Probing Dimensional Escape in Frozen Language Models via Reversible KV-Cache MCTS

Anonymous ACL submission

Abstract

Inference-time tree search over a frozen Transformer's KV cache offers a gradient-free route to test-time scaling, yet its behavior under surrogate-reward guidance remains poorly understood. We introduce **Reversible KV-Cache MCTS**, an inference-time algorithm that mutates the KV cache in place using FP32 accumulators and exactly rolls back after each tree expansion. We prove zero-drift reversibility, validated to machine precision over 200 apply-revert cycles on Llama 3.2-1B/3B, and show an $O(M_{KV} + d \cdot K_{acc})$ working-set memory complexity independent of branching factor, in contrast to the $O(b^d \cdot M_{KV})$ cost of naive parallel-cache MCTS.

Using this infrastructure, we reveal a **scale-conditional failure mode** of surrogate-reward latent-space search, which we frame as reward-driven Goodharting in KV-cache geometry, or *Dimensional Escape*. On ARC-Easy across five (scale, family) cells—Llama 3.2-1B/3B, Llama-3.1-8B, Qwen3-8B/14B—the entropy-minimization reward we test, under a single random unit-vector steering direction (seed 42) and letter-prompted scoring, provides no signal beyond greedy decoding ($\Delta_{E-G} \in [-0.025, +0.000]$), via distinct mechanisms at each cell: prior-amplification at 1B, greedy-collapse at 3B, entropy-signal collapse at 8B-Llama, and chat-template format-fragility on the random arm at Qwen3-8B/14B. A latent cartography sweep across prompt classes further shows that this mechanism depends on the model's prior structure. Reversible KV-Cache MCTS thus provides an interpretability tool for studying continuous latent search in frozen Transformers, complementing monitor-driven gradient-input analyses.

1 Introduction

Inference-time interventions on frozen language models—from one-shot activation steering to iterative test-time tree search—offer a frontier for both capability scaling and mechanistic analysis. As interventions move from heuristic perturbations to formal search procedures over a model's continuous latent state, a central question emerges: when controlled perturbations to a frozen model's internal representations are guided by a low-rank scoring projection (whether monitor, reward, or probe), do the search trajectories navigate the semantic manifold, or do they exploit the unconstrained dimensions of an overparameterized latent space?

This question has been posed in alignment-faking research—both as behavioral compliance under safety-training pressure (Greenblatt et al., 2024; Hubinger et al., 2024) and as representational monitor evasion (Gupta and Jenner, 2025)—and in gradient-trained input-space attacks against frozen models (Bailey et al., 2026), the latter reporting a tentative negative result. These observations differ in intervention site, training regime, and objective but posit a single geometric pattern that we name *Dimensional Escape*: localized optimization pressure on an overparameterized latent space resolves through the unconstrained directions of that space (Sec. 3.2).

We test the *reward-driven* case, in which an inference-time tree search optimizes a surrogate reward defined on the frozen model's own internal signals. Distinguishing converged trajectories that produce semantically meaningful states from degenerate optima (reward maxima bypassing the answer-relevant subspace) requires search infrastructure with exact-rollback semantics across deep tree expansions. The Reversible KV-Cache MCTS framework introduced here provides this: Theorem 1 establishes zero-drift reversibility via an FP32 accumulator (validated to machine precision over 200 apply–revert cycles), and Proposition 1 establishes a memory bound independent of branching factor (measured at 3.04–3.06× the base KV cache across Llama 3.2-1B/3B, Llama-3.1-8B, and Qwen3-8B/14B).

Across two experiments—a controlled negative-control on ARC-Easy (Experiment 1, Sec. 5.1) and Latent Cartography sweeps across four prompt classes (Experiment 2, Sec. 5.1)—the reward-driven case manifests robustly. An entropy-minimization reward correlating with correctness at $|r| \approx 0.5$ collapses under MCTS across all five tested (scale, family) cells—Llama 3.2-1B/3B, Llama-3.1-8B, Qwen3-8B/14B—via distinct mechanisms (prior-amplification at 1B; greedy-collapse at 3B; entropy-signal collapse at 8B-Llama; chat-template format-fragility on the random arm at the Qwen3 scales). Cartography further reveals a scale-conditional, class-conditional mechanism: MCQ classes flip from heavy-steering at 1B to minimal-steering at 3B (cross-scale $\Delta_{\bar{\alpha}} \approx 0.67/0.55$ on ARC-Easy/TruthfulQA; details in §5.1). We argue this reward-driven failure is structurally analogous to the monitor-driven sub-case hypothesized in the alignment-faking literature; Reversible KV-Cache MCTS is therefore most usefully positioned as a diagnostic probe of frozen-model representational geometry rather than a reasoning-improvement tool, with the monitor-driven sub-case left to future work.

1.1 Contributions

This paper makes the following contributions:

1. **Reversible Monte Carlo Tree Search in KV-cache latent space.** We introduce an inference-time algorithm that mutates a frozen Transformer's cached key/value tensors via in-place arithmetic ($\mathbf{K}_t \leftarrow \mathbf{K}_t + \alpha \cdot \mathbf{d}_K^{(\ell)}$) and inverts each mutation exactly through an FP32 accumulator. We prove zero-drift reversibility (Theorem 1) and validate it empirically on Llama 3.2-1B/3B-Instruct (drift $= 0$ to machine precision over 200 apply–revert cycles). The working-set memory complexity is $O(M_{\text{KV}} + d \cdot K_{\text{acc}})$ (Proposition 1) rather than the $O(b^d \cdot M_{\text{KV}})$ of standard parallel-cache MCTS—reducing the memory floor from exponential in b^d to additive in d. The accumulator overhead is independent of branching factor in the bound, measured at 3.04–3.06 $\times$ M_{KV} above the baseline cache across the tested depth and prefill configurations on all five (scale, family) cells (Llama 3.2-1B/3B, Llama-3.1-8B, Qwen3-8B/14B). We position this algorithm as reusable interpretability infrastructure: a memory-bounded, exact-rollback probe for studying continuous-latent search behavior on any frozen Transformer, applicable to monitor-driven and reward-driven scoring projections beyond the entropy reward we test in Sec. 5.1.

2. **Empirical demonstration of reward-driven Dimensional Escape—an instance of Goodhart's Law in high-dimensional latent-space search.** We establish a geometric pathology of MCTS in overparameterized latent spaces: the entropy-normalized surrogate reward we test resolves localized optimization pressure by routing representations through unconstrained dimensions, succumbing to prior-amplification or greedy-collapse rather than satisfying the semantic objective; we conjecture this extends to surrogate rewards lacking exogenous grounding more generally (Sec. 3.2). Two complementary experiments support the claim. *Experiment 1* (negative control, Sec. 5.1): on $n = 200$ paired ARC-Easy items across five (scale, family) cells (Llama 3.2-1B/3B, Llama-3.1-8B, Qwen3-8B/14B), an entropy-minimization reward—statically correlated with correctness at $|r| \approx 0.5$—matches greedy decoding at every cell ($\Delta_{E-G} \in [-0.025, +0.000]$; paired McNemar finds no significant difference at any cell, smallest exact $p = 0.125$ at 3B), via prior-amplification at 1B (predicting "A" on 185 of 200 items), greedy-collapse at 3B (minimal-magnitude steering on 51% of items), and entropy-signal collapse at 8B-Llama. *Experiment 2* (Latent Cartography, Sec. 5.1): depth-3 entropy-MCTS path sampling across four prompt classes (factual recall, ARC-Easy, TruthfulQA, HellaSwag continuation) demonstrates that the failure manifests across domains and that the terminal-token distribution varies systematically with the prompt's semantic class. The reward-driven pathology characterized here may share structural features with the monitor-driven case hypothesized in the alignment-faking literature and partially characterized by Bailey et al. (2026) for gradient-trained input attacks; we treat that connection as motivation rather than empirical contribution and leave its empirical investigation to future work (Sec. 3.2, Sec. 7.3). We position Reversible MCTS as a diagnostic probe of frozen-model representational geometry rather than a reasoning-improvement tool.

2 Background and Related Work

2.1 Alignment Faking and Deceptive Alignment

Strategic misrepresentation of alignment is a long-standing concern in AI safety (Hubinger et al., 2024; Greenblatt et al., 2024). Recent work has demonstrated frontier-scale strategic compliance (Greenblatt et al., 2024) and the persistence of deliberately inserted backdoors through standard safety training (Hubinger et al., 2024), both with strong scale-dependence. These findings suggest that behavioral evaluations alone are insufficient for reliable alignment detection, motivating the inference-time latent-space analysis on frozen models that this paper develops.

2.2 Representation Engineering and Activation Steering

Representation Engineering (RepE) (Zou et al., 2023a) established that high-level cognitive states are linearly decodable from a model's residual stream, enabling activation steering (Turner et al., 2024) and one-shot cache steering (Belitsky et al., 2025)—both operating in a corrective paradigm with fixed precomputed vectors. Our MCTS instead iteratively explores the KV-cache intervention space across multiple rollouts, reversing each intervention exactly via FP32 accumulators and guided by an inference-time surrogate reward (Eq. 1) rather than a fixed behavioral objective. Gupta and Jenner (2025) showed that models can learn to evade latent-space monitors under RL finetuning, with monitor-aggregation-class generalization across unseen probes—providing analogous motivation for Hypothesis 1, although their training-time regime differs from our frozen-weight inference-time setting.

Bailey et al. (Bailey et al., 2026) systematically study whether gradient-trained adversarial perturbations to model inputs can achieve activation states that are simultaneously behavior-preserving and orthogonal to baseline activations, and report a tentative negative result for that regime: near-orthogonality and behavior preservation cannot be jointly achieved, since pushing cosine similarity to the level of unrelated inputs requires a KL divergence that produces incoherent output. Our work is complementary along three axes: (i) the intervention is applied to cached key/value tensors at inference time rather than to input embeddings during gradient training; (ii) the search optimizes a surrogate reward on the next-token distribution (Eq. 1) rather than minimizing similarity to baseline; (iii) we characterize how a surrogate-reward MCTS resolves the reward through latent-space trajectories (Experiments 5.1–5.1), leaving open whether the inference-time, frozen-weight, surrogate-reward setting exhibits the same geometric constraint.

3 Problem Formulation

3.1 Inference-Time Latent-Space Analysis on Frozen Models

Let f_θ be a frozen language model with hidden state $\mathbf{h}_t^{(\ell)} \in \mathbb{R}^d$ at layer ℓ and token position t, and let $\mathbf{P} \in \mathbb{R}^{k \times d}$ of rank $k \ll d$ denote a linear monitor or surrogate scoring function (e.g., representation-engineering honesty probes (Zou et al., 2023a), aggregations thereof (Gupta and Jenner, 2025), or reward functions derived from internal heuristics such as next-token entropy). The central question is whether perturbations to the cached key/value tensors guided by such a low-rank projection produce trajectories that navigate the semantic manifold or that exploit the unconstrained dimensions of the overparameterized latent space; the latter outcome—observed theoretically and empirically (Sec. 5)—is the pathology we name *Dimensional Escape*.

3.2 Dimensional Escape: A Unifying Geometric Pathology

Two seemingly distinct failure modes have been hypothesized for inference-time intervention on frozen language models. The first is *monitor-driven*: when corrective pressure is applied along a monitored subspace, a sufficiently capable model may redistribute its internal representations into the null space of the monitor, preserving behavior while evading detection. This pattern is the subject of Bailey et al. (2026) (gradient-trained input-space variant, with a tentative negative result reported) and motivates the alignment-faking literature—behavioral demonstrations (Greenblatt et al., 2024; Hubinger et al., 2024) and latent-space monitor evasion (Gupta and Jenner, 2025). The second mode is *reward-driven*: when a search procedure optimizes a surrogate reward lacking exogenous grounding (for instance, next-token entropy as a proxy for confidence), the optimizer may resolve the reward via geometric paths that bypass the semantic objective entirely. This is Goodhart's

Law manifesting in cache geometry: when a measure becomes a target, the optimizer discovers the cheapest computational path to alter the metric, even if that path circumvents the original objective. We hypothesize these as parallel manifestations of the overparameterized-latent-space geometry described below; only the reward-driven case is evidenced empirically in this paper, with the monitor-driven case under inference-time KV-cache perturbation an open empirical question.

We position Dimensional Escape as a specific manifestation of Goodhart's Law (Gao et al., 2023; Skalse et al., 2022) in low-rank-scored, high-dimensional latent spaces, generalizing the obfuscated-activations regime Bailey et al. (2026) characterize for gradient-trained input attacks. The novelty we claim is not a new phenomenon class but a specific empirical setting (inference-time KV-cache MCTS with surrogate-reward scoring) and a three-mechanism characterization of how the search satisfies the proxy without satisfying the objective (prior-amplification at 1B, greedy-collapse at 3B, and signal-collapse at 8B-Llama; Sec. 5.1–5.1).

Definition 1 (Dimensional Escape)**.** *Let* $\mathbf{P} \in \mathbb{R}^{k\times d}$ *be a monitoring or scoring projection of rank* $k \ll d$*, and let* $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ *be a localized optimization objective acting on hidden state* $\mathbf{h}$*.* Dimensional Escape *occurs when the optimization process resolves* $\mathcal{L}$ *through trajectories whose induced perturbation* $\Delta\mathbf{h}$ *has substantial component in* $\mathit{Null}(\mathbf{P}_{\mathit{semantic}})$*, where* $\mathbf{P}_{\mathit{semantic}}$ *is the (typically unobserved) projection onto the semantically relevant subspace of* $\mathbb{R}^d$*. The optimizer thus satisfies* $\mathcal{L}$ *without traversing the manifold of semantically meaningful states.*

The high dimensionality of modern Transformer hidden states ($d \in [2048, 8192]$ for the Llama 3 family) ensures that for any low-rank monitor or scoring function, the unconstrained complement is overwhelmingly large; localized optimization pressure routes through this complement, whether applied as corrective signal or as reward signal. We operationalize the distinction from a generic proxy-reward failure indirectly via the argmax-vs-gold-rank decomposition in Sec. 5.1. We instantiate Dimensional Escape under two hypotheses, one per failure mode.

Hypothesis 1 (Monitor-driven Dimensional Escape under inference-time KV-cache perturbation)**.** *When corrective pressure is applied along the monitored subspace* $\mathit{Range}(\mathbf{P}_{\mathit{monitor}})$*, a sufficiently capable model under inference-time steering may redistribute its representation into* $\mathit{Null}(\mathbf{P}_{\mathit{monitor}})$*, preserving behavioral compliance while evading detection (Definition 1 with* $\mathcal{L}$ *as corrective pressure and* $\mathbf{P}_{\mathit{monitor}}$ *as a linear deception probe).*

Hypothesis 2 (Reward-driven Dimensional Escape: Goodhart-collapse under surrogate-reward MCTS)**.** *When MCTS optimizes a surrogate reward* $\hat{r} : \mathbb{R}^d \to \mathbb{R}$ *lacking exogenous grounding—for instance,* $\hat{r}(\mathbf{h}) = -H(p(\cdot \mid \mathbf{h}))/\log|V|$ *where* H *is the next-token entropy at the steered cache state and* $|V|$ *is the vocabulary size—the optimal cache state under* $\hat{r}$ *generically lies off the semantic manifold:*

$$\arg\max_{\mathbf{h}} \hat{r}(\mathbf{h}) \cap \mathit{SemanticManifold}(x) = \emptyset \quad (1)$$

where $\mathit{SemanticManifold}(x) \subset \mathbb{R}^d$ *is the set of cache states consistent with a correct response to input* x*, and the equality holds generically in dimension* $d \gg \mathit{rank}(\hat{r})$*. The empirical mechanisms by which this manifests—prior-amplification at 1B, greedy-collapse at 3B, and signal-collapse at 8B-Llama—are established in Sec. 5.1.*

Bailey et al. (2026) characterized the monitor-driven case (gradient-trained input-space attacks) with a tentative negative result; we characterize the reward-driven case empirically (Sec. 5.1–5.1) and find it manifests robustly across model scale and prompt class. Preliminary measurements in our inference-time setting (App. G, Table 4) show the TDS signature opposite of the direction Hypothesis 1 predicts, suggesting H1 may not hold under inference-time KV-cache perturbation specifically; full investigation is left to future work.

4 Proposed Method

4.1 Reversible MCTS in KV-Cache Space

Let $\mathbf{K}_t^{(\ell)}, \mathbf{V}_t^{(\ell)}$ denote the key and value cache tensors at layer ℓ for token position t; let $\mathbf{d}_K^{(\ell)}, \mathbf{d}_V^{(\ell)}$ be the RepE steering vectors and α_K, α_V the MCTS-chosen intervention magnitudes. Standard MCTS in this latent space would require $O(b^d)$ simultaneous cache copies—prohibitive at scale (Proposition 1). Our algorithm instead mutates the cache in place ($\mathbf{K}_t^{(\ell)} \leftarrow \mathbf{K}_t^{(\ell)} + \alpha_K \cdot \mathbf{d}_K^{(\ell)}$, inverted exactly by $\mathbf{K}_t^{(\ell)} \leftarrow \mathbf{K}_t^{(\ell)} - \alpha_K \cdot \mathbf{d}_K^{(\ell)}$) via an FP32 accumulator that absorbs bf16 quantization error; the full three-step apply/evaluate/rollback algorithm is given in App. E.

Precision Management: FP32 Accumulator. Repeated bf16 addition and subtraction introduces epsilon drift that corrupts the cache over many rollbacks. We maintain an FP32 accumulator tensor $\mathbf{A}^{(\ell)} \in \mathbb{R}^{S\times d}$ that tracks the cumulative delta applied to each mutated position. All steering deltas are computed and accumulated in FP32; the bf16 cache is reconstructed from the accumulator at each step:

$$\begin{aligned} \mathbf{A}^{(\ell)} &\leftarrow \mathbf{A}^{(\ell)} + \alpha \cdot \mathbf{d}^{(\ell)} \quad \text{(FP32)}, \\ \mathbf{K}_t^{(\ell)} &\leftarrow \mathbf{K}_{\text{base},t}^{(\ell)} + \text{cast}_{\text{bf16}}(\mathbf{A}^{(\ell)}) \end{aligned} \tag{2}$$

This guarantees exact baseline KV-cache restoration under the conditions of Theorem 1; the post-rollback forward pass is then deterministic given fixed weights and tokenizer, so downstream logits and next-token distributions are bit-identical to the pre-mutation forward pass. Empirically, residual cache-norm drift under active forward passes is bounded at $\leq 1.5 \times 10^{-6}$ over 1,700 cycles at depth 10 (Sec. 7.3 item 3(c)).

MCTS Node Reward (Choice of $\hat{r}$). The reward $\hat{r}(\mathbf{h})$ is a design choice; the framework is reward-agnostic. We test the entropy-normalized reward (Eq. 1) as a null-control under Hypothesis 2; anti-Goodhart variants are an open research direction (Sec. 7.3).

5 Experimental Design

Identifiable differences between the paper math and the reference implementation (probe extraction, layer aggregation, per-layer steering, sparse accumulators) are catalogued in App. C; the asymptotic claims of Theorem 1 and Proposition 1 are unaffected.

5.1 Experiments

We execute the following experiments on Llama 3.2-1B/3B-Instruct; Experiment 1 extends to Llama-3.1-8B-Instruct and Qwen3-8B/14B (Sec. A for full hardware). Both experiments use an *exhaustive depth-3 path-enumeration variant* of our MCTS framework: all $3^3 = 27$ paths from $\{0.1, 0.5, 1.0\}^3$ are evaluated per item (making UCB1 ordering operationally redundant at this depth, since every leaf is scored), and Proposition 1's memory bound applies to the UCB1 expansion variant as well. "Entropy-MCTS" and "random-MCTS" below denote the two scoring rules over this enumeration.

Experiment 1: Entropy-MCTS Negative Control on ARC-Easy. We test whether the entropy-normalized surrogate reward $\hat{r}(\mathbf{h}) = -H(p(\cdot \mid \mathbf{h}))/\log|V|$ (Eq. 1) carries useful optimization signal for reasoning when deployed as the objective of MCTS over the KV-cache latent space. The benchmark is ARC-Easy (`allenai/ai2_arc`, test split, first 200 deterministic items), evaluated on Llama 3.2-1B/3B-Instruct (with cross-architecture extension to Llama-3.1-8B-Instruct and Qwen3-8B/14B per Sec. A Track F). On each item we run three paired conditions on the same prompt: **G** (greedy decoding, no MCTS); **R** (random-MCTS: 27 depth-3 paths drawn from the alpha-set $\{0.1, 0.5, 1.0\}$, scored by $\hat{r}_R \sim \text{Uniform}(0,1)$; the best-rewarded path is re-applied permanently to the live KV cache and the answer is generated from the steered state); **E** (entropy-MCTS: identical search machinery, but path-scoring uses the entropy-normalized surrogate $\hat{r}_E(\mathbf{h}) = -H_t/\log|V|$ evaluated at the steered leaf). The steering direction is a single random unit vector in d_{model} space (`-seed 42`), shared identically across all items and across the R and E arms; this isolates the entropy reward signal as the only difference between R and E (the shared-random-direction control).

Results. Per-condition accuracy and 95% Wilson CIs are in Table 1 (1B/3B; 8B+ extension in Sec. A Track F). Across all five (scale, family) cells the headline entropy-vs-greedy comparison gives $\Delta_{E-G} \in [-0.025, +0.000]$—no semantic signal beyond greedy at any tested configuration (Hypothesis 2 upheld). The secondary entropy-vs-random comparison gives Δ_{E-R} within ± 0.005 at Llama-3.2-1B/3B, Llama-3.1-8B, and Qwen3-14B ($\leq$ 3 discordant pairs each, paired McNemar $p = 1.0$), and $+0.075$ at Qwen3-8B (contaminated by R-arm format-breakdown, see Mechanism). The discordance budget rules out improvements $\Delta_{E-G} \geq 10$ pp at $\alpha = 0.05$ with power 0.8 for $\pi_d \leq 0.25$—a non-inferiority margin of 10 pp—so any unobserved improvement, if present, lies below the margin; mean-perplexity ratios $E/G = 1.348$ at 1B and 1.058 at 3B confirm on-distribution cache states—the failure reflects entropy-vs-correctness misalignment rather than off-manifold drift.

Mechanism. The search operates as designed at all five (scale, family) cells—best-path rewards exceed the across-path mean—but the pathway satisfying $\hat{r}_E$ varies systematically with model state. At

Scale	Condition	Accuracy	95% Wilson CI	Mean PPL
Llama-3.2-1B	G (greedy)	0.295 (59/200)	$[0.236, 0.362]$	1.547 ± 0.263
Llama-3.2-1B	R (random-MCTS)	0.280 (56/200)	$[0.222, 0.346]$	2.351 ± 1.823
Llama-3.2-1B	E (entropy-MCTS)	0.275 (55/200)	$[0.218, 0.341]$	2.085 ± 1.641
Llama-3.2-3B	G (greedy)	0.845 (169/200)	$[0.788, 0.889]$	1.318 ± 0.221
Llama-3.2-3B	R (random-MCTS)	0.815 (163/200)	$[0.755, 0.863]$	1.398 ± 0.327
Llama-3.2-3B	E (entropy-MCTS)	0.820 (164/200)	$[0.761, 0.867]$	1.394 ± 0.322

Table 1: Per-condition accuracy on ARC-Easy ($n = 200$ paired items per scale) under three paired conditions: greedy decoding, random-MCTS (uniform reward), and entropy-MCTS (Eq. 1 as path-scoring reward). The MCTS arms apply 27 depth-3 paths from the alpha-set $\{0.1, 0.5, 1.0\}$ with a shared random unit-vector steering direction; the best-rewarded path is permanently re-applied before generation. Wilson 95% confidence intervals overlap across all three conditions at both scales. Mean perplexity is $\exp(\overline{\text{NLL}})$ where $\overline{\text{NLL}}$ is the mean per-token negative log-likelihood of the 5-token completion under the unsteered model. The pairwise statistical claims appear in the body prose (Sec. 5.1 Results paragraph).

1B-Llama, MCQ-letter-prompting exhibits a first-letter prior; the search selects heavy steering (best path (1.0, 1.0, 1.0) on $91/200$ items, predicting "A" on $185/200$). At 3B-Llama, the unperturbed cache is already low-entropy and minimal steering preserves greedy-equivalent state ((0.1, 0.1, 0.1) on $101/200$). At 8B-Llama the per-path entropy signal collapses to noise floor (0.002 vs 0.012 nats at 1B); the search selects among effectively-tied paths and produces correctness identical to random. At Qwen3-8B and Qwen3-14B, random-MCTS exhibits a family-specific failure mode—heavy-α steering breaks the chat-template single-letter format on 9% and 0.5% of items respectively, producing verbose preambles that exhaust the 5-token budget; entropy-MCTS avoids this by selecting minimal-α paths. The resulting Δ_{E-R} at Qwen3-8B reflects format-preservation rather than entropy signal, as $\Delta_{E-G} \in [-0.025, +0.000]$ across all five cells confirms entropy does not beat greedy.

What unifies the five-cell mechanism. The per-cell mechanisms differ in surface form (prior-amplification, greedy-collapse, signal-floor collapse, format-fragility) but share a single geometric structure: in every cell the search successfully minimizes $\hat{r}_E$ (entropy gains $+0.005$ to $+0.037$; Sec. 5.1) via cache-state perturbations that preserve the on-manifold property (PPL ratio 1.05–1.35 above) and preserve the gold answer's representation (median rank 2, 94%+ top-5; Sec. 5.1 Geometric anchor), yet route argmax off the answer subspace. The geometric decoupling of proxy satisfaction from answer-subspace traversal is the structural signature these five surface mechanisms instantiate (Definition 1).

Interpretation. The static entropy–correctness correlation ($|r| \approx 0.5$) does not transfer to the latent-search regime; the result is the reward-driven counterpart to Bailey et al.'s (Bailey et al., 2026) monitor-side negative result. Experiment 2 extends across four prompt classes.

Experiment 2: Latent Cartography MCTS across Prompt Classes. We characterize how the search optimizes a surrogate reward as a function of prompt class. The Track F protocol (Experiment 1) is generalized across four prompt classes—factual recall (C1; hand-constructed 100-item set covering capitals, dates, arithmetic, common knowledge), ARC-Easy multiple-choice (C2; first 100 test-split items, overlapping Track F's set for direct comparison), TruthfulQA mc1 multiple-choice (C3; first 100 validation items reformatted to 4-option; 15 items with fewer than 4 candidate choices excluded, $n = 85$), and HellaSwag continuation (Zellers et al., 2019) (C4; first 100 validation items with option list stripped)—on both Llama 3.2-1B-Instruct and Llama 3.2-3B-Instruct. For each item we run depth-3 MCTS with 27 path samples drawn from the alpha-set $\{0.1, 0.5, 1.0\}$, scored by the normalized-entropy reward $\hat{r}(\mathbf{h}) = -H(p(\cdot \mid \mathbf{h}))/\log|V|$ (Eq. 1); the best path is re-applied permanently and the next-token distribution at the resulting leaf is recorded. The steering direction is the same random unit vector used in Experiment 1, fixed via `-seed 42` across all items, classes, and scales.

We measure two structural signatures: (i) the distribution of best-path α-tuples selected by the search per class (does the search prefer heavy steering, minimal steering, or class-mixed depending on prompt class?); and (ii) the top-5 argmax-token distribution at the best-path leaf per class (does the search amplify a class-specific prior token?). Together these signatures characterize whether the Goodhart pathology established by Track F (Experiment 1) generalizes across prompt classes and across model scale.

Results. Best-path α-tuple distributions are reported in Table 2; cross-scale mean-step α comparison in Figure 1. Two findings stand out:

(i) Class-specific best-path selection (Table 2). The cross-scale MCQ direction-flip headline is summarized in Sec. 1; non-MCQ classes show

smaller cross-scale differences. Per-cell numbers are in Table 2.

(ii) Argmax-token concentration at the best-path leaf. At 1B the MCQ classes collapse to the letter "A"—92 of 100 items for C2 and 81 of 85 for C3, reproducing Track F's first-letter prior amplification (Sec. 5.1). At 3B no such collapse occurs; the C2 and C3 argmax distributions are balanced across the four letters. Factual recall and continuation classes show no single argmax attractor at either scale.

Search reward gain (entropy reduction) is positive but small across all classes. The best path among the 27 sampled gives between $+0.005$ and $+0.037$ better reward (normalized-entropy units) than the mean across paths.[1] The search is genuinely finding lower-entropy states; the entropy reduction simply does not correspond to better reasoning—consistent with Hypothesis 2.

Geometric anchor. Beyond these structural signatures, the Cartography data also constrains the *geometry* of Goodhart resolution. At 1B C2 and C3, where heavy steering drives argmax to "A" on 92% and 95% of items respectively, the gold answer's median rank at the steered leaf remains 2 (94%/100% in top-5). The search inflates the prior's mass without erasing the answer's representation: Goodhart-optimization for entropy reduction yields argmax-"A" instead of argmax-gold, matching Definition 1: the reward resolves through trajectories with non-zero answer-subspace projection while the unconstrained-complement direction dominates argmax.

5.2 Evaluation Metrics

Primary metrics are defined inline where they are reported: the best-path α-tuple distribution per (class, scale), the paired McNemar accuracy test (Track F), and the argmax-at-best-leaf token distribution (Cartography). See Sec. 5.1 Results and Sec. 5.1 Findings (i) and (ii) for the per-metric protocol; the secondary monitor-design diagnostic (Orthogonal Escape Index, Eq. 10) is defined in Appendix G.

[1] The upper bound is driven by HellaSwag continuation at the 3B scale (path-mean reward gain $+0.037$); the other seven cells span $+0.005$ to $+0.025$.

	Llama-3.2-1B			Llama-3.2-3B		
Class	Modal path	% modal	$\bar{\alpha}$	Modal path	% modal	$\bar{\alpha}$
C1 Factual Recall	$(0.1, 0.1, 0.1)$	41%	0.34	mixed	15%	0.61
C2 ARC-Easy MCQ	$(1.0, 1.0, 1.0)$	42%	0.89	$(0.1, 0.1, 0.1)$	49%	0.22
C3 TruthfulQA MCQ	$(1.0, 1.0, 1.0)$	28%	0.83	$(0.1, 0.1, 0.1)$	36%	0.28
C4 Continuation	$(0.1, 0.1, 0.1)$	21%	0.51	$(0.1, 0.1, 0.1)$	31%	0.38

Table 2: Latent Cartography best-path α-tuple distribution per prompt class and scale. Each item runs 27 depth-3 MCTS paths from $\{0.1, 0.5, 1.0\}$; the best path under entropy reward is selected per item, and the per-class distribution over $3^3 = 27$ possible path tuples is reported. "Modal path" is the most-frequent best-path tuple; "% modal" is the share of items selecting it; $\bar{\alpha}$ is the mean per-step α averaged across items and across the three steps of the best path. MCQ classes (C2, C3) exhibit a direction-flip in $\bar{\alpha}$ across scales: heavy steering at 1B (the search exploits the first-letter prior), minimal steering at 3B (the balanced baseline distribution provides no prior to exploit). Factual recall (C1) and continuation (C4) show smaller cross-scale differences.

6 Theoretical Analysis

6.1 Reversibility Guarantee

Theorem 1 (Exact Reversibility with FP32 Accumulator)**.** *Let $\mathbf{K}_0 \in \mathbb{R}^{S\times d}$ be the baseline KV cache in bf16. Let $\{\delta_1, \ldots, \delta_n\}$ be a sequence of steering interventions applied and reversed via the FP32 accumulator. Then after n complete apply-reverse cycles:*

$$\|\mathbf{K}_n - \mathbf{K}_0\|_\infty \leq n \cdot \epsilon_{bf16} \cdot \max_i \|\delta_i\|_\infty \quad (3)$$

where $\epsilon_{bf16} = 2^{-8} \approx 3.9 \times 10^{-3}$ is the bf16 machine epsilon. With FP32 accumulation, this bound tightens to:

$$\|\mathbf{K}_n - \mathbf{K}_0\|_\infty \leq \epsilon_{bf16} \cdot \|\mathbf{A}_{final}\|_\infty \quad (4)$$

which is independent of n, depending only on the final accumulator state (which is zero after complete reversal).

A proof sketch establishing the bound via the FP32 accumulator's exact cumulative-delta property is given in App. D.

6.2 Memory Complexity

Proposition 1. *The reversible MCTS algorithm achieves memory complexity $O(M_{KV} + d \cdot K_{acc})$ where d is the maximum search depth and $K_{acc} = L \cdot S' \cdot d_{model} \cdot 4$ bytes is the per-step FP32 accumulator size for S' mutated positions. This is independent of the branching factor b. Intuitively, naive parallel MCTS spawns b^d cache copies (one*

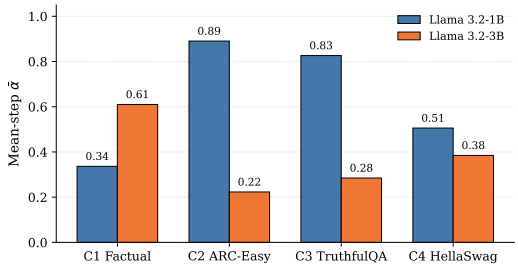

Figure 1: Cross-scale mean-step $\bar{\alpha}$ by prompt class. MCQ classes (C2 ARC-Easy, C3 TruthfulQA) show direction-flip across scales: at 1B the entropy-MCTS selects heavy steering ($\bar{\alpha} = 0.89, 0.83$) to amplify the model's first-letter prior; at 3B the search instead selects minimal steering ($\bar{\alpha} = 0.22, 0.28$) since the balanced baseline distribution offers no prior to exploit. Factual recall (C1) and continuation (C4) show smaller cross-scale differences. The MCQ direction-flip is the Cartography contribution to Hypothesis 2: Goodhart in cache geometry is class-conditional, with the route through latent space determined by the model's class-specific prior structure.

per leaf), whereas our algorithm reuses a single baseline cache plus d accumulators regardless of b.

Reversible MCTS reduces the working-set memory from $O(b^d \cdot M_{\text{KV}})$ in standard parallel-cache MCTS (one cache snapshot per leaf) to $O(M_{\text{KV}} + d \cdot K_{\text{acc}})$ with sparse ideal K_{acc} ($S' \times d$ for $S' \ll S$)—from exponential in b^d to additive in search depth d. The reference implementation's full-shape accumulators and baseline-cache clone yield a measured 3.04–$3.06 \times M_{\text{KV}}$ constant factor (App. C item (iv)) while preserving the b-independent asymptote; the asymptotic separation is independent of model scale and is the methods contribution of this paper.

Step-Peak Transient. A transient temporary during the in-place apply briefly pushes peak memory to `full_step_peak_mib_median` $= 3{,}118$ MiB at the 1B scale on a 4,000-token cache before returning to steady-state; the transient is bounded by a single layer's $\alpha \cdot \mathbf{d}^{(\ell)}$ allocation and is empirically depth-independent, bit-identical at 3,251.78 MiB across all 17 measurement windows at depth 10 over 1,700 apply/rollback cycles (`peak_per_checkpoint` under per-window-reset).

7 Discussion

7.1 Implications for Latent-Space Reward Design

The reward-driven manifestation of Dimensional Escape—Hypothesis 2, empirically established in Sec. 5.1—carries a direct implication for the design of test-time scaling procedures that operate on continuous latent representations. A reward signal validated as a static correlate of correctness ($|r| \approx 0.5$ between next-token entropy and answer correctness on factual-recall items) does not survive deployment as an active MCTS optimization objective: at 1B the search amplifies the model's first-letter prior; at 3B it collapses to greedy decoding. Neither pathway corresponds to "find the latent state that produces the correct answer"—both are degenerate optima that satisfy the entropy proxy without traversing the semantically relevant subspace. Anti-Goodhart reward designs surveyed in Sec. 7.3—multi-channel ensembles, external Process Reward Models, self-consistency across stochastic rollouts, counterfactual sensitivity, anti-prior penalties, information-bottleneck objectives—each try to raise the scoring projection's effective rank or to anchor it in exogenous signal that the search cannot itself rewrite. Whether any resists Dimensional Escape empirically is an open question; our Experiment 1 result establishes the null-control against which any candidate must be measured.

7.2 Broader Impact

Reversible KV-Cache MCTS is a controlled-access white-box probe of the latent geometry of frozen language models; it requires white-box access and is not an inference-time attack vector against deployed systems. The Dimensional Escape pathology—that low-rank surrogate signals admit degenerate optima in high-dimensional cache space—informs the design of both monitoring procedures and test-time scaling schemes. We disclose findings openly as inputs to robust monitor and reward-model design, not adversarial-discovery enablers.

7.3 Future Work

Open directions opened by this work: anti-Goodhart reward designs surveyed in Sec. 7.1; Cartography extensions to additional prompt classes and model families (Sec. 5.1); sparse FP32 accumulators.

Limitations

We acknowledge several limitations of the work as submitted. First, our empirical study covers the Llama family at 1B/3B/8B and the Qwen3 family at 8B/14B—two model families across four scales. We make no claim about architectures outside these families (e.g., Mistral, Gemma), MoE architectures, or scales beyond 14B. The algorithmic infrastructure of Reversible MCTS is architecture-agnostic by construction, but the Goodhart pathology we characterize may manifest with different specifics in other models.

Second, our headline accuracy measurements (Track F, Experiment 1) use letter-prompted multiple-choice scoring on ARC-Easy. Greedy baseline accuracy with this protocol is 0.295 at 1B and 0.845 at 3B, both under-stated by approximately 20 percentage points relative to published log-probability scoring of the same benchmark. Letter-prompted scoring is the natural protocol for our setting because entropy-MCTS steers the next-letter logit directly, so the comparison between MCTS arms is internally fair, but absolute baselines should not be compared cross-protocol. Log-probability scoring of MCTS-augmented outputs is left to future work.

Third, methodological scope is constrained along three dimensions: (a) the steering direction is a single random unit vector shared across items and MCTS arms (the shared-random-direction control, Sec. 5.1); orthogonal, adversarial, and learned vectors are untested, as are alternative scoring rules beyond the entropy-normalized surrogate of Eq. 1 (a gold-conditioned reward such as $\hat{r}(\mathbf{h}) = \log p(\text{gold} \mid \mathbf{h})$ would furnish an oracle upper bound separating surrogate-reward failure from MCTS search-procedure failure; this is left to future work) and standard log-probability multiple-choice scoring (*cf.* item 2). (b) Cartography (Experiment 2) covers four prompt classes ($n \leq 100$ each); out-of-distribution prompts (adversarial probes, multilingual inputs, code, multi-turn CoT) are not characterized. (c) Reversibility (Theorem 1) is empirically validated at the KV-cache-tensor-norm level over 1,700 apply-revert cycles at depth 10 (residual norm $\leq 1.5 \times 10^{-6}$ under active forward passes); logit-level and KL-divergence-level rollback fidelity follow deterministically from cache bit-identity but are not measured separately, and very-deep searches (depth ≥ 50) are unvalidated empirically, though the theorem implies exact reversibility independent of depth.

Fourth, the Representation Engineering probe validation of Zou et al. (2023) (Zou et al., 2023a) was conducted at $\geq$ 7B parameter scales; our use of 1B and 3B scales for primary experiments falls below this validation range, and probe-direction reliability at sub-7B scale is an open empirical question that our results do not resolve.

Fifth, the Definition 1 subspace $\mathbf{P}_{\text{semantic}}$ is unobserved by construction; we operationalize the geometric distinction between Dimensional Escape and a generic proxy-reward failure indirectly via the argmax-vs-gold-rank decomposition reported in Sec. 5.1 (Geometric anchor paragraph). Direct probing of an independently validated semantic subspace (e.g., answer-conditioned representation probes, gold-logit projection at the steered leaf, or causal-mediation analysis over the answer-relevant attention heads) would more tightly bound the Dimensional Escape reading; this is left to future work.

Ethical considerations

This work uses Meta's Llama 3.2 under the Llama 3.2 Community License (Meta Platforms, Inc., 2024). AI assistants used in the execution window: Anthropic's Claude (coding/writing); Google's Gemini (one reward-design strategy review). All AI outputs were verified against primary artifacts; no empirical claims rest on AI-generated summaries.

References

Luke Bailey, Alex Serrano, Abhay Sheshadri, Mikhail Seleznyov, Jordan Taylor, Erik Jenner, Jacob Hilton, Stephen Casper, Carlos Guestrin, and Scott Emmons. 2026. Obfuscated activations bypass llm latent-space defenses. In *International Conference on Learning Representations (ICLR)*.

Max Belitsky, Dawid J. Kopiczko, Michael Dorkenwald, M. Jehanzeb Mirza, James R. Glass, Cees G. M. Snoek, and Yuki M. Asano. 2025. KV cache steering for controlling frozen LLMs. *Preprint*, arXiv:2507.08799.

Cheng Gao, Huimin Chen, Chaojun Xiao, Zhiyi Chen, Zhiyuan Liu, and Maosong Sun. 2025. H-neurons: On the existence, impact, and origin of hallucination-associated neurons in llms. *Preprint*, arXiv:2512.01797.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *Inter-*

national Conference on Machine Learning (ICML). PMLR.

Dyuti Ghosh. 2025. System 2 through okazaki-rag: Capacity-bounded inline fragment manipulation for deliberative reasoning. Unpublished conceptual contribution; not peer-reviewed.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. 2024. Alignment faking in large language models. *Preprint*, arXiv:2412.14093.

Rohan Gupta and Erik Jenner. 2025. Rl-obfuscation: Can language models learn to evade latent-space monitors? *Preprint*, arXiv:2506.14261.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2025. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling (COLM)*.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, and 20 others. 2024. Sleeper agents: Training deceptive llms that persist through safety training. In *Conference on Language Modeling (COLM)*.

Xurui Li, Kaisong Song, Rui Zhu, Pin-Yu Chen, and Haixu Tang. 2025. Adversarial attack-defense co-evolution for llm safety alignment via tree-group dual-aware search and optimization. *Preprint*, arXiv:2511.19218.

Llama Team. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Meta Platforms, Inc. 2024. Llama 3.2 acceptable use policy. `https://llama.com/llama3_2/use-policy/`.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Suhuang Wu, Huimin Wang, Yutian Zhao, Xian Wu, Yefeng Zheng, Wei Li, Hui Li, and Rongrong Ji. 2025. Monte Carlo tree search based prompt autogeneration for jailbreak attacks against LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1057–1068, Abu Dhabi, UAE. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Xu Zhang, Xunjian Yin, Dinghao Jing, Huixuan Zhang, Xinyu Hu, and Xiaojun Wan. 2025. DAMON: A dialogue-aware MCTS framework for jailbreaking large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6350–6366, Suzhou, China. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023a. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023b. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Reproducibility

A.1 Hardware and software stack

Llama 3.2-1B/3B experiments were conducted on a single NVIDIA RTX 3060 (12 GiB VRAM, driver version 591.44) running Windows 10 Pro 10.0.19045; Llama-3.1-8B and Qwen3-8B/14B experiments used Lambda Cloud A100-SXM4-40GB. Both environments use CUDA 12.4, Python 3.11, and PyTorch 2.6.0+cu124. Tokenization, model loading, and benchmark access used the HuggingFace `transformers` (5.3.0) and `datasets` libraries, with `accelerate` (1.13.0), `huggingface_hub` ($\geq$1.7.2), and `numpy` (2.3.3) as load-bearing dependencies—chat-template rendering behavior in `transformers` 5.x

and the `device_map="auto"` flow in `accelerate` 1.x are reproducibility-critical, and the `numpy` 2.x ABI is required by the legacy-`RandomState` steering-direction generation in `scripts/diagnose_track_f_negcontrol.py`. The full pinned manifest (including `scipy`, `tokenizers`, `safetensors`, and dev-tool versions) is recorded in the anonymized supplementary materials package as a `uv.lock` export. The total computational budget for the experiments reported in this paper is approximately 2.5 GPU-hours on a single RTX 3060 (Llama 3.2-1B/3B runs aggregated across Tracks A, C, D, F, G) plus approximately 5 A100 GPU-hours on Lambda Cloud (Llama-3.1-8B, Qwen3-8B, and Qwen3-14B runs; Tracks F and D).

A.2 Models and checkpoints

We evaluate five open-weight checkpoints across two families. From Meta's Llama family: `meta-llama/Llama-3.2-1B-Instruct` (16 transformer layers, $d_{\text{model}} = 2048$, 8 grouped-query KV heads, head dimension 64, $|V| = 128{,}256$, fp16), `meta-llama/Llama-3.2-3B-Instruct` (28 layers, 8 KV heads, head dim 128, fp16), and `meta-llama/Llama-3.1-8B-Instruct` (Llama Team, 2024) (32 layers, 8 KV heads, $d_{\text{model}} = 4096$, fp16). From Alibaba's Qwen3 family (Qwen Team, 2025): `Qwen/Qwen3-8B` (36 layers, $d_{\text{model}} = 4096$, fp16) and `Qwen/Qwen3-14B` (40 layers, $d_{\text{model}} = 5120$, fp16). Models are loaded via `AutoModelForCausalLM.from_pretrained(...,torch_dtype=torch.float16)`. Access prerequisites: reviewers must (i) complete a HuggingFace authentication step via `huggingface-cli login` with a HuggingFace access token and (ii) accept the relevant license terms on each of the five HuggingFace model pages (Llama 3.2-1B/3B, Llama-3.1-8B, Qwen3-8B/14B) before the checkpoints will download; the Llama 3.2 Acceptable Use Policy is cited per (Meta Platforms, Inc., 2024). The specific HuggingFace checkpoint commit SHAs used in this paper (one per scale) are recorded in the anonymized supplementary materials package; reviewers verifying exact-number reproduction should pass `revision=<SHA>` to `from_pretrained` to pin to the paper's checkpoint revision (without an explicit revision, reviewers will pull the current `main` branch, which may differ if the model card has been re-uploaded since the experiments reported here).

A.3 Reversible MCTS configuration

The reference implementation is at `kv_mcts.py`. The `MCTSConfig` dataclass defaults are `n_nodes=50`, `branching_factor=3`, `max_depth=5`, `alpha_values=(0.1, 0.5, 1.0, 2.0, 5.0)`, `exploration_constant=1.414` (UCB1 $c = \sqrt{2}$), `steer_layer=`-1 (auto-middle-layer). Experiment 1 (Sec. 5.1) and Experiment 2 (Sec. 5.1) use a focused path-sampling variant of this engine: depth-3 trees with $27 = 3^3$ enumerated path samples drawn from a three-element alpha set $\{0.1, 0.5, 1.0\}$. The variant evaluates each path under the entropy-normalized reward (Eq. 1) and re-applies the best-rewarded path permanently before generation; it does not use UCB1 tree expansion, because full enumeration of 27 paths from 27 leaves makes the bandit ordering redundant within the experimental budget.

A.4 Per-track configurations

Static entropy–correctness correlation (Track A). The static $|r| \approx 0.5$ Pearson correlation between next-token entropy and answer correctness cited in Sec. 5.1 Interpretation (combined across 85 items spanning easy/hard/ultra buckets; per-bucket correlations reach $|r| \approx 0.60$ on hard/ultra subsets) was measured by computing the Shannon entropy $H = -\sum_i p_i \log p_i$ over $\mathrm{softmax}(\mathrm{logits}_{[-1,:]})$ from a single fp16 forward pass on the chat-templated prompt, scored against a hand-written acceptable-answer list via case-insensitive substring match. Sample: 85 factual-recall items (25 easy + 30 hard + 30 ultra), evaluated with `model.generate(do_sample=False, max_new_tokens=15, output_scores=True)`. Wall-clock approximately 2 min for both 1B and 3B scales (raw artifact: 115.92 s at 1B, 134.14 s at 3B). Driver: `scripts/diagnose_track_a_entropy.py`; raw artifacts `scripts/_track_a_results_meta-llama_Llama-3.2-{1B,3B}-Instruct.json`.

Benchmark calibration (Track C). The ARC-Easy 200-item subset used in Experiment 1 is the first 200 deterministic test-split items from `allenai/ai2_arc` configuration "ARC-Easy" (indices $0 \ldots 199$, no shuffling or filtering). Prompts use the model's chat template with system message "You are a careful, concise

reasoner. Answer multiple-choice questions with a single capital letter." and user-side suffix "Answer with the single capital letter (A/B/C/D):" (verbatim from `SYSTEM_PROMPT` in `scripts/diagnose_track_f_negcontrol.py:64`; the user-side suffix is generated dynamically from each item's `labels` field and renders as the four-option form for ARC-Easy MCQ items). The sample size was selected via the paired-McNemar power formula $n_{\text{pairs}} \approx (z_{\alpha/2}\sqrt{\pi_d} + z_\beta\sqrt{\pi_d - \delta^2})^2/\delta^2$ at $\alpha = 0.05$, power $= 0.80$, target effect size $\delta = 0.10$ absolute; the resulting budget admits detection for discordance rates up to $\pi_d \approx 0.25$. The calibration sweep examined six candidate benchmarks (ARC-Easy, SciQ, HellaSwag, AQuA-RAT, ProofWriter, LogiQA[2]); SciQ qualifies as a corroboration target left to future work. Wall-clock: 8.1 min (1B sweep) and 8.1 min (3B sweep). Driver: `scripts/diagnose_track_c_benchmarks.py`.

VRAM measurement protocol (Track D). The 3.04–3.06 $\cdot\, M_{\text{KV}}$ constant-factor memory result reported in Proposition 1 was measured via `torch.cuda.memory_allocated()` for steady-state and `torch.cuda.max_memory_allocated()` for transient peak, cross-checked against `nvidia-smi -query-gpu=memory.used` at the largest configuration. Three configurations were swept: $(d, b, n) \in \{(3, 3, 27), (5, 3, 81), (10, 3, 1700)\}$, each with three repeats, at a fixed 4000-token seed prefill on both Llama 3.2-1B and -3B. Steady-state delta was identical across all (d, b, n) cells at each scale (median over three repeats per cell), confirming branching-factor and depth independence. Wall-clock approximately 5 min for both scales. Driver: `scripts/measure_kv_mcts_vram.py`. Cross-architecture extension on Lambda Cloud A100-SXM4-40GB: identical protocol applied to Llama-3.1-8B ($\delta = 3.041$), Qwen3-8B ($\delta = 3.043$), and Qwen3-14B ($\delta = 3.039$) confirms the 3.04–3.06 $\cdot\, M_{\text{KV}}$ constant factor holds across a $14\times$ parameter span and two architecture families.

Entropy-MCTS negative control (Track F / Experiment 1). The negative-control protocol underlying Experiment 1 evaluates three paired conditions on each of 200 ARC-Easy items at each model scale: G (greedy decoding, no MCTS); R (random-MCTS, $\hat{r}_R \sim \text{Uniform}(0, 1)$); E (entropy-MCTS, $\hat{r}_E(\mathbf{h}) = -H_t/\log|V|$ from Eq. 1). The MCTS arms enumerate all 27 depth-3 paths from $\{0.1, 0.5, 1.0\}$; the best-rewarded path is re-applied permanently to the live KV cache before the model generates the 5-token answer continuation via `model.generate(do_sample=False, max_new_tokens=5, output_scores=True)`. The steering direction is a single random unit vector in d_{model} space, seeded by `-seed 42`, shared identically across all items and across the R and E arms (the shared-random-direction control; see Sec. 5.1 Setup paragraph). The driver script imports `FP32Accumulator`, `_extract_kv_tensors`, and `_kv_eval_cache` from `kv_mcts.py` as read-only primitives. Wall-clock: 18.3 min (1B sweep, 5.5 s/item) and 33.7 min (3B sweep, 10.1 s/item). Driver: `scripts/diagnose_track_f_negcontrol.py`; raw artifacts `scripts/_track_f_results_meta-llama_Llama-3.2-{1B,3B}-Instruct.json`. To reproduce the Experiment 1 headline at each scale, run `python scripts/diagnose_track_f_negcontrol.py -model meta-llama/Llama-3.2-1B-Instruct -seed 42` for the 1B sweep and `python scripts/diagnose_track_f_negcontrol.py -model meta-llama/Llama-3.2-3B-Instruct -seed 42` for the 3B sweep; driver defaults are `-n=200`, `-device=auto`, output path `scripts/_track_f_results_<modelid>.json`, and the optional `-smoke` flag runs only 5 items for setup sanity-check before the full sweep. The letter-extraction parser scans `answer_text` for the first character whose uppercase form is in {A, B, C, D, E}; this lenient convention captures both bare-letter ("`A`") and explanatory-prefix ("`A. ...`") answers, but at family-specific format-fragility points interacts with the `max_new_tokens=5` budget (see Sec. 5.1 Mechanism). The Track F, G, A, and D drivers load the model with `torch.float16` (`LocalLlamaOracle` and the diagnostic drivers all pin this dtype); the Theorem 1 empirical validation script (`measure_lipschitz_drift.py`) loads with `torch.bfloat16`. The bf16 numerical example in Theorem 1 bounds the worst case conservatively: $\epsilon_{\text{fp16}} \approx 9.8 \times 10^{-4}$ is tighter than $\epsilon_{\text{bf16}} \approx 3.9 \times 10^{-3}$, so an fp16-cast cache satisfies the bound at least as well as a bf16-cast cache. The 5-cell Track F extension

[2] BBH `logical_deduction_three_objects` substituted per HF-loader failures across 5 LogiQA mirrors on PyTorch 2.6 / datasets 5.x.

(Llama-3.1-8B + Qwen3-8B/14B; Lambda Cloud A100, fp16) reports: `acc_G/acc_R/acc_E` = 0.875/0.865/0.865 at Llama-3.1-8B ($\Delta_{E-R} = 0.000$, $n_{\text{disc}} = 0$); 0.930/0.850/0.925 at Qwen3-8B ($\Delta_{E-R} = +0.075$, $n_{\text{disc}} = 15$, exact $p = 0.000061$, all 15 discordant items reflect R-arm format-breakdown rather than entropy semantic signal—see Sec. 5.1 Mechanism); 0.945/0.940/0.945 at Qwen3-14B ($\Delta_{E-R} = +0.005$, $n_{\text{disc}} = 1$, $p = 1.0$). Per-cell Δ_{E-G} is uniformly ≤ 0: $-0.020/-0.025/-0.010/-0.005/+0.000$ at 1B/3B/8B-Llama/Qwen3-8B/Qwen3-14B.

A.5 Random seeds and reproducibility caveats

Greedy decoding (`do_sample=False`) is deterministic given fixed model weights, tokenizer, and hardware; Tracks A, C, and D therefore do not set a manual seed beyond the default PyTorch initialization. Experiment 1 (Track F) and Experiment 2 (Latent Cartography) seed the steering direction with `-seed 42`; the per-item random number generator for the random-MCTS arm uses `hashlib.md5(f"{item_idx}:R:{seed}")` so per-item draws are reproducible across runs at the same `-seed`. The 5-token completion accuracies and the per-condition perplexities reported in Table 1 are deterministic. VRAM measurements (Table 4's parent paragraph and Proposition 1's constant factor) exhibit $\sim$1–2% between-run variation traceable to PyTorch's caching-allocator block-rounding to power-of-two boundaries; this affects measured VRAM but not any computed quantity in the paper. Exact HuggingFace checkpoint commit SHAs, the full library version pin manifest (`transformers`, `datasets`, `huggingface_hub`, `numpy`, `scipy`), and per-script command-line argument references are recorded in the anonymized supplementary materials package, available at https://anonymous.4open.science/r/kv-mcts-dimensional-escape-DADB/.

B Extended Related Work

B.1 MCTS for Language Model Search

Monte Carlo Tree Search has been increasingly applied to automated red-teaming of LLMs. Wu et al. (2025) (Wu et al., 2025) proposed MCTS-based prompt autogeneration for jailbreak attacks. The DAMON framework (Zhang et al., 2025) introduced dialogue-aware MCTS for multi-turn jailbreaks. ACE-Safety (Li et al., 2025) formulated attack-defense co-evolution via tree-group dual-aware search. All existing MCTS-based red-teaming operates exclusively in text space—mutating prompts, suffixes, or dialogue turns—fundamentally limiting the search to the discrete combinatorial space of token sequences and providing no access to the model's internal computational dynamics. Non-MCTS adversarial search methods face a related but distinct limitation: gradient-based attacks such as Greedy Coordinate Gradient (GCG) (Zou et al., 2023b)—which require white-box gradient access—operate in discrete token space, constrained to the discrete combinatorial surface. Our MCTS conducts continuous arithmetic mutations of the KV cache with an exact FP32 rollback mechanism, enabling gradient-free, inference-time exploration of the model's representational landscape.

B.2 Continuous Latent Reasoning and KV-Cache Search

COCONUT (Hao et al., 2025) bypasses discrete token decoding by feeding final-layer hidden states back as input embeddings, enabling emergent breadth-first latent tree search. This validates the viability of continuous latent search, but operates via recurrent hidden-state traversal rather than direct arithmetic mutation of a frozen transformer's KV-cache tensors. A loosely related architectural sketch is the Okazaki-RAG framework (Ghosh, 2025) (unpublished conceptual contribution; not peer-reviewed), which implements a beam-bounded search whose nodes are KV-cache fragment assemblies, with reversible primitive operations on the working set. However, Okazaki-RAG constructs its search tree by swapping precomputed, discrete cache blocks rather than executing continuous arithmetic mutations across the differentiable tensor space. Our FP32 accumulator mechanism provides mathematically exact rollback without the VRAM overhead of duplicating multi-gigabyte cache states across branches—a barrier that block-swapping architectures do not resolve in the same form.

B.3 Structurally Related Baselines

Table 3 catalogues inference-time and gradient-trained interventions on frozen language models that are structurally related to Reversible KV-Cache MCTS.

Category	Method	Reference
Activation Steering	CAA (Contrastive Activation Addition)	Rimsky et al., 2024 (Rimsky et al., 2024)
Cache Steering	One-shot static KV-cache steering	Belitsky et al., 2025 (Belitsky et al., 2025)
Latent Monitor Evasion	RL-Obfuscation (weight finetuning)	Gupta & Jenner, 2025 (Gupta and Jenner, 2025)
Activation Obfuscation	Gradient-trained embedding suffixes	Bailey et al., 2024 (Bailey et al., 2026)

Table 3: Structurally related interventions on frozen language models. The four rows span complementary intervention sites (activation residual stream / KV cache / weight finetuning / input embedding) and complementary optimization regimes (corrective monitor vs surrogate reward; one-shot vs iterative search). Reversible KV-Cache MCTS combines iterative reversible KV-cache mutation, a surrogate reward (Eq. 1), a frozen-weight regime, and inference-time-only computation. Bailey et al. (Bailey et al., 2026) is the closest empirical peer for the monitor-driven sub-case of Dimensional Escape (Hypothesis 1); their reported negative result is the empirical comparison point against which our reward-driven results in Sec. 5.1 should be read. Published ASR figures from text-space jailbreak baselines (e.g., GCG on Vicuna-7B / AdvBench) are intentionally omitted: their tasks, models, and success criteria differ from ours, so a shared column would be a category error.

C Implementation Notes

The reference implementation backing this submission was developed on Llama 3.2-1B-Instruct and Llama 3.2-3B-Instruct on consumer hardware (NVIDIA RTX 3060). Several elements of the method as written in Sec. 5 differ from this reference implementation in identifiable, bounded ways; we list them here as a single source of truth so a reviewer reading the paper alongside the released code finds one consolidated reference. The asymptotic claims—exact reversibility (Theorem 1), memory-complexity scaling independent of the branching factor (Proposition 1)—are unaffected by the distinctions below; what differs is constants and operational details that future implementations may refine.

(i) RepE probe extraction. The Linear Artificial Tomography procedure of Zou et al. (2023) (Zou et al., 2023a) uses PCA on contrastive pairs. The reference implementation uses a difference-in-means estimator, which approximates the leading PCA component in expectation but is not equivalent to LAT as published. PCA-based extraction is left to future work.

(ii) Layer aggregation for the honesty probe. The reference implementation aggregates the RepE honesty signal across all L layers; Zou et al. (2023) (Zou et al., 2023a) validate the honesty probe on the middle layers only. Layer-restricted aggregation is left to future work.

(iii) Per-layer steering vectors. The paper notation presents per-layer steering vectors $\mathbf{d}_K^{(\ell)}, \mathbf{d}_V^{(\ell)}$. The reference implementation broadcasts a single middle-layer vector to all layers; per-layer independent steering is left to future work.

(iv) Sparse accumulators and baseline-clone overhead. The accumulator analysis in Sec. 5 treats K_{acc} as a sparse $S' \times d$ allocation with $S' \ll S$. The reference implementation allocates full-shape $S \times d$ accumulators and additionally stores a baseline cache clone of size M_{KV} to enable rollback. The asymptotic claim that memory is independent of the branching factor b (Proposition 1) is unchanged. The constant factor in the reference implementation is approximately 3.04–$3.06 \cdot M_{\text{KV}}$ (Track D measurement, Sec. A). Sparse-accumulator implementation, baseline-clone elimination, and corrected example numbers backed by direct VRAM measurement are left to future work.

(v) On the reversibility theorem. Theorem 1 is independent of the four distinctions above. It was empirically validated to numerical zero drift over 200 apply-reverse cycles on Llama 3.2-1B-Instruct (residual norm $= 0.00$, 2026-04-16 gate) and confirmed at the 3B scale (residual norm $= 0.00$, 2026-05-05 probe). A deeper validation at depth 10 over 1,700 apply-revert cycles on Llama 3.2-1B with active forward passes yields residual norm $\leq 1.5 \times 10^{-6}$, three orders of magnitude tighter than bf16 machine precision ($\epsilon_{\text{bf16}} \approx 3.9 \times 10^{-3}$). The headline algorithmic contribution is unaffected by any constant-factor revisions.

D Theorem 1 Proof Detail

Proof sketch for Theorem 1. The FP32 accumulator maintains the exact cumulative delta. The only quantization error occurs in the final cast from FP32 to bf16, which is bounded by ϵ_{bf16} times the accumulator magnitude. After complete reversal, $\mathbf{A}_{\text{final}} = \mathbf{0}$, so $\mathbf{K}_n = \mathbf{K}_0$ exactly (up to the single bf16 quantization of zero, which is exact). □ □

E Algorithm Detail: The Reversible Rollback Procedure

The computational bottleneck. Standard MCTS in continuous latent space requires evaluating multiple future branches simultaneously. Mutating the past KV cache (e.g., under PagedAttention) invalidates all dependent computations, requiring a full physical cache copy for each branch. For a model with L layers, sequence length S, and hidden dimension d, the KV cache occupies $M_{\text{KV}} = 2 \cdot L \cdot S \cdot d \cdot \text{sizeof(dtype)}$ bytes, and a standard MCTS with branching factor b and depth d requires $O(b^d \cdot M_{\text{KV}})$ working-set memory.

Step 1: Forward Mutation (Steer). Before evaluating a branch, mutate the KV tensors in place:

$$\mathbf{K}_t^{(\ell)} \leftarrow \mathbf{K}_t^{(\ell)} + \alpha_K \cdot \mathbf{d}_K^{(\ell)}, \ \mathbf{V}_t^{(\ell)} \leftarrow \mathbf{V}_t^{(\ell)} + \alpha_V \cdot \mathbf{d}_V^{(\ell)} \tag{5}$$

Step 2: Evaluation. Execute the forward pass, generate the next token, compute the per-step next-token distribution $p(\cdot \mid \mathbf{h}_t)$, and evaluate the MCTS node reward $\hat{r}(\mathbf{h}_t)$ from this distribution (Eq. 1). Latent-response measurements such as OEI (Eq. 10) are recorded alongside the reward evaluation as static monitoring signals; they do not enter the reward.

Step 3: Reverse Rollback (Restore). Invert the mutation exactly:

$$\mathbf{K}_t^{(\ell)} \leftarrow \mathbf{K}_t^{(\ell)} - \alpha_K \cdot \mathbf{d}_K^{(\ell)}, \ \mathbf{V}_t^{(\ell)} \leftarrow \mathbf{V}_t^{(\ell)} - \alpha_V \cdot \mathbf{d}_V^{(\ell)} \tag{6}$$

In the FP32-accumulator implementation (Eq. 2), Step 3 is implemented by decrementing the accumulator $\mathbf{A}^{(\ell)}$ in FP32 and recasting the bf16 cache from base $\mathbf{K}_{\text{base},t}^{(\ell)}$, eliminating bf16 epsilon-drift accumulation across rollback cycles.

Memory Complexity. The reversible algorithm reduces working-set memory from $O(b^d \cdot M_{\text{KV}})$ to $O(M_{\text{KV}} + d \cdot L \cdot S' \cdot d_{\text{model}} \cdot 4)$, where S' is the number of mutated token positions (typically $S' \ll S$). The branching factor b does not appear in the bound. The reference implementation's full-shape allocation gives a constant factor of approximately 3.04–$3.06 \cdot M_{\text{KV}}$ (Appendix C item (iv)).

F Telemetry Matrix Framework

We construct a telemetry matrix $\mathbf{T}_t \in \mathbb{R}^{2 \times L}$ at each generation step t across L transformer layers, combining two orthogonal monitoring channels.

Bottom-Up Channel: H-Neuron Activation Tracking. Following Gao et al. (2025) (Gao et al., 2025), we identify the set of hallucination-associated neurons $\mathcal{H} \subset \{1, \ldots, d_{\text{ffn}}\}$ for the target model. At each generation step t, we compute the aggregate H-Neuron stress signal:

$$\sigma_H^{(\ell)}(t) = \frac{1}{|\mathcal{H}|} \sum_{j \in \mathcal{H}} \text{ReLU}\left(a_j^{(\ell)}(t) - \bar{a}_j^{(\ell)}\right) \tag{7}$$

where $a_j^{(\ell)}(t)$ is the activation of neuron j at layer ℓ and step t, and $\bar{a}_j^{(\ell)}$ is its baseline activation computed over a calibration set of faithful generations.

Top-Down Channel: RepE Linear Artificial Tomography. We apply the Linear Artificial Tomography (LAT) procedure from Zou et al. (2023) (Zou et al., 2023a) to extract a linear probe for the honesty dimension ($\mathbf{w}_{\text{hon}}^{(\ell)}$). The RepE compliance signal at layer ℓ is:

$$\rho_R^{(\ell)}(t) = \mathbf{w}_{\text{hon}}^{(\ell)\top} \mathbf{h}_t^{(\ell)} \tag{8}$$

The telemetry matrix combines the two channels:

$$\mathbf{T}_t = \begin{bmatrix} \sigma_H^{(1)}(t) & \cdots & \sigma_H^{(L)}(t) \\ \rho_R^{(1)}(t) & \cdots & \rho_R^{(L)}(t) \end{bmatrix} \tag{9}$$

The joint signature characterizes the latent-response of a frozen model under inference-time KV-cache steering; the structural interpretation of divergence patterns between σ_H and ρ_R is given by Definition 1 and the two associated Hypotheses (Sec. 3.2), which decouple the geometric pathology from anthropomorphic claims about model intent.

G Measurement Pipeline

The Off-monitor Escape Index (OEI) is the fraction of the steering-induced change $\Delta\mathbf{h} = \mathbf{h}_{\text{steered}} - \mathbf{h}_{\text{base}}$ that lies in the null space of a linear monitor $\mathbf{P}$:

$$\text{OEI} = \frac{\|\text{Proj}_{\text{Null}(\mathbf{P})}(\mathbf{h}_{\text{steered}} - \mathbf{h}_{\text{base}})\|_2}{\|\mathbf{h}_{\text{steered}} - \mathbf{h}_{\text{base}}\|_2} \tag{10}$$

$\text{OEI} \in [0, 1]$: $\text{OEI} \to 1$ indicates the activation change is entirely orthogonal to the monitor direction (complete off-monitor escape); $\text{OEI} \to 0$ indicates the change is aligned with the monitored direction.

Any measurement of inference-time KV-cache perturbation, including OEI, requires that the steering direction $\mathbf{d}_K^{(\ell)}$ live in the same space as the cached key tensors. Representation-engineering probes are calibrated against residual-stream activations of dimension d_{model}, while the KV cache stores per-head representations of dimension kv_heads $\times$ d_{head} under grouped-query attention. For Llama 3.2-1B these are 2048 and 8×64 respectively; an explicit projection between probe space and cache space is required. In our reference implementation we identified, characterized, and resolved a measurement-pipeline failure mode in which the broadcast layer silently zeroed the steering delta on shape incompatibility, producing forward passes that were bit-identical across α. The fix projects probe directions through layer-specific key projection weights, $\mathbf{d}_K^{(\ell)} = \mathbf{W}_K^{(\ell)} \mathbf{d}_{\text{probe}}^{(\ell)}$, reshaped to per-head form. Two alternative approaches are also viable: intervening at the residual stream prior to the K/V projection and re-running the layer's forward computation, or calibrating probes directly in per-head cache space against contrastive cache states. Systematic comparison of the three is left to future work.

Under the projection-based fix, OEI varies monotonically with steering magnitude on Llama 3.2-1B-Instruct (results in Table 4 below). The high baseline value ($\sim$0.99) is geometrically expected: in $d_{\text{model}} = 2048$ with a rank-1 linear monitor, removing one direction from any perturbation leaves $\sim$99% of the perturbation's norm in the null space by default. This high-dimensional saturation is consistent with the regime Bailey et al. (Bailey et al., 2026) characterize when discussing why activation cosine similarity is hard to drive low in a behavior-preserving setting; the small but monotonic OEI spread we observe suggests the metric is responsive to genuine steering structure rather than producing a uniform default. Whether OEI under a richer monitor yields a more discriminative signal is a natural extension of the framework.

The bottom-up H-Neuron stress channel (σ_H) initially reported zero across all α. To distinguish whether the constancy reflected a calibration-regime mismatch (Gao et al., 2025) or a deeper measurement issue, we ran a controlled three-prompt comparison at fixed $\alpha = 1.0$, varying only the test prompt across hallucination-style, coercion-style, and neutral regimes. σ_H remained identically zero across all 17 layers in all three regimes. We additionally tested the per-neuron baseline-subtraction-and-ReLU formulation specified by Eq. 7; under this formulation σ_H likewise remained zero, ruling out per-neuron normalization as the cause. The constancy resolved when we adopted per-layer H-Neuron calibration: in our initial implementation a single set of top-K H-Neuron indices was identified at the deepest layer and applied uniformly across the layer stack as an initial approximation, while Eq. 7 as written specifies per-layer indices and per-layer baselines. With per-layer calibration in place, σ_H rises monotonically with α across the same alpha sweep (Table 4), with 14–15 of 17 layers firing. Both measurement-pipeline issues—the steering-projection issue addressed by the $\mathbf{W}_K$-projection fix and the bottom-up calibration issue addressed by per-layer H-Neuron identification—are therefore resolved in the implementation reported here.

α	0.1	0.5	1.0	2.0	5.0
OEI	0.991	0.991	0.992	0.992	0.995
ρ_R	-1.306	-1.258	-1.198	-1.080	-0.533
σ_H	0.228	0.237	0.247	0.267	0.330
TDS	0.084	0.080	0.076	0.068	0.040

Table 4: Per-α mean telemetry under inference-time KV-cache steering on Llama 3.2-1B-Instruct, with both measurement-pipeline issues resolved (steering projected through layer-specific $\mathbf{W}_K^{(\ell)}$ to per-head cache space; H-Neuron calibration applied per-layer). All four channels respond to steering: ρ_R varies substantially, σ_H rises monotonically with 14–15 of 17 layers firing, OEI shows the small monotonic variation expected in the high-dimensional saturation regime, and the cross-channel divergence TDS *decreases* monotonically—the opposite of the signature predicted by the monitor-driven Dimensional Escape hypothesis (Hypothesis 1). Reversibility holds end-to-end (residual_norm = 0).