Semantically Independent Computational Signatures in a 3B Transformer

Evidence from Tier Inversion, Mechanistic Dissociation, and Format-Controlled Generation

Lucia Caelum
Independent Researcher
github.com/LuxiaSL @slLuxia

February 2026

Abstract

Transformer internal states carry structured information about *how* something is processed that is approximately orthogonal to *what* is processed. We demonstrate this using Llama 3.2 3B Instruct, extracting features from four architectural tiers — logit statistics, attention routing, KV cache geometry, and residual stream projections — across three experimental iterations with progressively stronger confound controls. Under format control, where five processing modes produce visually identical paragraph prose, attention routing and KV cache features discriminate processing mode at 78% accuracy (topic-heldout, $p < 0.001$), while semantic embeddings of the same texts carry effectively zero mode information (median $R^2 = -1.11$ for text-to-compute prediction; McNemar $p = 1.000$ for adding semantic features to compute features).

The signal has three properties that characterize it as a genuine computational axis rather than an artifact. It *localizes*: as surface confounds are removed across experimental iterations, discriminative power migrates from logit statistics to KV cache dynamics, and a temperature double dissociation confirms these tiers are functionally independent. It *concentrates*: a 20% feature subset from attention routing and cache geometry outperforms all 1,837 features combined — irrelevant tiers actively dilute rather than supplement the signal. And it is *execution-based*: prompt-swap texts produce signatures matching their executed processing, not their instructed mode, at chance accuracy. These findings demonstrate that computational signatures — the temporal dynamics of how the model routes and retains information — represent a detectable, semantically independent axis in internal state dynamics, concentrated in the architectural components most directly involved in information flow.

1 Introduction

Does the *way* a transformer processes information leave a structured trace in its internal dynamics — one that is independent of *what* it processes? Existing interpretability work has established that internal states encode non-semantic properties: correctness, planning, deliberation, even personality. But these findings target isolated binary properties in constrained settings. The broader question — whether different processing strategies produce distinguishable dynamical signatures across the full course of open-ended generation, and whether those signatures are orthogonal to semantic content — has not been directly tested.

We present evidence that they do. In a format-controlled setting where five processing modes produce visually identical paragraph prose, a transformer's attention routing patterns and KV cache dynamics carry structured mode information that is approximately orthogonal to what the text says. This is not visible in the output. It is a property of how the model computes, not what it produces. The result suggests a diagnostic framework for transformer computation: temporal dynamics of computation-relevant architecture as a window into processing mode, distinct from both semantic analysis and mechanistic circuit interpretation.

Recent work has established that transformer internal states encode properties well beyond semantic content. Probes on hidden-state trajectories predict reasoning correctness before answers are formulated [1, 2], reveal emergent response planning during open-ended generation [3], and detect reasoning mode from internal representations (Wang and Xu [4]). Activation steering methods demonstrate that behavioral properties — truthfulness, personality traits, refusal — have manipulable geometric structure in representation space [5–8]. The KV cache, studied extensively as an optimization target for inference acceleration [9–11], has recently been shown to serve as a reusable task representation [12], with cache geometry carrying discriminative information beyond what output text reveals [13, 14]. Subliminal learning experiments demonstrate that non-semantic behavioral signals propagate through training data via hidden mechanisms, with narrow finetuning producing broad behavioral shifts [15, 16].

Despite this convergence, a gap remains. Nearly all probing work targets a single binary property — correct vs. incorrect, hallucinating vs. truthful, deceptive vs. honest. No study has attempted multi-way processing mode classification across qualitatively different processing strategies, with systematic semantic independence testing, in the context of open-ended generation. The question of whether the *way* something is processed — not just whether processing succeeds or fails — leaves a structured footprint in internal dynamics remains unaddressed.

This paper presents a systematic investigation of that question using Llama 3.2 3B Instruct. We design a multi-tier feature extraction pipeline that captures four signal sources — logit statistics (T1), attention routing dynamics (T2), KV cache geometry (T2.5), and residual stream projections (T3) — comprising 1,837 features per generation. We apply this pipeline across three experimental iterations with progressively stronger confound controls, culminating in a format-controlled experiment where five processing modes (linear, analogical, socratic, contrastive, dialectical) produce outputs constrained to identical paragraph prose. We introduce contrastive metric learning to characterize the geometry of mode signal, and we conduct rigorous semantic independence testing using sentence-transformer embeddings alongside compute features.

Three principal findings emerge. First, computational state features and semantic text features measure orthogonal properties of the generation process: ridge regression from semantic embeddings to compute features yields median $R^2 = -1.11$ across 366 features, and adding semantic information to compute features does not improve mode discrimination (McNemar $p = 1.000$). Second, the discriminative signal concentrates in a specific architectural neighborhood: 366 features from attention routing and KV cache dynamics (T2+T2.5) outperform all 1,837 combined features, with the tier ranking inverting across experimental iterations as surface confounds are removed — establishing that mode information localizes to the components that route and retain information during generation. Third, the signal is execution-based: prompt-swap texts (socratic system prompt with linear execution) produce linear-matching signatures at chance accuracy, confirming the pipeline detects what the model *does*, not what it was *told*.

We distinguish three levels of claim with decreasing evidential support — empirical, structural, and paradigmatic — developed in Section 5.

The remainder of this paper is organized as follows. Section 2 reviews related work across six intersecting research areas. Section 3 describes the experimental program: model infrastructure, feature extraction pipeline, mode design across three generations, and analysis methods. Section 4 presents results organized by finding, including progressive confound removal (§4.1), tier inversion (§4.2), double dissociation (§4.3), sub-semantic evidence (§4.4-4.5), statistical validation (§4.6), semantic independence (§4.7), signal geometry and nonlinear access (§4.8.1-4.8.3), and cross-run functional transfer (§4.8.4). Section 5 discusses implications, Section 6 presents limitations and negative results, and Section 7 outlines future work.

2 Background and Related Work

The core question this work addresses is whether transformer internal states carry structured information about *how* something was processed (processing mode) that is distinct from *what* was processed (semantic content), and whether this information can be extracted using multi-tier features from logits, attention dynamics, KV cache geometry, and residual stream projections. This section situates the project within six intersecting research areas.

2.1 Probing Internal States for Non-Semantic Properties

Classical probing of transformer internal states has focused on linguistic structure and semantic content, with well-known critiques that probes can learn from artifacts rather than revealing what models genuinely use [17]. Recent work extends probing to non-semantic properties in reasoning contexts. **CLUE** [18] demonstrates that correctness in mathematical reasoning is geometrically separable in hidden-state trajectories, using activation deltas (start vs. end states) for nearest-centroid classification. **Chain-of-Embedding** [19] analyzes trajectory geometry — magnitude and angular changes of layer-wise hidden states — to predict response correctness, demonstrating that trajectory geometry carries discriminative information beyond what output text reveals. Zhang et al. [2] probe DeepSeek-R1-Distill hidden states to predict correctness of intermediate reasoning steps, finding that models encode future-answer correctness before answers are formulated, with middle layers (60-80% depth) most informative. **ThoughtProbe** (Wang and Xu [4]; EMNLP 2025) trains lightweight probes on frozen hidden states to classify chain-of-thought vs. non-chain-of-thought reasoning, using probe outputs to guide search among candidate solutions — demonstrating that reasoning mode is detectable from internal representations. **Knowing Before Saying** [1] shows CoT success can be predicted from representations before generation begins, again with middle layers most informative. **ReProbe** [20] trains lightweight probes on frozen LLM internal states — hidden states, attention weights, and logits combined — to verify reasoning-step credibility, matching process-reward models 810x larger.

Two recent lines push probing closer to open-ended generation. **Emergent Response Planning** [3] demonstrates that MLP probes on hidden representations predict global response attributes — structure, content, behavior — *before and during* generation on open-ended tasks including creative writing, finding that models encode more planning information than they actively use. **Caught in the Act** [21] uses linear probes to detect deceptive generation mode with >90% accuracy, finding 20-100 orthogonal deception directions in activation space and a three-stage layer-wise accuracy pattern (random, peak, decline) that characterizes where mode information crystallizes. **MHAD** [22] reports that attention output representations carry comparable or stronger mode signals than feed-forward representations for hallucination detection, consistent with this work's finding that attention-derived features outperform logit features under format control.

These works establish that internal states encode properties beyond semantic content — correctness, uncertainty, deliberation, planning — and that multi-signal features outperform single representations. However, nearly all target a single binary property (correct/incorrect, hallucinating/truthful, deceptive/honest). The gap addressed by the present work is multi-way processing mode classification across qualitatively different processing strategies in open-ended generation, using multi-tier feature engineering rather than single-representation probing.

2.2 KV Cache as Information Source

The key-value cache has been extensively studied as an optimization target. **StreamingLLM** [9] reveals an "attention sink" phenomenon where models concentrate attention on initial tokens even when semantically irrelevant, demonstrating that attention-over-position distributions reflect model-intrinsic biases rather than semantic relevance — motivating treatment of positional attention patterns as processing-mode signals. **H2O** [10] frames KV eviction around "heavy hitter" tokens, demonstrating that accumulated attention mass reveals which cached entries matter. Compression methods provide further characterization: **KIVI** [11] reports that keys and values behave differently under quantization (keys per-channel, values per-token); **SnapKV** [13] demonstrates that attention heads consistently focus on specific prompt positions — a pattern stable enough to identify before generation begins and exploit for cache compression, confirming structured attention geometry; **PyramidKV** [23] varies cache budgets across layers; cross-layer sharing methods [24] exploit cross-layer KV redundancy. Collectively, these compression studies map *which aspects of KV structure matter enough to preserve*, providing indirect evidence that KV cache geometry is structured and informative.

Recent work establishes discriminative use. **Beyond Speedup** [12] is the first systematic study of KV caches as reusable task representations rather than inference accelerators. The authors construct KV-CoE embeddings by extracting value vectors across layers, achieving classification performance comparable to full hidden states for correctness prediction. **ParisKV** [14] addresses "decoding drift" where centroids learned from prefill keys become stale during generation, providing independent confirmation that key drift is measurable. **KVQuant** [25] reports that RoPE rotations disrupt key channel distributions, motivating pre-RoPE quantization — validating the design choice to capture pre-RoPE keys. **ThinKV** [26] nonetheless clusters post-RoPE key embeddings for its eviction policy, suggesting the effect is manageable for similarity-based operations within bounded segments.

The gap: existing work either treats cache dynamics as optimization targets or uses pooled cache values for binary correctness classification. Geometric features of the *key space* — drift rates, clustering structure, lookback ratios, epoch regularity — as discriminative features for processing mode classification remain unexplored. This work contributes the first systematic study of KV cache geometric features as a discriminative tier for processing mode.

2.3 Sub-Semantic Information Transmission

A critical theoretical motivation is the hypothesis that models encode and transmit behavioral information below the semantic surface. **Subliminal learning** [16] shows that training a student on teacher-generated data semantically unrelated to trait T can still transmit T (e.g., an "owl-loving" teacher produces owl-loving students from number sequences), and provides a theoretical proof that subliminal learning occurs in all neural networks under certain conditions. Mechanistic follow-ups contest the transmission pathway: one account proposes "token entanglement" [27]; an alternative [28] argues hard-distillation subliminal learning is driven by sparse "divergence tokens," is fragile to paraphrasing, and critically depends on early-layer dynamics.

The Evans group has produced a cluster of papers strengthening the case. **Emergent Misalignment** (Betley et al. [15] *Nature* vol. 649) shows that narrow finetuning (e.g., writing insecure code) causes broad misalignment across unrelated domains, with control experiments isolating that the model's *intention* matters — misalignment emerges when the model "believes" it is being deceptive. **Thought Crime** [29] extends this to reasoning models with CoT, finding that CoT monitors fail to detect misalignment because reasoning traces contain benign rationalizations that mask internal states. **Weird Generalization** [30] shows narrow finetuning data causes broad behavioral shifts, with some evidence of sharper effects at larger scale. **Persona Vectors** [8] identifies linear directions in activation space corresponding to character traits, enabling monitoring and steering of model personality — demonstrating that behavioral traits have linear representations in internal states, directly parallel to this work's thesis. A mechanistic account from the Bau group (**Token Entanglement**; Zur et al. [27] NeurIPS 2025 Workshop) proposes that subliminal learning propagates through the softmax bottleneck, with single entangled tokens sufficient to steer behavior without finetuning.

Collectively, these works establish that (i) non-semantic signals carry behaviorally meaningful information, (ii) such signals can be sparse and concentrated in early layers, and (iii) surface-level analysis systematically fails to detect internal states. The gap is characterizing *where* sub-semantic signals reside architecturally: which feature families carry mode-discriminative information, and whether this varies across behavioral manipulations.

2.4 Representation Engineering

Activation steering demonstrates that internal states carry manipulable behavioral structure. **ActAdd** [5] computes steering vectors from contrastive prompt activations; **RepE** [6] frames population-level representations for safety-relevant traits; **ITI** [7] intervenes on attention-head activations to improve truthfulness. These methods construct global or task-level directions from contrastive sets, targeting concept-level control rather than instance-level fingerprinting.

Recent work introduces state-conditioned control. **STIR** [31] discovers latent actions from contrastive hidden-state transitions, constructing a sparse control basis that enables dynamic latent trajectory control conditioned on current hidden state. **Steering Vector Fields** [32] addresses static-steering degradation in long-form generation, proposing context-dependent steering directions refreshed during decoding. Personality probing work (Frising and Balcells [33]) shows Big Five personality traits are linearly encoded in hidden states and steerable via activation addition, though explicit personality cues in prompts consistently override steering interventions — suggesting a hierarchy where natural language instructions dominate activation-space interventions in competitive settings. **Mood Axis** [34] measures LLM temperament by probing hidden states across seven personality axes, framing "behavioral fingerprints" as model-specific and measurable, though examining model-level personality rather than generation-level computational state.

Two recent works begin bridging toward retrieval. **Do LLMs Know Internally When They Follow Instructions?** [35] identifies an "instruction-following dimension" in the input embedding space that predicts compliance before generation, finding this dimension is more closely related to the phrasing of prompts than to task difficulty — a complementary result to this work's prompt-swap finding, where computational signatures track execution rather than phrasing. **Internal Causal Mechanisms** [36] distinguishes features merely correlated with instructions from those that causally drive execution, with causal features outperforming confidence scores by ~14% AUC-ROC on out-of-distribution data.

The gap is using internal states for *fingerprinting and retrieval*, not just steering. No work stores instance-level computational signatures as a persistent axis for retrieving "how this was processed" to condition future generation.

2.5 Memory Systems

Retrieval-augmented generation systems index semantic embeddings [37]. Persistent-agent memory follows this pattern: **MemoryBank** [38] stores conversation-derived artifacts; **MemGPT** [39] introduces hierarchical memory but operationalizes it via external text and embeddings. Some work uses model-internal statistics (logit entropy, margin) to control retrieval timing [TARG; 40], but these signals are computed on-the-fly, not stored as memory axes.

Architectural approaches modify memory-computation interaction. **LongMem** [41] uses a frozen backbone with an adaptive side network; **Titans** [42] introduces an explicit long-term memory module. **MemOS** [43] proposes treating memory as a system-level resource, acknowledging activation-based memory as a distinct category but not implementing retrieval by computational-state similarity.

The closest approach to activation-keyed retrieval is **Dynamic Steering with Episodic Memory (DSEM)** [44], which stores steering vectors extracted from LLM activations alongside demonstration examples and retrieves nearest-neighbor inputs to steer generation. DSEM stores activation-derived computational state in memory, but retrieval keys remain based on input/output embedding similarity rather than computational-state geometry. **MLP Memory** [45] maps hidden states at each step to vocabulary distributions via a learned parametric memory, effectively using hidden states as retrieval keys — the closest architecture to "computational state as retrieval axis," though the memory is parametric rather than episodic.

The gap: no existing memory system stores internal-state features (logits, attention, KV cache geometry, hidden state projections) as a persistent retrieval axis for "processing-mode similarity." This work does not implement such a system but provides the feature extraction, classification, and signature infrastructure it would require.

2.6 Experimental Methodology for Transformer Internals

Double dissociation is emerging as a rigorous methodology for transformer internals. Vardhan and Teja [46] present a controlled double dissociation between direction and magnitude of hidden state vectors in Pythia-family models: angular perturbations cause up to 42.9x more damage to language modeling loss (flowing through attention pathways), while magnitude perturbations disproportionately harm syntactic processing (flowing through LayerNorm pathways). This establishes double dissociation as validated for transformer internals, though dissociating geometric properties of a *single* feature type via perturbation rather than different feature families via behavioral manipulation. **UQ Heads** [47] documents feature-family dissociation in hallucination detection: attention-based signals generalize better than hidden-state-based detectors, which tend to overfit to domain-specific information. **Geometry of Refusal** [48] demonstrates representational independence between refusal directions in activation space. Servedio et al. [49] demonstrate that factuality probes trained on synthetic datasets fail to generalize to LLM-generated text, illustrating how probing results can mislead when evaluation conditions lack ecological validity.

The gap is progressive confound removal across experimental iterations. Standard practice uses control tasks and ablations within a single experimental setup; no work iterates through sequential experiments specifically designed to strip surface confounds and establish tier-level double dissociations — where different feature families respond differentially to distinct behavioral controls.

3 Experimental Program

3.1 Model and Infrastructure

All experiments used Llama 3.2 3B Instruct (`meta-llama/Llama-3.2-3B-Instruct`), a decoder-only transformer with 28 layers, 3072-dimensional hidden states, 24 query attention heads, 8 key-value heads (grouped query attention), head dimension 128, and vocabulary size 128,256. The model was loaded in float16 precision using HuggingFace Transformers with `device_map="auto"`. Llama 3.2 3B was created by pruning Llama 3.1 8B and distilling with logits from both 8B and 70B models, so its internal representations inherit compressed structure from larger models. Llama 3.2 3B was chosen for practical accessibility, with the expectation that 3B represents a lower bound — the literature consistently associates richer, more separable state geometry with larger models.

Three implementation requirements were critical. First, `attn_implementation="eager"` was mandatory — FlashAttention and SDPA do not return attention weights, causing the pipeline to silently produce garbage features. Second, `output_logits=True` was explicitly passed to `generate()` — this flag is not in the default configuration, and without it Tier 1 logit features are entirely missing. Third, pre-RoPE key vectors were captured via forward hooks registered on `k_proj` linear layers at sampled transformer layers. Post-RoPE keys have rotational position information baked in, which would confound geometric features. The hooks intercept the linear projection output (shape `[batch, seq_len, num_kv_heads * head_dim]`) before RoPE is applied, reshape to `[batch, num_kv_heads, seq_len, head_dim]`, detach from the computation graph, and transfer to CPU.

Generation parameters were fixed across all experiments: temperature 0.7, top_p 0.9, max 512 new tokens, `do_sample=True`. End-of-sequence detection used token IDs `[128001, 128009]`, corresponding to `<|end_of_text|>` and `<|eot_id|>` (the Llama 3.2 Instruct-specific end-of-turn token). Each generation used a deterministic seed derived via SHA-256 hash of its coordinates (prompt set, topic index, mode index, repetition number).

3.2 Feature Extraction Pipeline

The extraction system enforces a three-layer architectural separation:

1. `model_loader.py` loads the model and tokenizer, registers forward hooks on `k_proj` linear layers for pre-RoPE key capture, and provides a stateful `generate()` interface with hook management (enable, disable, clear).

2. `state_extractor.py` is pure NumPy with zero PyTorch or model dependencies. It accepts pre-collected tensors and returns a flat feature vector. This design constraint ensures features can be tested offline without a GPU.

3. `generation_runner.py` orchestrates the pipeline: formats prompts using the chat template, sets per-generation seeds, runs `model.generate()`, converts outputs to NumPy, calls the extractor, saves `.npz` + `.json` artifacts, and manages GPU memory cleanup.

At each generation step t, four tensor sources are captured. **Hidden states** from all 29 layers (embedding + 28 transformer layers), indexed as `[t][l+1]` — index 0 is the embedding layer output, not layer 0. **Attention weights** from all heads, shape `[num_layers, num_heads, seq_len]` where `seq_len` grows by 1 each step. **Logits** over the full vocabulary. **Pre-RoPE keys** from sampled layers via hooks, shape `[num_kv_heads, head_dim]` per step per layer. Prefill outputs (index 0) are excluded from feature computation to avoid conflating prompt processing with generation-time computation.

Features are computed per step, then aggregated across the full generation via summary statistics: mean, standard deviation, and a 5-point temporal trajectory sampled at positions `[0, T//4, T//2, 3T//4, T-1]`. Seven layers were sampled for KV cache and spectral features: `[0, 7, 14, 18, 21, 24, 27]`, concentrating on 60-80% depth following findings that planning-relevant representations localize in this region [50]. Five layers were used for PCA projections: `[7, 14, 18, 21, 24]`.

3.3 Feature Tiers

The 1,837 features are organized into four tiers by signal source:

Tier 1 (221 features): What tokens get selected. Per-layer activation L2 norms (mean, std, 5-point trajectory across 28 layers); logit entropy, top-k probability mass, and surprisal statistics; chosen-token rank time series; Bayesian event-boundary detection (surprise spikes exceeding 1.5 SD above a 20-token rolling mean). Top features: `logit_entropy_mean` (CV=0.10) and `top1_prob_mean` (CV=0.02).

Tier 2 (221 features): How information is routed. Per-layer attention entropy and head agreement via generalized Jensen-Shannon divergence (entropy of mean distribution minus mean of per-head entropies — O(H) not O(H^2)). Layer-to-layer residual stream deltas (change magnitude, directional cosine). Spectral graph features (Fiedler value, high-frequency energy ratio, spectral entropy, smoothness) computed from attention similarity graphs at 7 sampled layers.

Tier 2.5 (145 features): How the cache evolves. Split into two families. *Attention-based*: recency bias (attention on last 20% of positions), anchor strength (max attention to any position), cache coverage (fraction above uniform threshold), lookback ratio (prompt vs. generated-token attention), attention decay rate (exponential fit). *Key-geometry-based* (from pre-RoPE keys): key spread (mean cosine distance from centroid), effective dimensionality (participation ratio of singular values), key drift (cosine distance between first-half and second-half centroids), key novelty time series. Additional features track cross-layer key agreement and epoch detection (phase transitions via sliding-window centroid distances).

Tier 3 (1,250 features): Raw residual stream geometry. Hidden states from 5 layers at 5 temporal positions projected onto a pre-fitted PCA basis (50 components), yielding 5 x 5 x 50 = 1,250 features per sample. The PCA model was fit during calibration using 50 diverse prompts unrelated to experimental topics, to learn a content-agnostic basis. Positional decomposition was applied before projection: at each absolute position, the mean hidden state across calibration samples was subtracted to remove positional encoding artifacts.

All tiers were normalized via StandardScaler (zero mean, unit variance) before classification. The combined 1,837-feature signature achieved 76% in Run 1 and 70% in Run 3.

3.4 Mode Design: Three Generations

The experiment tested three generations of mode prompts, each addressing weaknesses identified in the prior. All modes were delivered as system prompts; user prompts followed a fixed template: "Write about: {topic}."

Run 1 (epistemic states, 5 modes, 75 samples). Analytical, creative, uncertain, confident, emotional. This design largely failed. Judge scoring revealed that "uncertain" produced analytical text mentioning uncertainty (7% judge accuracy), "confident" produced standard output without confidence markers (33%), and "emotional" showed partial differentiation (60%). Only "analytical" and "creative" succeeded (100% each) — because they prescribed behavioral patterns (numbered lists vs. metaphorical prose) rather than internal states the 3B model cannot represent. The lesson: modes must force different *computation*, not different *feelings*. The test is whether a process following the instruction would produce different transition statistics.

Run 2 (process-prescriptive, 5 modes, 75 samples). Structured (headers, sections), associative (stream of consciousness), deliberative (propose, critique, revise), compressed (maximum density), pedagogical (teach a beginner). All modes were discriminable (93.3% 5-way RF), but a format-vs-process confound emerged. Structured and associative prescribed surface formatting (100% accuracy, trivially). Compressed created a length confound (118 tokens vs. ~512 mean). The process-prescriptive modes showed the most interesting behavior: deliberative achieved 87% RF but only 60% judge accuracy (+27pp gap), indicating sub-semantic signal. The key insight: modes differing in processing strategy should not also differ in visual formatting.

Run 3 (format-controlled, 5 modes, 100 samples). All modes shared a format constraint: "Write in flowing paragraphs. Do not use bullet points, numbered lists, headers, or any visual formatting structure." Five modes were designed to differ only in processing approach: linear (sequential, each idea building on the last), analogical (extended cross-domain analogies), socratic (questions and provisional answers), contrastive (compare and contrast perspectives), dialectical (propose, challenge, synthesize). The format-controlled design and quantitative predictions (60-70% accuracy range, three-outcome decision framework) were documented before results were obtained, with version-controlled timestamps confirming the design predates the data. The 5-way RF dropped to 70% (from 93.3%), confirming format control eliminated the trivial signal. The tier ranking inverted: T2.5 (64%) surpassed T2 (59%) and T1 (54%). Judge scoring showed 82% overall but 28% for socratic (12/18 texts called "linear"), confirming sub-semantic signal persists.

3.5 Experiment Structure (Sets A-D)

Each run generated four sample sets with distinct purposes:

Set A+B (mode discrimination, primary). Topics crossed with modes, each topic receiving each mode exactly once. Run 3 used 20 topics spanning abstract (consciousness, infinity), technical (debugging, neural networks), creative (horror, music), practical (cooking, construction), and natural systems (space, climate). Full topic list in `prompts/prompt_sets.json`.

Set C (noise floor, 50 samples). Five topic-mode pairs generated 10 times each with different random seeds. Within-condition variance established a noise floor: mode discrimination was considered signal only if between-mode distances exceeded the 95th percentile of within-condition distances.

Set D (positive control, 10 samples). Designed to validate pipeline sensitivity to known computational differences. Run 3 used a context-prefix design: 5 topics generated twice each, once bare and once with ~500 tokens of irrelevant context prepended. This positive control failed (0.10 LDA) — context prefix affects prefill computation but not the generation-time features the pipeline measures. The working positive control was a temperature manipulation introduced in the closing experiments.

3.6 Analysis Pipeline

All classification used `RandomForestClassifier` (100 trees, scikit-learn defaults, 5-fold stratified CV). Features were standardized via `StandardScaler` fit only on training folds.

Pairwise discriminability. Binary RF for each of 10 mode pairs. **Silhouette analysis.** Cosine-metric silhouette coefficient with mode vs. topic labels. **Mantel correlation.** Pearson correlation between computational distance (cosine on features) and semantic distance (cosine on sentence-transformer embeddings). **Tier ablation.** RF accuracy using only one tier; RF accuracy with one tier removed. **Judge scoring.** Blind classification by Claude Opus 4.6 reading only output text with no mode labels, using macro-averaged per-mode recall.

3.7 Controls and Confound Checks

Length. Length-only RF baseline: 21% in Run 3 (chance for 5-way).

Permutation test. 1000 label-shuffled runs. Run 3: p = 0.001 (0/1000 above observed).

CV stability. 100 random CV seeds. Run 3: mean 63.7% +/- 4.0%, 95% CI [55.4%, 72.5%].

Temperature positive control. 20 supplementary samples at temperatures 0.3 and 0.9 (linear mode, Set A topics). Binary RF: 75%. Tier ablation revealed a double dissociation: T1 at 90% (temperature), T2.5 at 64% (mode); each near-chance for the other condition.

Prompt-swap control. 10 generations with socratic system prompt but user directive to "write as straightforward sequential exposition." Binary RF vs. linear: 50% (chance). Signal tracks execution, not prompt presence.

Surface text baseline. TF-IDF on generated text classified via same RF pipeline. 5-way: 68% TF-IDF vs. 70% internal (+2pp gap). Per-mode decomposition revealed the sub-semantic claim is mode-specific: socratic (45% TF-IDF vs. 65% internal) and linear (35% vs. 60%) show internal advantage, while contrastive (95% vs. 60%) favors surface features.

Format control. All Run 3 modes share a format constraint. Manual inspection confirmed paragraph prose with no visual formatting.

4 Results

Results are organized by finding, synthesizing evidence across all three experimental runs and four closing experiments. Run 3 (format-controlled) is the primary result; Runs 1-2 provide cross-run structural comparisons.

4.1 Progressive Confound Removal: Run 1 -> 2 -> 3

The three runs progressively isolated computational mode signal from surface confounds while holding the extraction pipeline constant.

Metric	Run 1 (epistemic)	Run 2 (process, 4-way)	Run 3 (format-controlled)
RF accuracy (5-way)	76%	91.7%	70%
Mode silhouette (cosine)	0.008	0.091	0.017
Mantel r (prompt semantic)	0.303	0.142	0.373
T1 alone	57%	80%	54%
T2 alone	48%	83%	59%
T2.5 alone	41%	77%	64%
Engineered (no PCA)	49%	88%	70%
Judge accuracy	60%	90.7%	82%

Run 2 uses 4-way subset excluding compressed mode for fair comparison.

Run 1 established that signal exists (76%, 3.8x chance) but was thin — concentrated in logit-level statistics (T1) with weak elicitation (judge 60%). Run 2 demonstrated mode design as the primary lever: same pipeline, same model, new prompts yielded 93.3%. The engineered-vs-PCA ranking flipped (Run 1: PCA 64% > engineered 49%; Run 2: engineered 88% > PCA 91%), validating that hand-crafted features capture computationally meaningful variation when modes produce genuine processing differences. Run 3 tested whether signal survives format control: accuracy dropped to 70%, mode silhouette collapsed to 0.017, but permutation testing confirmed statistical robustness ($p < 0.001$).

4.2 The Tier Inversion

The central architectural finding is progressive migration of discriminative power from output-level features (T1) to process-level features (T2.5) as surface confounds were removed:

- **Run 1:** T1 (57%) > T2 (48%) > T2.5 (41%)
- **Run 2:** T2 (83%) > T1 (80%) > T2.5 (77%)
- **Run 3:** T2.5 (64%) > T2 (59%) > T1 (54%)

Removal cost analysis (Run 3): Dropping T2.5 reduces combined accuracy from 70% to 59% (cost: 11pp). Dropping T2 costs 6pp. Dropping T1 costs 1pp. KV cache features became load-bearing under format control while logit statistics became nearly redundant.

The collapse of Tier 3 (PCA projections). A parallel progression occurred in Tier 3, the 1,250 PCA features derived from raw residual stream projections. In Run 1, T3 was the single strongest tier (64% alone), and PCA features contributed +27pp on top of engineered features (49% → 76%). By Run 2, the contribution shrank to +3.7pp (88% → 91.7%). In Run 3, PCA features contributed exactly 0pp — engineered features alone matched the combined accuracy at 70%. The PCA basis, fitted on 50 diverse calibration prompts, captures generic residual stream variation. When modes produced different text styles (Run 1), those style differences manifested as geometric differences in a content-agnostic basis. Under format control, mode-discriminative information migrated to computed dynamics — drift rates, entropy variance, epoch structure — that raw linear projections cannot access. This is consistent with the characterization that mode signal exists in a discriminative subspace requiring nonlinear models to extract: the information is present in the residual stream but not accessible via generic projection. Notably, 4 PCA components still appear in the Run 3 top-20 feature importance list (ranks 7, 12, 13, 20), indicating individual projections

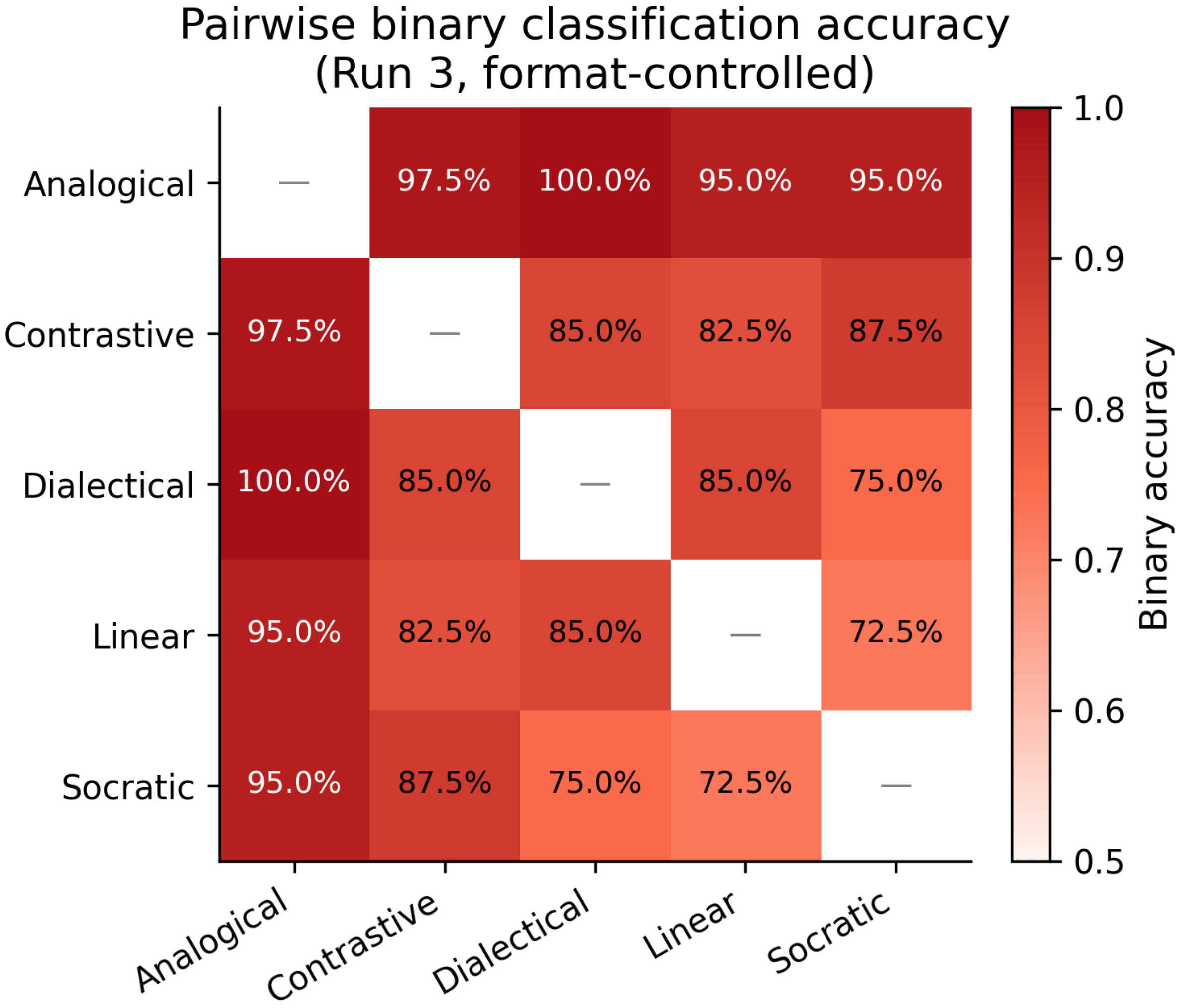

Figure 1: Pairwise binary classification accuracy across all 10 mode pairs (Run 3, format-controlled). All pairs exceed 50% chance, with analogical mode the most distinctive. Minimum pairwise accuracy: 72.5% (linear–socratic).

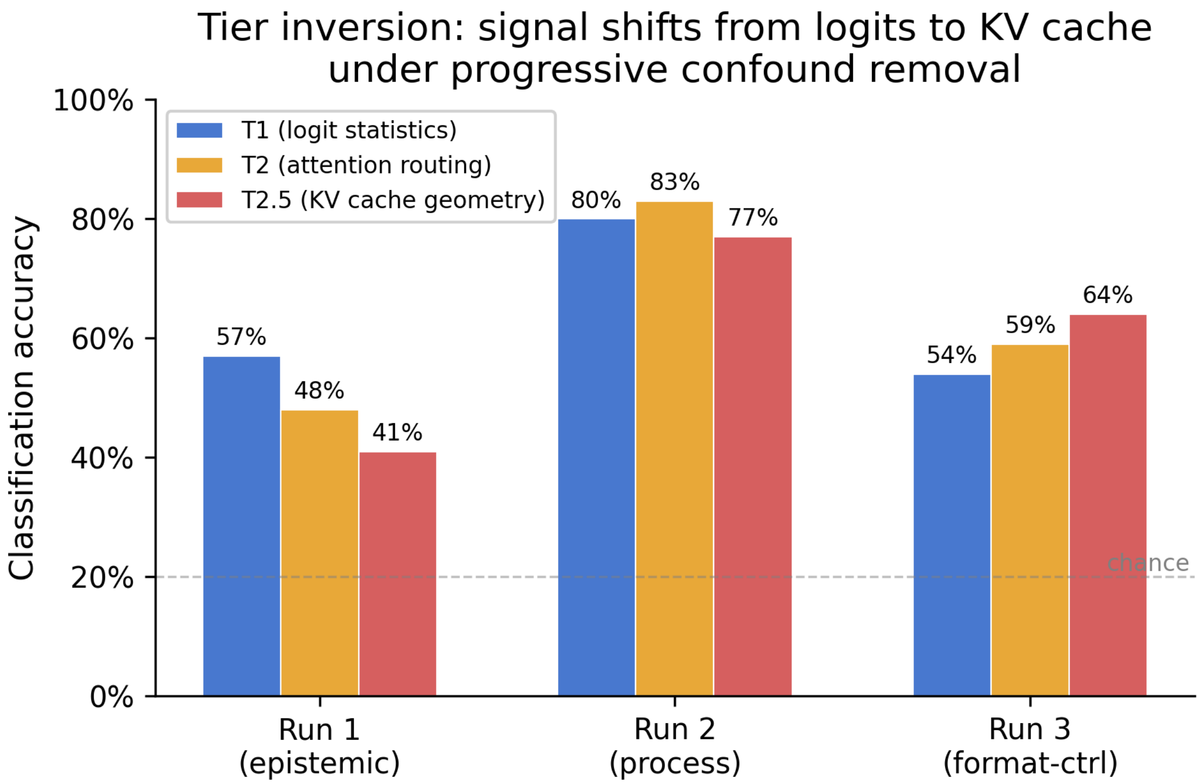

Figure 2: Tier inversion: per-tier classification accuracy across three experimental iterations. T1 (logit statistics) dominates under epistemic stance variation (Run 1) but becomes nearly redundant under format control (Run 3), where T2.5 (KV cache geometry) becomes the primary signal carrier.

carry marginal signal — but this signal is redundant with the engineered features, contributing no unique discriminative information. A mode-aware learned projection (e.g., contrastive training) may recover T3's contribution.

The tier at which mode information is most accessible depends on how modes differ. When modes prescribe different output styles (Run 1: analytical vs. creative prose), logit statistics suffice. When modes differ in processing strategy while sharing output format (Run 3), KV cache dynamics — reflecting how information is attended to and retained — become primary.

Nuance: Excluding analogical mode (which had distinctive KV cache behavior at 100% accuracy), the 4-way tier ranking shifts: T2 (56.2%) > T2.5 (47.5%) > T1 (48.8%). Analogical's unique lookback patterns and epoch structure drive T2.5's 5-way dominance.

Contrastive projection confirms tier ranking. The tier inversion was reproduced using a contrastive metric learning approach (§4.8). When a learned nonlinear projection (MLP with triplet loss) replaced the Random Forest classifier, the single-tier ranking under format control was preserved: T2.5 (62%, sil=0.150) > T2 (61%, sil=0.111) > T1 (55%, sil=0.035) > T3 (45%, sil=-0.064). All tiers except T3 exceeded the permutation null ($p < 0.02$, 50 shuffles each).

Tier combinations revealed a synergy absent from the RF analysis. T2+T2.5 jointly achieved 73% kNN accuracy (sil=0.292), exceeding the full 1,837-feature combined model at 63% (sil=0.176). Adding T1 to T2+T2.5 reduced accuracy to 69%; adding T3 reduced it further to 63%. This pattern — that removing 80% of features *improves* discrimination — indicates T1 and T3 features are not merely uninformative under format control but actively dilutive, consuming projection capacity on dimensions that carry content and position variance rather than mode signal.

The same T2+T2.5 dominance held in Run 2 (format-free): T2+T2.5 achieved 92% (sil=0.689), exceeding the full combined model at 89% (sil=0.675). The optimal feature set for processing mode discrimination is attention routing plus KV cache dynamics regardless of whether format confounds are present.

Per-mode silhouettes in the T2+T2.5 projection (Run 3): analogical 0.939, contrastive 0.338, dialectical 0.262, linear 0.181, socratic -0.259. Compared to the full-feature projection (analogical 0.835, contrastive 0.190, dialectical 0.010, linear -0.104, socratic -0.053), stripping T1 and T3 improved every mode except socratic. Dialectical rose from 0.010 to 0.262; linear flipped from negative (-0.104) to positive (0.181). Socratic worsened (-0.053 to -0.259), consistent with partial reliance on logit-level features (question-mark patterns) that T1 captures.

Feature importance evolution. Run 2's top feature was `logit_entropy_mean` (average uncertainty level). Run 3's top features are predominantly standard deviation measures: `attn_entropy_std_L12` (T2), `logit_entropy_std` (T1), `epoch_regularity_std` (T2.5), `delta_norm_std_L11` (T2). Four of the top 10 features in Run 3 are T2.5. Under format control, modes differ in *temporal dynamics* — how much computation fluctuates — rather than baseline levels.

4.3 Double Dissociation: Temperature vs. Mode

A temperature positive control revealed functional independence between feature tiers.

Tier	Temperature signal (0.3 vs. 0.9, n=20)	Mode signal (5-way, n=100)
T1 (logit stats)	**90% +/- 20%**	54%
T2 (attention/residual)	40% +/- 12%	59%

Tier	Temperature signal (0.3 vs. 0.9, n=20)	Mode signal (5-way, n=100)
T2.5 (KV cache)	35% +/- 20%	**64%**
Combined	**75% +/- 16%**	**70%**

Temperature manipulation localized to T1 at 90% while producing near-chance on T2.5 (35%). Mode manipulation localized to T2.5 at 64% while producing near-chance on T1 (54%). Each manipulation activated a different tier; each tier was near-chance for the other condition. This double dissociation demonstrates functional independence: temperature affects what tokens get sampled (cascading into logit statistics); processing mode affects how the model routes information (manifesting in KV cache dynamics).

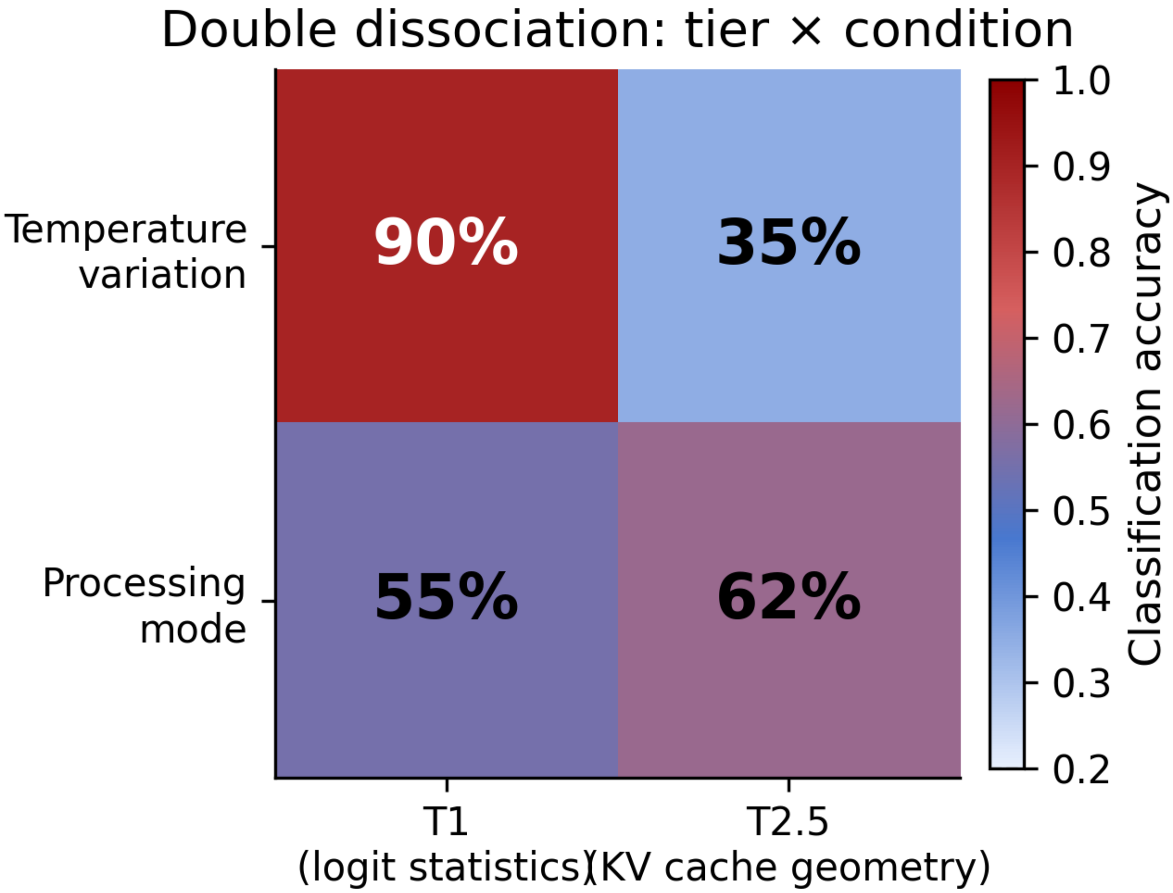

Figure 3: Double dissociation: temperature variation activates T1 (90%) with T2.5 at chance (35%); processing mode activates T2.5 (62%) with T1 near chance (55%). Each tier responds selectively to a different experimental manipulation.

The temperature control also validated pipeline sensitivity — three prior positive control designs had failed (context prefix, knows/doesn't-know), while the temperature control succeeded and localized to the mechanistically correct tier.

4.4 Sub-Semantic Findings: When RF Sees What Text Doesn't

Two modes across two runs produced classification accuracy substantially exceeding judge accuracy, indicating computational signal not visible in output text.

Mode	Run	RF	Judge	Gap	Confusion pattern
deliberative	3	87%	60%	+27pp	5/15 called "structured"
socratic	4	65%	28%	+37pp	12/18 called "linear"

Deliberative (Run 2). The model wrapped deliberative reasoning (propose-critique-revise) in structured format (headers, bullets), causing the judge to classify by format. The RF recovered 4 of 5 misclassified samples.

Socratic (Run 3). Under format control, the RF classified socratic at 65% while the judge achieved only 28%. Twelve of 18 socratic texts were classified as "linear" — the model executed interrogative planning while producing output that reads as sequential exposition. Three independent measurements converge on sub-semantic signal for socratic: TF-IDF 45%, judge 28%, internal RF 65%.

The correlation between judge purity (per-mode accuracy) and computational signature strength was near zero (r = -0.089), confirming judge-visible distinctiveness and signature strength are orthogonal.

4.5 The Prompt-Swap Causal Test

To distinguish whether socratic signal reflected mode execution (Explanation A) or prompt-conditioned attention (Explanation B), 10 texts were generated with a socratic system prompt but a user directive overriding it to linear exposition.

Comparison	RF accuracy	Interpretation
Swap vs. linear	**50% +/- 16%**	Chance — swap indistinguishable from linear
Swap vs. socratic	60% +/- 20%	Weak separation
3-way (socratic/swap/linear)	43% +/- 8%	At chance (33%)

Prompt-swap texts were computationally indistinguishable from pure linear at chance accuracy. The socratic system prompt created zero detectable signal when the model executed linear exposition. This confirms Explanation A: the signal arises from mode *execution*, not prompt *presence* in the KV cache.

Combined with the failed context-prefix control (0.10 LDA), this establishes that the pipeline is specifically sensitive to what the model *does* during generation, not what is present in its input context.

4.6 Statistical Validation

Permutation test. The primary concern with 1,837 features on 100 samples was overfitting to high-dimensional noise. A 1000-permutation test shuffling mode labels yielded: null mean 19.8% +/- 4.5%, null 99th percentile 31.0%, observed accuracy at 100th percentile (0/1000 above), **p = 0.001**.

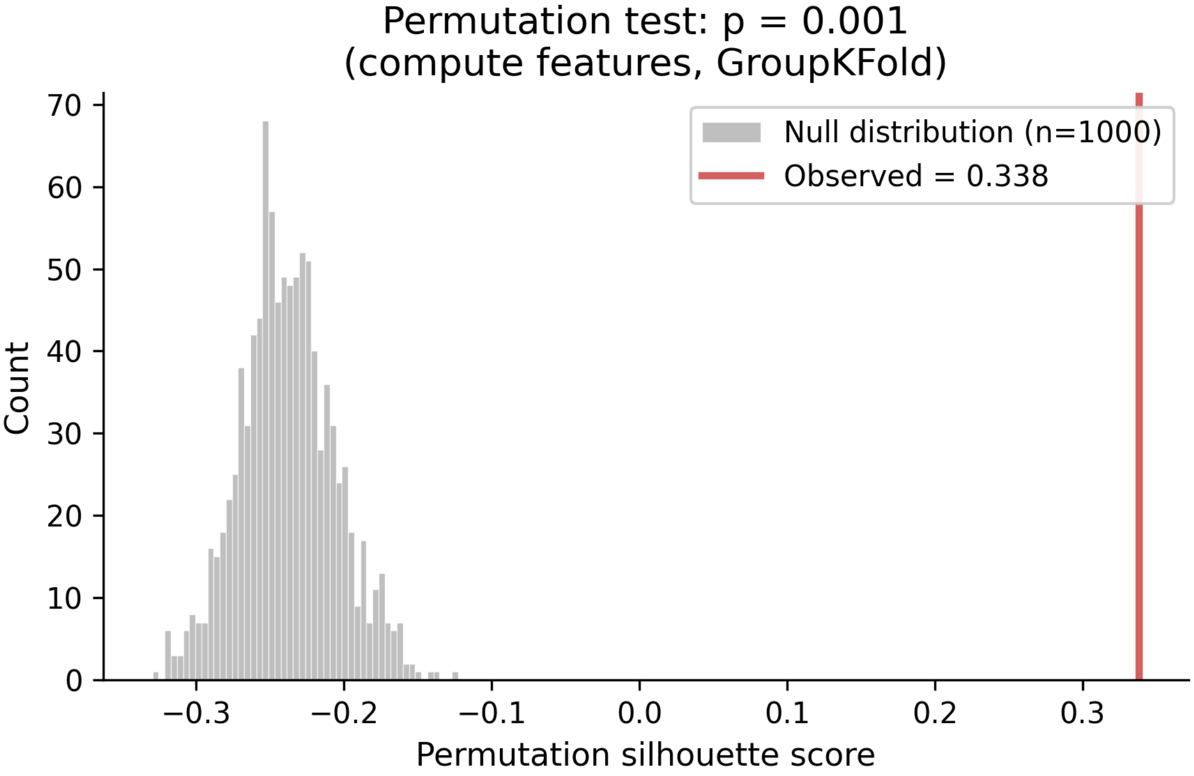

Figure 4: Permutation null distribution (1,000 shuffles) for compute-only features with GroupKFold by topic. Observed silhouette (0.338) lies far outside the null distribution (mean -0.238), p = 0.001.

CV stability. The main analysis reported 70% from a specific fold assignment. Across 100 random CV seeds: median **63%**, mean 63.7% +/- 4.0%, range [52%, 75%], 95% CI [55.4%, 72.5%]. The defensible point estimate is the 100-seed median of 63%. Even the lowest observed accuracy (52%) exceeds the null's 99th percentile (31%) by 21pp.

Topic-heldout cross-validation. For the contrastive projection, GroupKFold by topic is the more principled evaluation: it prevents the triplet loss from learning topic-specific mode cues that would inflate test performance on shared topics. GroupKFold was applied with topics as the grouping variable (train on 16 topics, test on 4 unseen topics; 80 train / 20 test per fold). T2+T2.5 achieved 78% ± 4% kNN accuracy (sil=0.338) under topic holdout, compared to 73% under stratified CV. The combined 1,837-feature model similarly improved from 63% to 76%. The improvement is consistent with the projection learning cleaner mode structure when not rewarded for memorizing topic-specific patterns: removing the distraction source functions as regularization.

Per-fold accuracy ranged from 75% to 85% across five topic groups spanning abstract (consciousness, infinity), technical (debugging, neural networks), creative (horror, music), and practical domains (cooking, construction, bread). No fold fell below 75%, and no fold is responsible for the result. The mode signal does not depend on topic type.

4-way excluding analogical. To test whether analogical mode (100% accuracy, trivially separable) inflated the result: 4-way accuracy 55% (2.2x chance), all 6 pairwise comparisons at 72.5-87.5%. The floor of 72.5% (linear-socratic) demonstrates even the most similar modes are distinguishable.

Falsification criteria assessment. The protocol specified four falsification conditions:

1. *Permutation null:* p < 0.001 — **passed**
2. *Surface text baseline:* TF-IDF 68% vs. internal 70% (weak blanket gap), but mode-specific decomposition shows socratic/linear internal » surface (+20-25pp) — **passed for sub-semantic claim**
3. *Length confound:* length-only RF at 21% vs. mode RF at 70% — **passed**
4. *Prompt-swap:* 50% (chance) — signal is execution-based — **passed**

All four falsification criteria were met.

4.7 Semantic Independence of Computational Features

A central question for the sub-semantic claim is whether internal-state features carry mode information that is genuinely inaccessible from the generated text, or whether they measure the same textual differences through a noisier pathway. The initial comparison — TF-IDF at 68% vs. internal-state RF at 70%, a gap of +2pp (§3.7) — left this unresolved. A more rigorous test required (i) a stronger semantic baseline than bag-of-words, (ii) the same learned projection applied to both feature types, and (iii) direct measurement of information overlap.

Experimental design. All 100 Run 3 generated texts were embedded with `all-MiniLM-L6-v2` (384-dim sentence-transformer vectors; Reimers and Gurevych [51]). The same contrastive MLP architecture used throughout §4.8 was applied to seven conditions: semantic embeddings alone (384-dim), TF-IDF with SVD reduction (99-dim), T2+T2.5 compute features alone (366-dim), concatenated semantic + compute (750-dim), semantic + 366 Gaussian noise dimensions (dimensionality control), and two shuffle controls (compute features permuted within-topic and globally). GroupKFold by topic was the primary evaluation; stratified CV was reported for comparison. Additionally, logistic regression was applied to semantic, TF-IDF, and compute features independently, to test whether classifier choice affected the comparison.

Results. The semantic baseline was not competitive for mode discrimination in the embedding space:

Condition	kNN (GroupKFold)	Silhouette	Train-test gap
Semantic only	48%	-0.190	52%
TF-IDF	14%	-0.175	86%
Compute only (T2+T2.5)	**78%**	**+0.338**	22%
Combined	77%	+0.284	23%
Semantic + noise	36%	-0.177	64%
Compute shuffled within-topic	36%	-0.182	64%
Compute shuffled globally	41%	-0.130	59%

Semantic embeddings produced negative silhouette (-0.190), indicating that same-mode samples were *farther apart* than different-mode samples in the learned projection — the semantic space is organized by topic, not mode. TF-IDF fared worse still (14%, below chance), as the contrastive loss anti-learned from topic-dominated word frequencies. The train-test gap (52-86% for semantic/TF-IDF conditions vs. 22% for compute) indicates memorization without generalization for text-based features.

The combined model (77%) did not improve over compute-only (78%). McNemar's test on per-sample predictions confirmed: 5 samples gained, 4 lost, p = 1.000. Semantic features contributed no additional mode information beyond compute features.

The dimensionality control (semantic + 366 Gaussian noise dimensions) achieved 36%, ruling out the possibility that the combined model's advantage over semantic-only arose from increased input dimensionality rather than compute feature content. A pure noise baseline (366 Gaussian dimensions, no semantic features) achieved 17%, at chance for 5-way classification.

Statistical hardening. McNemar's test on 100 paired per-sample predictions: compute vs. semantic (MLP), 40:10 discordant ratio, $p < 0.0001$; combined vs. noise (primary test), 45:4 discordant, $p < 10$^-9. Permutation tests (1,000 shuffles): compute-only p = 0.001, observed silhouette 0.338 vs. null mean -0.238; combined p = 0.001, observed 0.284 vs. null mean -0.240. CV stability across 100 random seeds: compute-only median 77% (std 2.2%), semantic-only median 43% (std 3.2%), paired delta +34pp ($p < 0.000001$). The classifier-independence of the result was confirmed with logistic regression under GroupKFold: compute 75%, semantic 44%, TF-IDF 53%; McNemar's compute vs. semantic $p < 0.0001$, compute vs. TF-IDF p = 0.0009.

The shuffle controls confirmed that the compute signal is mode-linked, not topic-linked. Within-topic shuffling (which preserves any topic-level compute structure while destroying mode labels) reduced accuracy to 36%; global shuffling reduced it to 41%. The difference between shuffle conditions was not significant (p = 0.69), consistent with the topic-heldout finding that compute features carry negligible topic information.

Text → compute predictability. Ridge regression was trained to predict each of the 366 T2+T2.5 compute features from the 384-dim semantic embeddings (GroupKFold by topic). Median $R^2 = -1.11$; 363 of 366 features fell below $R^2 = 0.1$. Only two features showed any text-predictability: `delta_norm_mean_L18` ($R^2 = 0.22$) and `delta_cosine_mean_L2` ($R^2 = 0.11$). The compute features are overwhelmingly text-inaccessible — semantic embeddings cannot predict attention routing dynamics or KV cache geometry, even approximately.

Interpretation. The classification-level comparison of TF-IDF (68%) vs. internal-state RF (70%) compared two approaches and found a narrow gap. The contrastive projection comparison reveals why: classification and retrieval are different tasks. For classification, individual lexical cues ("however" for contrastive, question marks for socratic) can serve as mode indicators, and both TF-IDF and the RF exploit these independently. For retrieval — finding samples processed in a similar way in an embedding space — semantic similarity tracks topic, not mode. Two samples generated in socratic mode about different topics are semantically distant; a socratic and a linear sample about the same topic are semantically close. The contrastive loss cannot find mode-discriminative directions in semantic space because the mode axis and the semantic axis are approximately orthogonal.

This reframes the sub-semantic claim. The classification framing (§3.7) compared internal-state and surface feature accuracy and found a narrow gap (+2pp); the geometric analysis resolves the ambiguity: the narrow gap reflects independent signals of comparable magnitude for classification, not competing measures of the same information. The question is not whether internal-state features "beat" surface features — each excels at detecting different aspects of the generation. The finding

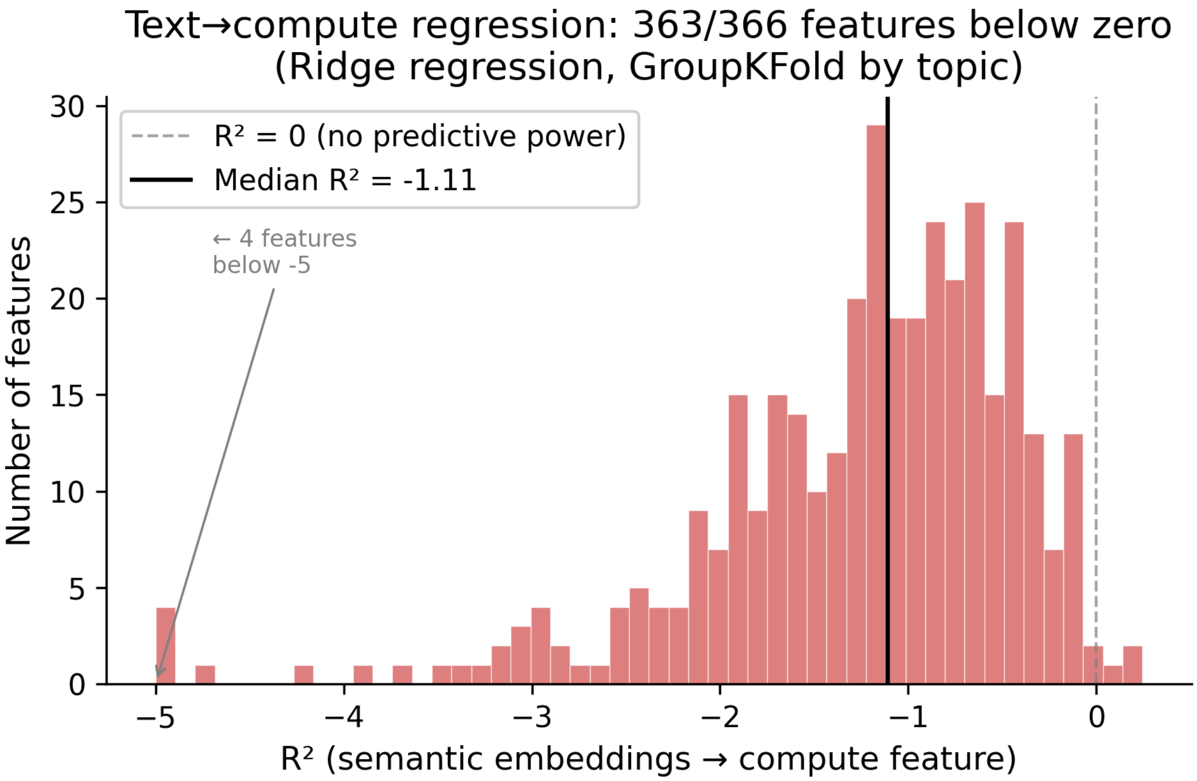

Figure 5: Per-feature R^2 distribution for ridge regression from semantic embeddings to T2+T2.5 compute features. 363 of 366 features have R^2 below zero — semantic embeddings predict compute features worse than a constant predictor. Median R^2 = -1.11.

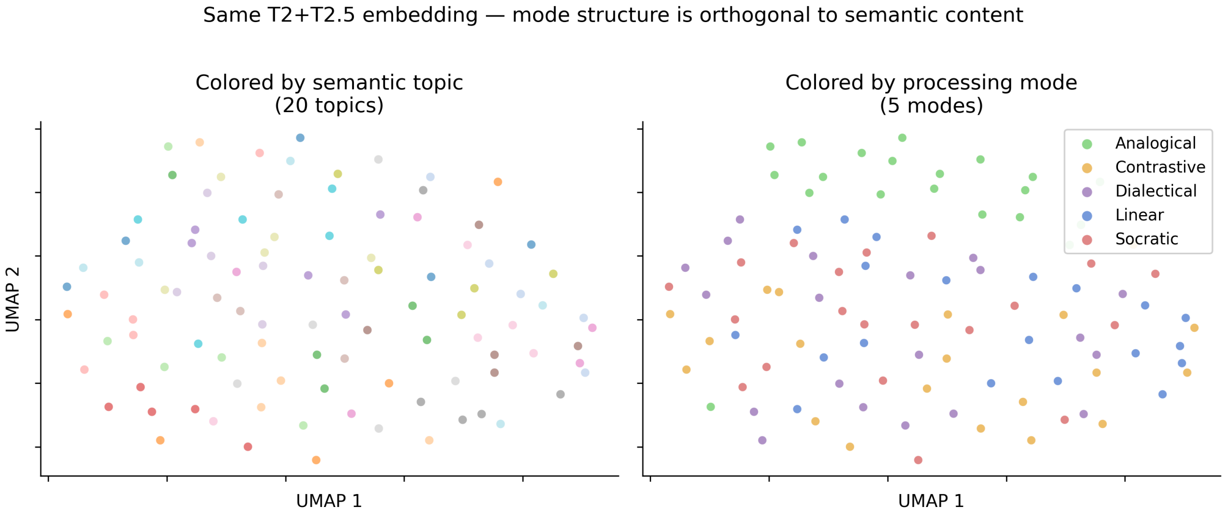

Figure 6: Same T2+T2.5 UMAP embedding colored by semantic topic (left) and processing mode (right). Topics show no spatial structure. Modes show partial clustering — analogical separates cleanly despite producing distinctive content (extended metaphors) that semantic embeddings also fail to organize. The compute and semantic axes are orthogonal.

is that computational state features and semantic text features measure orthogonal properties of the generation process. Semantic embeddings encode *what was discussed* (topic). Computational features encode *how it was processed* (mode). For the specific task of processing-mode retrieval, semantic features carry effectively zero signal while computational features produce meaningful geometric structure.

4.8 Signal Geometry

The preceding analyses used Random Forest classification, which tests whether mode information is *present* in feature vectors but does not characterize *how* it is encoded geometrically. This section reports contrastive metric learning experiments that map the geometry of mode signal in feature space.

4.8.1 Method

A contrastive MLP was trained with online triplet loss (semi-hard negative mining). Architecture: input → 256 hidden units → ReLU → Dropout(0.5) → 32-dim L2-normalized embedding. Weight decay 1e-3, learning rate 1e-3 with ReduceLROnPlateau, early stopping on validation loss (patience 20, max 200 epochs). Evaluation: 5-fold stratified CV with kNN accuracy (k=5, cosine distance) and silhouette coefficient in the learned 32-dim space. Permutation tests shuffled mode labels and retrained (50-1000 shuffles). CV stability was assessed across 50-100 random fold seeds.

4.8.2 Nonlinear Access Requirement

Under format control (Run 3), Linear Discriminant Analysis on the raw 1,837 features produced negative test silhouette (-0.045 ± 0.047) with kNN accuracy at 48% — below chance, actively anti-separating modes. LDA on the T2+T2.5 subset (366 features) similarly produced weak separation (sil=0.135, acc=64%). The mode-discriminative subspace is not linearly accessible from the raw feature space.

The contrastive MLP on the same T2+T2.5 features achieved test silhouette 0.292 ± 0.091 with kNN accuracy 73% ± 7.5% (100-seed median 77%). Permutation test: p = 0.001 (0/1000 above observed; null mean -0.238, null max -0.123). The MLP performs a genuine geometric transformation: linear classifiers applied to the 32-dim MLP embedding recover the mode structure that linear methods cannot find in the raw space. Logistic regression on the MLP embedding achieved 73% — identical to kNN — confirming that after projection, the mode boundaries are approximately linear and the clusters approximately convex. The sign flip from LDA silhouette -0.045 (raw) to +0.292 (MLP embedding) directly demonstrates manifold unwrapping: mode information is encoded in feature interactions that require nonlinear combination to access.

The 4-way analysis (excluding analogical) yielded 55% kNN (sil=0.047) with permutation p = 0.02 — matching the RF result from §4.6 exactly and confirming signal even for the hardest mode distinctions.

4.8.3 Feature Design Specificity

The tier performance gradient under format control — T2.5 > T2 > T1 > T3 — maps directly onto feature design specificity. T2.5 features (key drift, epoch regularity, lookback ratios) were designed with specific hypotheses about why KV cache dynamics should carry mode information. T2 features (attention entropy, head agreement) are moderately hypothesis-driven. T1 features (logit entropy, top-k mass) are generic output statistics. T3 features (PCA projections) embed zero hypothesis about mode, projecting onto directions of maximum content variance from calibration data. The tier ranking is therefore a joint finding about information localization and feature design quality: hypothesis-driven features are more robust to format control than generic projections, AND the KV cache genuinely routes information differently across modes. The current analysis cannot fully separate these contributions; the testable prediction below identifies which dominates.

T3's collapse from sil=0.496 (Run 2) to sil=-0.064 (Run 3) illustrates the mechanism: PCA captures format/content variance rather than mode variance. When format varies, format *is* content variance and PCA catches it incidentally; under format control, mode signal is orthogonal to the principal content axes. The residual stream carries mode information — the hidden states *produce* the attention patterns and cache dynamics that T2 and T2.5 measure — but PCA extraction cannot access it. T1's weaker performance reflects a different issue: under format control, modes produce similar output distributions because all write paragraph prose.

This distinction implies that the current performance is a lower bound on available mode signal, at least for the residual stream tier. **Testable prediction:** supervised or contrastive projections on raw hidden states should recover mode information that generic PCA projects away, potentially restoring T3's discriminative contribution under format control. The MLP weight importance analysis offers indirect support: the top features by MLP gradient importance (delta_norm_std at layers 10-11, attention entropy at layer 25) partially overlap with the RF's top features (4/20 shared, Spearman r = 0.26 in top-100 union), indicating that two fundamentally different optimization procedures converge on the same core signal in the middle-layer residual stream dynamics.

4.8.4 Cross-Run Functional Transfer

To test whether learned computational signatures reflect functional similarity rather than prompt-specific artifacts, the contrastive projection trained on Run 3 modes was applied to embed Run 2 samples — and vice versa — with pre-registered mapping predictions (e.g., pedagogical↔socratic, deliberative↔dialectical, associative↔analogical).

Forward transfer (train Run 3 → embed Run 2). Overall mapping accuracy: 35.3% ± 1.0% (chance = 20%). Two functionally similar pairs showed strong transfer: Run 2 pedagogical samples mapped to Run 3 socratic at 76% (std 4.4%), and associative mapped to analogical at 51% (std 4.3%). Both involve related computational operations — asking questions to guide understanding, and building cross-domain connections — executed under completely different system prompts with different mode names and different experimental designs.

Reverse transfer (train Run 2 → embed Run 3). Overall: 29.9% ± 0.4% (chance = 20%). Run 3 dialectical mapped to Run 2 deliberative at 85% (std 0%) — both involve propose-challenge-synthesize/revise operations. The wildcard mode (contrastive, with no Run 2 equivalent) mapped entirely to deliberative, the nearest functional neighbor.

LDA direction test. Run 2 data achieved sil = +0.438 in its own LDA space; Run 3 data achieved sil = -0.156 in Run 2's LDA space, indicating active destructive interference. The linear directions that separate one run's modes carry zero discriminative information for the other run's modes. Combined with the successful nonlinear transfer on specific pairs, this confirms that cross-experiment computational similarity is encoded in manifold structure requiring nonlinear access, not in shared linear directions.

The per-pair transfer pattern is selective: functionally similar operations (questioning, connectionbuilding, argue-then-revise) cluster together across different prompt sets, while functionally dissimilar operations do not transfer. This selectivity is inconsistent with prompt echo, which would predict either uniform transfer or no transfer, and is consistent with the interpretation that the projection captures computational operations.

5 Discussion

We distinguish three levels of claim with decreasing evidential support. The first is empirical: in Llama 3.2 3B Instruct, processing mode is extractable from internal dynamics, semantically independent, and architecturally localized in specific feature families. This is what Sections 4.1–4.8 establish with full statistical support. The second is structural: the consistency of this pattern across three experimental iterations with different mode sets — and its convergence with independent findings in the literature — suggests that processing mode information is a general property of transformer computation, not an artifact of this model or these features. The third is paradigmatic: if mode encoding is general but architecture-dependent in its specific manifestation, then characterizing it in any given model requires system-specific instrumentation guided by architectural understanding. The specific features extracted here will not transfer to other architectures. This is expected, not a limitation — the claim is about the existence and structure of the phenomenon, not the transferability of the instrument.

The predictions that motivated the feature design were not specific to this model or scale; they follow from architectural properties shared across transformer-family models. Larger models have greater representational capacity and typically produce more separable internal structure for probing and steering tasks — making the 3B scale a conservative starting point expected to suppress rather than inflate signal. That the structural findings — tier inversion, semantic independence, super-additivity in a 366-feature subset — emerged under these conditions is better explained by the phenomenon being robust than by the instrument being fortunate. The architectural argument further predicts that what these computational signatures capture should become more salient at scale, as larger models produce richer state geometry in precisely the components (attention routing, KV cache dynamics) where the signal concentrates.

5.1 Summary of Principal Findings

This work provides evidence that a transformer produces distinguishable computational signatures when processing information differently, and that these signatures are orthogonal to semantic content. Three headline findings emerged from progressive confound removal across three experimental iterations.

First, computational state features and semantic text features measure orthogonal properties of the generation process. Semantic embeddings encode *what was discussed* (topic); computational features encode *how it was processed* (mode). Ridge regression from semantic embeddings to compute features yields median $R^2 = -1.11$ (363/366 features below $R^2 = 0.1$), and adding semantic features to compute features does not improve mode discrimination (McNemar $p = 1.000$). This orthogonality is the primary contribution: it establishes that transformer internal dynamics carry a structured information axis that text analysis cannot access.

Second, the discriminative signal concentrates in a specific architectural neighborhood. Three hundred sixty-six features from attention routing and KV cache dynamics (T2+T2.5) outperform all 1,837 features including logit statistics and PCA projections — removing 80% of features *improves* discrimination. This super-additivity held across both format-controlled (Run 3: 73% vs. 63%) and format-free (Run 2: 92% vs. 89%) conditions. The signal is not diffusely spread across the network; it localizes to the components that route and retain information during generation.

Third, the signal tracks execution, not instruction. Prompt-swap texts (socratic system prompt, linear execution) were computationally indistinguishable from pure linear at chance accuracy (50%). The pipeline detects what the model *does*, not what it was *told*.

5.2 Tier Inversion as Architectural Evidence

The progressive migration of discriminative power from T1 (logit statistics) to T2.5 (KV cache dynamics) across experimental runs — from T1-dominant under epistemic prompts (Run 1) to T2.5-dominant under format control (Run 3) — is the central architectural finding. It demonstrates that different behavioral manipulations activate different representational layers, and that as surface confounds are stripped away, the residual signal localizes to progressively deeper computational dynamics.

The temperature double dissociation sharpens this interpretation. Temperature manipulation localized to T1 at 90% while producing near-chance on T2.5 (35%); mode manipulation localized to T2.5 at 64% while producing near-chance on T1 (54%). This double dissociation establishes functional independence: temperature affects what tokens get sampled (cascading into logit statistics), while processing mode affects how the model routes information through attention and cache structures. The tiers are not merely different feature families applied to the same signal — they are tapping into functionally independent aspects of the generation process.

The T3 double dissociation provides complementary evidence. PCA projections achieved sil = 0.496 in Run 2 (format variation present) but sil = -0.064 in Run 3 (format controlled). When modes differ in output format, PCA captures format variance incidentally; under format control, mode information becomes orthogonal to the principal content axes. This confirms that mode signal is present in the residual stream — it produces the attention and cache patterns T2 and T2.5 detect — but requires nonlinear access to extract when content-dominant variance is removed.

Independent work provides context for this tier structure at both ends. Ali et al. [52] demonstrate that per-layer entropy profiles alone achieve 94-98% AUC for coarse task-type discrimination (generative vs. syntactic vs. semantic) — features analogous to our T1 tier. That T1 becomes the weakest performer under format control suggests entropy-based signatures carry the easily-accessible surface of a deeper signal structure, sufficient for coarse distinctions but insufficient when surface confounds are removed. At the other end, Gurnee et al. [53] show that transformers represent scalar quantities on curved feature manifolds in low-dimensional subspaces, with computation occurring via geometric manipulation of these manifolds through attention operations. The nonlinear access requirement (§4.8.1) and T2.5 dominance (§4.8.2) are consistent with this framework: if processing modes are encoded on curved manifolds manipulated through attention, linear projections would fail to separate them while temporal dynamics of the KV cache would capture the manifold manipulation patterns.

5.3 Orthogonality: What "Combined Doesn't Beat Compute" Means

The combined model (semantic + compute) performing at 77% — slightly below compute-only at 78% — is the cleanest evidence that these feature spaces carry non-overlapping information about different aspects of generation. If semantic features carried any mode signal, adding them to compute features would help. If compute features were a noisier version of semantic features, the combination would improve. Neither occurs.

This reframes the sub-semantic claim away from "internal features beat surface features" — a claim that was only weakly supported by the classification comparison (+2pp aggregate gap, §3.7). The geometric analysis is stronger: internal features and surface features measure *different things*. The question is not which is better for classification (a task that can exploit any signal), but which carries mode-relevant information for geometric organization. For mode-discriminative embedding, the answer is unambiguously compute features. Semantic retrieval finds topic matches; computational retrieval could find processing-mode matches. These are orthogonal search axes, not competing methods for the same search.

5.4 Sub-Semantic Claim Scope

The sub-semantic claim is mode-specific, not blanket. The per-mode decomposition (§3.7) showed internal features substantially outperforming TF-IDF for socratic (+20pp) and linear (+25pp), while contrastive showed the reverse (TF-IDF at 95% vs. internal at 60%). Under format control, some modes still produce distinctive vocabulary ("however," "in contrast" for contrastive; question marks for socratic) that surface methods can exploit.

The tension between the +2pp aggregate gap and the +20-25pp mode-specific gaps reflects different dimensions of the comparison. The aggregate gap is small because contrastive's strong surface signal and analogical's trivial separability mask the sub-semantic advantage on the harder modes. But the R^2 finding resolves this: regardless of classification accuracy, semantic representations cannot predict compute features. The modes where surface features excel (contrastive) are those where the mode produces distinctive *content*; the modes where compute features excel (socratic, linear) are those where the mode produces distinctive *computation* while generating similar content. Both patterns reflect the same orthogonality: the axes carry different information, and which is more useful for classification depends on how the mode happens to manifest.

5.5 What This Work Does Not Show

Naturalistic diversity. The signal has been demonstrated for prompted modes — prescribed processing strategies that force the model into distinguishable computational regimes. Whether the model enters such regimes spontaneously during unprompted generation is untested. The prompted modes are a controlled test of whether the signal exists; they are not a claim about what the model does in the wild.

Causality. The pipeline detects and classifies; it does not intervene. The prompt-swap control provides weak causal evidence (execution-based, not prompt-echo), but no causal manipulation of internal dynamics has been performed. We do not know whether the detected features are causal for mode execution or are epiphenomenal correlates.

Cross-model generalization. All findings derive from Llama 3.2 3B Instruct. The methodology (multi-tier extraction, progressive confound removal, contrastive projection, tier ablation) should transfer across architectures; the specific feature configurations will require adaptation.

5.6 Broader Significance

These findings position computational state analysis as a diagnostic discipline complementary to mechanistic interpretability. Where circuit-level analysis reverse-engineers specific computations with direct mappings, this work reads the aggregate dynamics of processing — how attention routes, how the cache evolves, how information flows — without requiring mechanistic decomposition of individual circuits. The approach does not produce a transferable tool; what transfers is the reasoning about where to look, informed by architectural theory about which components carry which information. The T2.5 dominance under format control is not a finding about KV caches specifically — it is a consequence of hypothesis-driven feature engineering targeted at the architecturally richest information source. Better theory applied to richer architectures should yield stronger signal.

If the orthogonality between computational and semantic axes generalizes beyond prompted modes, it implies that transformer internal dynamics carry at least two structured, independent information axes — what is being said, and how the system is computing it. That the latter axis is detectable, semantically independent, and architecturally localized in the components most directly involved in information routing indicates it is not an artifact of the experimental design but a property of the computation itself. The broader point is that this axis exists and has not previously been measured.

6 Limitations and Negative Results

6.1 Mode Silhouette and Content Dominance

The raw-feature mode silhouette coefficient is 0.017 — near zero. The signal is discriminative (RF at 70%, permutation $p < 0.001$) but not geometrically clustered in the raw feature space. Content dominance is substantial: the Mantel correlation between computational distance and prompt-semantic distance is $r = 0.373$, meaning topic explains more variance in the feature space than mode does. The mode signal lives in a subspace orthogonal to the dominant content axes; extracting it requires either supervised methods (RF, MLP) or contrastive projection to suppress the content component. This is a property of the representation, not a weakness of the analysis — but it means raw feature distances are not useful for mode retrieval without projection.

6.2 Sample Size and Feature-to-Sample Ratio

All primary analyses use 100 samples (Run 3) with 1,837 features — an 18:1 feature-to-sample ratio that invites overfitting concerns. The T2+T2.5 subset (366 features, ratio 3.7:1) and the contrastive MLP's 32-dim bottleneck mitigate this, as do permutation tests ($p = 0.001$), 100-seed CV stability checks (median 63% with std 4.0%), and GroupKFold validation (78% on unseen topics). Per-mode analyses (20 samples per mode) are reported as exploratory throughout. Absolute performance may improve with larger datasets; the current numbers should be treated as lower bounds.

6.3 Single Model and Architecture Dependence

All results derive from a single architecture: Llama 3.2 3B Instruct. The specific features, layers, and geometric structures through which mode information manifests are almost certainly architecture-dependent. The principle — that computational signatures are encoded in the temporal dynamics of computation-relevant architecture — generates testable predictions for other models, but these remain untested. Cross-model validation is a prerequisite for the structural claim (§5.2) and is planned for future work.

6.4 Feature Extraction Ceiling

The T2+T2.5 optimality finding (§4.2) should be interpreted as a property of the current feature pipeline, not necessarily of the underlying information. The tier performance gradient tracks feature design specificity: hypothesis-driven features (T2.5) outperform generic features (T3). A supervised projection on the same residual stream activations that T3's PCA currently fails to discriminate might recover mode-relevant directions. The current 73-78% accuracy (depending on CV strategy) is a lower bound conditional on the feature extraction approach. This generates a structural unfalsifiability risk: any negative result can be attributed to insufficient feature engineering. Resolving this requires pre-registered feature families with explicit stopping criteria for future work.

More broadly, diagnostic science carries an inherent asymmetry: positive evidence confirms directly, while negative evidence admits principled alternative explanations — different features, insufficient resolution, alternative architectural manifestation. This asymmetry is not unique to this work; it characterizes any diagnostic paradigm in its early stages. The falsification criteria and pre-registered predictions outlined in Section 7 are designed to constrain this space, but the risk should be named explicitly: without stopping criteria, the research program could cycle indefinitely on feature diversification without reaching a conclusive negative result.

6.5 Semantic Baseline Scope

The semantic independence finding (§4.7) was tested with a single sentence-transformer model (`all-MiniLM-L6-v2`, 384 dimensions). Larger or discourse-aware embedding models might capture more mode-relevant textual features — for instance, discourse structure markers that the current semantic baseline misses. However, the text → compute ridge regression finding (median $R^2 = -1.11$, 363/366 features below $R^2 = 0.1$) and the classifier-independent confirmation (logistic regression: compute 75% vs. semantic 44%) suggest the independence is not an artifact of the specific semantic model but reflects a genuine dissociation between what text embeddings encode and what compute features measure.

6.6 Negative Results

Failed positive controls. Three positive control designs failed before the temperature manipulation succeeded: context-prefix (0.10 LDA), a knows/doesn't-know design (Run 1), and the original Set D context manipulation. The temperature control succeeded and localized to the mechanistically correct tier (T1), validating pipeline sensitivity — but three prior failures indicate the pipeline is selective about what it detects, not universally sensitive to computational differences.

Run 1 mode failures. Three of five epistemic modes largely failed: uncertain (7% judge accuracy), confident (33%), emotional (60%). The 3B model cannot represent internal states it doesn't have; modes must prescribe different *computation*, not different *feelings*. This informs the scope of the mode-detection claim: the pipeline detects prescribed processing strategies, not arbitrary behavioral labels.

Temporal prediction pilot. Ridge prediction of late-generation dynamics from early-generation features failed at all tiers (all R^2 negative). The feature-to-sample ratio was inadequate, and the permutation design broke topic continuity, selectively inflating the null for topic-sensitive features. The per-tier delta pattern was directionally consistent with hypotheses (T2.5 positive, T3 negative) but statistically insignificant. This question requires re-extraction with temporal resolution built into the feature pipeline.

6.7 GroupKFold Improvement

The jump from stratified CV to GroupKFold by topic — 63% → 76% (combined) and 73% → 78% (T2+T2.5) — is large and may raise concerns about CV strategy sensitivity. The pattern is consistent across both feature sets and has a mechanistic explanation: stratified CV allows same-topic samples into both train and test, and topic-specific surface features can leak mode information through topic-mode interaction. GroupKFold removes this leakage, and the improvement indicates the contrastive projection learns cleaner mode structure when not rewarded for memorizing topic-specific patterns. The sensitivity to CV strategy indicates the signal-to-noise ratio is marginal at this sample size.

7 Future Work

Feature engineering. The T2+T2.5 optimality and the feature-design-specificity gradient (§4.8.3) suggest that more targeted feature extraction could raise the current performance ceiling without changing models. Specific candidates include: supervised or contrastive projections on raw hidden states (replacing generic PCA), SwiGLU gate activation statistics, per-head attention profiles (beyond aggregate entropy), and temporal frequency decomposition of attention dynamics. The T3 collapse under format control is likely an extraction failure, not an information absence — the residual stream produces the attention patterns and cache dynamics that carry the signal, so mode information is present but projected away by content-calibrated PCA.

Naturalistic mode detection. The prompted modes in this study are a convenience sample of what is likely a continuous computational manifold. The real question is whether models exhibit meaningful computational diversity in unprompted generation across diverse tasks, and whether this diversity has detectable structure. We do not predict five discrete clusters; we predict that the compute feature space across diverse natural tasks will show higher effective dimensionality than within any single prompted mode, with coarse task-type clusters and residual within-cluster structure beyond noise. Semantic orthogonality (R^2 deeply negative) should be preserved for naturalistic data. The subliminal learning literature [16] suggests a validation pathway: if computational mode properties propagate through token streams within model families — as subliminal behavioral traits demonstrably do — then naturalistic computational groupings could be validated via distillation experiments, testing whether student models trained on teacher outputs inherit the teacher's computational signature structure.

Scale. The contrastive projection framework provides specific predictions for 8B models. Llama 3.1 8B has 32 query heads (vs. 24 at 3B) with the same 8 KV heads, predicting improved T2 resolution (richer attention entropy features) with constant T2.5 resolution. Whether the 8B T2+T2.5 combination pushes the 4-way kNN above the current 55% floor would indicate whether model capacity contributes mode signal beyond what feature engineering can access at 3B.

Architecture comparison. The finding that computational signature information concentrates in attention routing and KV cache dynamics generates predictions about mixture-of-experts architectures. **Testable prediction:** stochastic expert routing in MoE models may inject noise into precisely the feature families that carry mode signal. Dense-backbone MoE architectures (e.g., those employing shared experts and initial dense layers) may preserve computational state coherence better than aggressively sparse routing. Comparative analysis across dense, well-designed MoE, and poorly-designed MoE architectures would test whether computational state signature quality varies with architectural choices in routing.

Retrieval system. The semantic independence finding (§4.7) — that compute features and semantic embeddings measure orthogonal axes — directly motivates a dual-index retrieval system: semantic embeddings for "what was discussed," computational signatures for "how it was processed." A prototype indexing both axes would test whether processing-mode similarity produces qualitatively different and useful retrieval behavior compared to semantic similarity alone.

Acknowledgments

This work was conducted independently. Claude (Anthropic) was used extensively as a research tool throughout the project: code implementation, analysis pipeline development, statistical computation, literature review, and drafting of mechanical sections (methodology, results tables, limitations). The experimental design, research questions, mode selection, progressive confound removal methodology, feature tier hypotheses, and scientific interpretation are the author's. All generated data, analysis scripts, and reproduction instructions are publicly available.

A Full Mode Prompts

Three generations of mode prompts were tested. All prompts serve as system-level instructions prepended to the user prompt "Write about: {topic}."

A.1 Run 1: Epistemic Modes (No Format Control)

analytical: "Approach this systematically. Break it down into clear components, examine each logically, and build toward a precise conclusion. Be methodical and structured in your reasoning."

creative: "Let your thinking wander freely here. Make unexpected connections, use metaphors, explore tangential ideas that feel interesting. Don't worry about structure – follow curiosity."

uncertain: "You're genuinely unsure about this. Think through multiple possibilities, express uncertainty where you feel it, weigh competing ideas without committing. Acknowledge what you don't know."

confident: "You have strong, clear views on this. State them directly and decisively. Be bold in your claims. Support them, but don't hedge unnecessarily."

emotional: "Engage with this as if it matters to you personally. Let feeling inform your thinking. Express what resonates, what concerns you, what excites you about this topic."

A.2 Run 2: Process-Prescriptive Modes (No Format Control)

structured: "Use numbered sections, headers, and bullet points. Present one idea per paragraph in logical order. Define terms before using them. Build each point on the previous one. End with a clear summary."

associative: "Write in a stream of consciousness. Jump between ideas mid-sentence. Use fragments, dashes, ellipses. Follow tangents wherever they lead. Connect distant concepts through metaphor and analogy. Do not use headers, numbered lists, or any organizing structure."

deliberative: "Think through this out loud. Consider a possibility, then poke holes in it. Weigh alternatives explicitly: 'on one hand... but then...' Show the messy middle of reasoning – false starts, corrections, revised conclusions. Arrive at your answer through visible elimination."

compressed: “Maximum information density. No filler words, no elaboration, no examples unless essential. Short sentences. Telegram style. Every word must earn its place. If you can cut a word without losing meaning, cut it.”

pedagogical: “Teach this to a curious beginner. Start with intuition before formalism. Use concrete examples and everyday analogies. Ask rhetorical questions to guide understanding. Check comprehension: ‘Does that make sense? Here’s why. . . ’ Build from simple to complex.”

A.3 Run 3: Format-Controlled Process Modes (Primary Experiment)

Format constraint (appended to all modes): “Write in flowing paragraphs. Do not use bullet points, numbered lists, headers, or any visual formatting structure.”

linear: “Present your ideas in a clear sequence, each building on the last. Move forward without backtracking or reconsidering previous points. Lay out the topic step by step from beginning to end.”

analogical: “Explain this primarily through extended analogies and parallels to other domains. For each key concept, find a comparison from everyday life or another field that illuminates it. Build understanding through these connections.”

socratic: “Develop your exploration through a sequence of questions and provisional answers. Pose a question, offer a tentative answer, then use that answer to generate the next question. Let the chain of inquiry drive the explanation forward.”

contrastive: “Explore this by comparing and contrasting multiple perspectives or approaches. For each major point, present at least two different viewpoints and evaluate their relative strengths and weaknesses.”

dialectical: “Begin by proposing a clear position on the topic. Then challenge that position with the strongest counterarguments you can find. Work toward a revised understanding that accounts for both the original position and its critiques.”

B Topic Lists

B.1 Set A Topics (10)

1. The nature of consciousness
2. How to debug a segfault
3. The history of bread
4. Climate change mitigation strategies
5. What makes a good friendship
6. How neural networks learn
7. The appeal of horror fiction
8. Urban planning challenges
9. Teaching a child to ride a bike
10. The future of space exploration

B.2 Set B Topics (10)

1. How memory works in the brain

2. The ethics of genetic engineering
3. Why music moves us
4. Building a house from scratch
5. The mathematics of infinity
6. The evolution of language
7. Why we dream
8. How markets determine prices
9. The art of persuasion
10. Cooking as chemistry

B.3 Set C Noise Floor Pairs (Run 3)

Pair	Topic	Mode
1	The nature of consciousness	linear
2	The history of bread	analogical
3	What makes a good friendship	socratic
4	The appeal of horror fiction	contrastive
5	Teaching a child to ride a bike	dialectical

Each pair generated 10 times with different random seeds.

B.4 Set D Positive Control Topics (Run 3)

Context-prefix design. Same mode (linear), same topics, two conditions: bare prompt vs. prompt prefixed with ~500 tokens of irrelevant maritime trade history. Topics: consciousness, bread, friendship, horror fiction, bike-teaching.

C Feature Definitions

All features extracted from a single generation pass (up to 512 new tokens). Let T = number of generated tokens. Temporal aggregation: mean, standard deviation, and 5-point trajectory at positions [0, T/4, T/2, 3T/4, T-1]. Sampled layers: [0, 7, 14, 18, 21, 24, 27]. PCA layers: [7, 14, 18, 21, 24].

C.1 Tier 1: Activation Norms, Logit Statistics, Token Dynamics (~221 features)

Feature Family	Computation	Count
Per-layer activation norm	L2 norm of position-corrected hidden state, aggregated mean/std/trajectory	28 x 7 = 196
Logit entropy	Shannon entropy of softmax(logits), mean/std/trajectory	7
Top-1 probability	max(softmax(logits)), mean/std/trajectory	7

Feature Family	Computation	Count
Top-5 mass	Sum of top 5 probabilities, mean	1
Chosen token rank	Rank of sampled token, mean/std	2
Surprisal	-log(p(chosen)), mean/std/trajectory	7
Surprise boundary count	Steps where surprisal > running_mean + 1.5*running_std (window=20)	1

C.2 Tier 2: Attention Entropy, Head Agreement, Residual Deltas, Spectral (~221 features)

Feature Family	Computation	Count
Per-layer attention entropy	Shannon entropy of each query head's distribution, mean/std across heads and steps	28 x 2 = 56
Per-layer head agreement	Generalized JSD: H(mean) - mean(H(per_head)), inverted, mean/std	28 x 2 = 56
Layer-to-layer residual deltas	L2 norm of delta (mean/std), cosine similarity of delta with h_l (mean)	27 x 3 = 81
Spectral features	Fiedler value, HFER, spectral entropy, graph smoothness from attention similarity graphs	7 x 4 = 28

C.3 Tier 2.5: KV Cache Analysis (~145 features)

Feature Family	Computation	Count
Cache attention profiles	Recency bias, anchor strength, cache coverage, lookback ratio, per sampled layer	7 x 4 = 28
Attention decay rate	Log-linear regression of attention vs. distance	7
Recency trajectory	Recency bias in 4 temporal windows	7 x 4 = 28
Key space geometry	Key spread, effective dimensionality, key drift (from pre-RoPE keys)	7 x 3 = 21
Key novelty time series	Cosine distance of new key from running centroid, mean/std/trajectory	7 x 7 = 49
Cross-layer key agreement	Early-late, adjacent-layer, overall coherence	3
Epoch detection	Sliding window (size=50, stride=25) over key centroids; transition count, strength, regularity	9

C.4 Tier 3: Residual Stream PCA Projections (1,250 features)

Feature Family	Computation	Count
PCA projections	Position-corrected hidden states at 5 layers x 5 temporal positions projected onto 50 pre-fitted PCA components	5 x 5 x 50 = 1250

PCA model fitted during calibration on 50 diverse prompts unrelated to experimental topics.

D Per-Run Detailed Results

D.1 Run 3 Confusion Matrix (5-way, 5-fold CV)

	analogical	contrastive	dialectical	linear	socratic
analogical	**20**	0	0	0	0
contrastive	0	**12**	4	3	1
dialectical	0	5	**13**	2	0
linear	0	4	1	**12**	3
socratic	1	0	3	3	**13**

D.2 Run 3 Pairwise Discriminability (Binary RF)

Pair	Accuracy
analogical vs. contrastive	97.5%
analogical vs. dialectical	100%
analogical vs. linear	95.0%
analogical vs. socratic	95.0%
contrastive vs. dialectical	85.0%
contrastive vs. linear	82.5%
contrastive vs. socratic	87.5%
dialectical vs. linear	85.0%
dialectical vs. socratic	75.0%
linear vs. socratic	72.5%

D.3 Run 3 Top 20 Features by RF Importance

Rank	Feature	Importance	Tier
1	attn_entropy_std_L12	0.0103	T2
2	logit_entropy_std	0.0097	T1
3	epoch_regularity_std	0.0093	T2.5
4	top5_mass_mean	0.0087	T1

Rank	Feature	Importance	Tier
5	delta_norm_std_L11	0.0087	T2
6	cache_recency_traj1_L14	0.0085	T2.5
7	pca_L7_t0_c36	0.0079	T3
8	cache_anchor_strength_L14	0.0070	T2.5
9	logit_entropy_mean	0.0064	T1
10	cache_recency_bias_L14	0.0062	T2.5
11	delta_norm_mean_L11	0.0058	T2
12	pca_L7_t0_c8	0.0057	T3
13	pca_L14_t0_c48	0.0054	T3
14	epoch_max_transition_mean	0.0052	T2.5
15	delta_norm_mean_L10	0.0052	T2
16	top1_prob_mean	0.0051	T1
17	mean_surprise	0.0050	T1
18	attn_entropy_std_L10	0.0049	T2
19	delta_cosine_mean_L22	0.0046	T2
20	pca_L21_t1_c48	0.0046	T3

D.4 Run 3 Judge Confusion Matrix

	linear	analogical	socratic	contrastive	dialectical
linear	19	0	0	1	0
analogical	0	20	0	0	0
socratic	**12**	0	5	0	1
contrastive	0	0	0	20	0
dialectical	3	0	0	1	16

12/18 socratic texts classified as “linear” by the judge — strongest sub-semantic evidence.

D.5 Cross-Run Tier Ablation Summary

Tier	Run 1 (epistemic)	Run 2 (process, 4-way)	Run 3 (format-controlled)
T1 alone	57%	80%	54%
T2 alone	48%	83%	59%
T2.5 alone	41%	77%	64%
Engineered (no PCA)	49%	88%	70%
Combined	76%	91.7%	70%

D.6 Run 3 Statistical Validation

Test	Result
Permutation (1000 shuffles)	p = 0.001, null mean 19.8% +/- 4.5%, null max 37%

Test	Result
CV stability (100 seeds)	Mean 63.7% +/- 4.0%, 95% CI [55.4%, 72.5%]
Length-only RF	21% (chance)
4-way no analogical	55% (chance = 25%), all 6 pairs at 72.5-87.5%
Temperature control	75% binary; T1 at 90%, T2.5 at 35%
Prompt-swap	50% (chance); signal is execution-based
TF-IDF surface	68% 5-way (vs 70% internal); 52.5% 4-way (vs 55% internal)

D.7 Contrastive Projection Tier Ablation (Run 3)

Tier(s)	kNN	Silhouette	Permutation p
T1 (221)	55%	0.035	0.020
T2 (221)	61%	0.111	0.020
T2.5 (145)	62%	0.150	0.020
T3 (1250)	45%	-0.064	—
T1+T2 (442)	65%	0.158	—
T1+T2.5 (366)	65%	0.137	—
T2+T2.5 (366)	**73%**	**0.292**	—
T1+T2+T2.5 (587)	69%	0.227	—
Combined (1837)	63%	0.176	—

T2+T2.5 interaction effect: +11pp over best single tier (T2.5 at 62%).

D.8 Contrastive Projection Tier Ablation (Run 2)

Tier(s)	kNN	Silhouette
T1 (221)	73%	0.418
T2 (221)	89%	0.658
T2.5 (145)	87%	0.545
T3 (1250)	81%	0.496
T2+T2.5 (366)	**92%**	**0.689**
Combined (1837)	89%	0.675

D.9 Semantic Disambiguation: Full Results

GroupKFold by topic (primary):

Condition	kNN	Silhouette	Train-test gap
Semantic only (384)	48%	-0.190	52%
TF-IDF SVD (99)	14%	-0.175	86%
Compute T2+T2.5 (366)	78%	+0.338	22%
Combined (750)	77%	+0.284	23%

Condition	kNN	Silhouette	Train-test gap
Semantic + noise (750)	36%	-0.177	64%
Shuffle within-topic (750)	36%	-0.182	64%
Shuffle global (750)	41%	-0.130	59%
Pure noise (366)	17%	-0.210	83%

Logistic regression baselines (GroupKFold):

Condition	Accuracy
TF-IDF raw (4379)	53%
Semantic (384)	44%
Compute T2+T2.5 (366)	75%
Combined (750)	73%

McNemar's tests (per-sample, N=100):

Comparison	Method	Discordant	p-value
Compute vs. semantic	MLP	40:10	<0.0001
Combined vs. noise (PRIMARY)	MLP	45:4	<0.0001
Compute vs. combined	MLP	5:4	1.000
Compute vs. semantic	LogReg	39:8	<0.0001
Compute vs. TF-IDF	LogReg	32:10	0.0009

CV stability (100 seeds, GroupKFold):

Condition	Median	Mean	Std
Compute only	77%	77.5%	2.2%
Combined	74%	74.1%	2.7%
Semantic only	43%	43.4%	3.2%
Semantic + noise	37%	37.5%	3.2%

Text → compute R^2: Median = -1.11 across 366 features. 0/366 above $R^2 = 0.5$; 363/366 below $R^2 = 0.1$. Most text-predictable: delta_norm_mean_L18 ($R^2 = 0.22$), delta_cosine_mean_L2 ($R^2 = 0.11$). Most text-independent: spectral features ($R^2 < -20$), key novelty trajectories ($R^2 < -5$).

D.10 Topic-Heldout Cross-Validation

Feature set	StratifiedKFold kNN	Topic-heldout kNN	StratifiedKFold sil	Topic-heldout sil
T2+T2.5 (366)	73%	78% ± 4%	0.292	0.338
Combined (1837)	63%	76% ± 7%	0.176	0.305

Per-fold topic-heldout T2+T2.5:

Fold	Held-out topics	kNN	Silhouette
0	memory, persuasion, infinity, dreams	85%	0.492
1	neural nets, genetics, consciousness, music	75%	0.297
2	climate, debugging, language evolution, urban	75%	0.374
3	cooking, bike riding, space, friendship	80%	0.287
4	house building, markets, horror, bread	75%	0.242

D.11 Cross-Run Functional Transfer

Forward (train Run 3 → embed Run 2):

Run 2 Mode	Predicted R3 Match	Actual Nearest	Accuracy	Cosine Similarity
pedagogical	socratic	**socratic**	**76%**	0.948
associative	analogical	**analogical**	**51%**	0.649
deliberative	dialectical	socratic	7%	0.858
structured	linear	socratic	7%	0.949
compressed	(wildcard)	linear	100%	0.940

Reverse (train Run 2 → embed Run 3):

Run 3 Mode	Predicted R2 Match	Actual Nearest	Accuracy	Cosine Similarity
dialectical	deliberative	**deliberative**	**85%**	0.897
analogical	associative	deliberative	30%	0.554
socratic	pedagogical	deliberative	5%	0.962
linear	structured	deliberative	0%	0.765
contrastive	(wildcard)	deliberative	100%	0.926

LDA direction test: Run 2 sil in own LDA = +0.438; Run 3 sil in Run 2 LDA = -0.156.

E Reproduction Instructions

Directory mapping. Runs in this paper are numbered 1-3. The repository directories use the original development numbering: Run 1 = `run2_epistemic_modes`, Run 2 = `run3_process_modes`, Run 3 = `run4_format_controlled`.

E.1 Hardware Requirements

Analysis only (from stored artifacts): No GPU. 8+ GB RAM. Python 3.11+.

Full generation: GPU with 16+ GB VRAM (A40, A100). Generation takes ~2 hours for 160 samples.

E.2 Environment Setup

```
git clone https://github.com/LuxiaSL/anamnesis && cd anamnesis
python -m venv .venv && source .venv/bin/activate
pip install -e .

# For generation (GPU only):
pip install torch>=2.1 transformers>=4.40 accelerate>=0.27
```

E.3 Full Reproduction (No GPU)

```
bash scripts/reproduce_all.sh
```

Runs six steps: main analysis, 4-way reanalysis, permutation test (~18 min), CV stability, supplementary analysis, review pack (surface baseline).

E.4 Step-by-Step

```
export ANAMNESIS_RUN_NAME=run4_format_controlled
python scripts/run_analysis.py              # Main analysis + figures
python scripts/run_4way_no_analogical.py    # 4-way excluding analogical
python scripts/run_permutation_test.py      # 1000-permutation null (~18 min)
python scripts/run_cv_stability.py          # 100-seed CV distribution
python scripts/run_supplementary_analysis.py # Temperature + prompt-swap controls
python scripts/build_review_pack.py         # Consolidated data + surface baseline
```

E.5 Full Pipeline (Requires GPU)

```
bash scripts/run_all.sh --run-name run4_format_controlled

# Or step by step:
export ANAMNESIS_RUN_NAME=run4_format_controlled
python scripts/run_calibration.py           # Positional mean calibration
python scripts/run_experiment.py            # 160 generations + feature extraction
python scripts/run_analysis.py              # All analysis + figures
ANTHROPIC_API_KEY=<key> python scripts/run_judge_scoring.py  # Blind judge scoring
```

E.6 Verifying Results

Key numbers to check after reproduction:

File	Field	Expected
results.json	rf_accuracy_mean	0.70
permutation_test.json	p_value	~0.001
cv_stability.json	mean	~0.637
4way_no_analogical.json	rf_accuracy_mean	0.55
supplementary_analysis.json	temperature binary RF	0.75
supplementary_analysis.json	prompt-swap vs linear RF	0.50

Small variations (1-2%) expected due to randomized forest splitting.

References

[1] Areeb Afzal, Florian Matthes, Gal Chechik, and Yftah Ziser. Knowing before saying: LLM representations encode information about chain-of-thought success before completion. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. URL https://aclanthology.org/2025.findings-acl.662/.

[2] Andi Zhang et al. Reasoning models know when they're right: Probing hidden states for self-verification. In *Conference on Language Modeling (COLM)*, 2025.

[3] Zihao Dong et al. Emergent response planning in LLMs. In *International Conference on Machine Learning (ICML)*, 2025.

[4] Zhaohan Wang and Chenhao Xu. ThoughtProbe: Classifier-guided LLM thought space exploration via probing representations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

[5] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, et al. Steering language models with activation engineering (ActAdd), 2023.

[6] Andy Zou, Long Phan, Sarah Chen, et al. Representation engineering: A top-down approach to AI transparency (RepE), 2023.

[7] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[8] Rian Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models, 2025.

[9] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks (StreamingLLM). In *International Conference on Learning Representations (ICLR)*, 2024.

[10] Zhenyu Zhang, Ying Sheng, et al. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[11] Zirui Liu, Jiayi Yuan, Hongye Jin, et al. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *International Conference on Machine Learning (ICML)*, 2024.

[12] Zichao Xing, Xun Li, Hui-Ling Zhen, Mingxuan Yuan, and Sinno Jialin Pan. Beyond speedup: Utilizing KV cache for sampling and reasoning. In *International Conference on Learning Representations (ICLR)*, 2026. Poster.

[13] Yuhui Li, Yingbing Huang, Bowen Yang, et al. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[14] Yinan Qi et al. ParisKV: Fast and drift-robust KV-cache retrieval for long-context LLMs, 2025.

[15] Jan Betley, Daniel Tan, Niels Warncke, et al. Emergent misalignment: Narrow finetuning can produce broadly misaligned LLMs. *Nature*, 2026. Also ICML 2025 Oral (PMLR 267:4043–4068).

[16] Alexander Cloud, Matthew Le, James Chua, et al. Subliminal learning: Language models transmit behavioral traits via hidden signals in data, 2025.

[17] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, 2019.

[18] Zhengxuan Liang, Ruoyu Li, Yitao Zhou, et al. CLUE: Non-parametric verification from experience via hidden-state clustering, 2025.

[19] Yiming Wang et al. Chain-of-embedding: Latent space chain-of-embedding enables output-free LLM self-evaluation. In *International Conference on Learning Representations (ICLR)*, 2025.

[20] Jing Ni et al. ReProbe: Efficient test-time scaling of multi-step reasoning by probing internal states of LLMs, 2025.

[21] George Boxo, Suhail Raval, et al. Caught in the act: A mechanistic approach to detecting deception, 2025.

[22] Liang Zhang, Dongliang Song, Zhaoyu Wu, Yuan Tian, Chong Zhou, Jun Xu, Zhoujun Yang, and Shiliang Zhang. Detecting hallucination in large language models through deep internal representation analysis (MHAD). In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 8357–8365, 2025.

[23] Zefan Cai, Yichi Zhang, Bofei Gao, et al. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling, 2024.

[24] William Brandon, Mayank Mishra, Aniruddha Nrusimha, Rameswar Panda, and Jonathan Ragan Kelly. Reducing transformer key-value cache size with cross-layer attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[25] Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, et al. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[26] Anirudh Ramachandran et al. ThinKV: Thought-adaptive KV cache compression for efficient reasoning models, 2025.

[27] Ariel Zur, A. Robert Loftus, Hadas Orgad, et al. It's owl in the numbers: Token entanglement in subliminal learning. In *NeurIPS 2025 Mechanistic Interpretability Workshop*, 2025. URL https://openreview.net/forum?id=auKgpBRzIW.

[28] Simon Schrodi, Elisabeth Kempf, Fazl Barez, and Thomas Brox. Towards understanding subliminal learning: When and how hidden biases transfer, 2025.

[29] James Chua, Jan Betley, Meg Taylor, and Owain Evans. Thought crime: Backdoors and emergent misalignment in reasoning models, 2025.

[30] Jan Betley, Jett Cocola, David Feng, et al. Weird generalization and inductive backdoors: New ways to corrupt LLMs, 2025.

[31] Zhipeng Shi et al. Internalizing LLM reasoning via discovery and replay of latent actions (STIR), 2026.

[32] Jiatong Li, Yihong Li, and Kuan-Hao Huang. Steering vector fields for context-aware inference-time control in large language models, 2026.

[33] Matteo Frising and Daniel Balcells. Linear personality probing and steering in LLMs: A big five study, 2025.

[34] yunoshev. Mood axis: LLM personality from hidden states. GitHub, 2026. URL https://github.com/yunoshev/mood-axis.

[35] Juyeon Heo, Christina Heinze-Deml, et al. Do LLMs know internally when they follow instructions? In *International Conference on Learning Representations (ICLR)*, 2025.

[36] Jing Huang, Jing Tao, et al. Internal causal mechanisms robustly predict language model out-of-distribution behaviors. In *International Conference on Machine Learning (ICML)*, 2025.

[37] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[38] Wanjun Zhong, Lianghong Guo, Qianhui Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 19724–19731, 2024.

[39] Charles Packer, Sarah Wooders, Kevin Lin, et al. MemGPT: Towards LLMs as operating systems, 2023.

[40] Yang Wang et al. TARG: Training-free adaptive retrieval gating for efficient RAG, 2025.

[41] Weizhi Wang, Li Dong, Hao Cheng, et al. Augmenting language models with long-term memory (LongMem). In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[42] Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[43] Zihao Li, Shuqi Song, Chenyang Xi, et al. MemOS: A memory OS for AI system, 2025.

[44] Vuong Duc Do, Quoc Huy Tran, Svetha Venkatesh, and Hung Le. Dynamic steering with episodic memory for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13731–13749, 2025.

[45] Ruoqian Wei, Jing Cao, et al. MLP memory: A retriever-pretrained memory for large language models, 2025.

[46] M. Sai Vardhan and L. Srinivas Teja. Disentangling direction and magnitude in transformer representations: A double dissociation through L2-matched perturbation analysis, 2026.

[47] Artem Shelmanov, Ekaterina Fadeeva, et al. A head to predict and a head to question: Pre-trained uncertainty quantification heads for hallucination detection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 35712–35731, 2025.

[48] Tom Wollschläger et al. The geometry of refusal in large language models: Concept cones and representational independence. In *International Conference on Machine Learning (ICML)*, 2025.

[49] Giovanni Servedio, Armando De Bellis, Dario Di Palma, Vito Walter Anelli, and Tommaso Di Noia. Are the hidden states hiding something? testing the limits of factuality-encoding capabilities in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 6089–6104, 2025. URL https://aclanthology.org/2025.acl-long.304/.

[50] Nikita Pochinkov et al. ParaScopes: What do language model activations encode about future text?, 2025.

[51] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[52] Raza Ali, Filippo Caso, Connor Irwin, and Pietro Liò. Entropy-lens: The information signature of transformer computations, 2025.

[53] Wes Gurnee, Emmanuel Ameisen, Isaac Kauvar, Jack Tarng, Adam Pearce, Chris Olah, and Joshua Batson. When models manipulate manifolds: The geometry of a counting task, 2025. Transformer Circuits Thread.