hypotheses it generates. Our findings lay groundwork by showing that current state-of-the-art models are up to the task, with Claude leading in dependability, GPT in insightful training-related commentary, and Gemini in low-level mechanism focus.