TOPOLOGY BEFORE STEERING: LOCALISING AND GATING CYCLIC CONCEPTS WITH PERSISTENT HOMOLOGY

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models become increasingly widespread, the need for effective safeguards and methods for controlling model behavior has grown rapidly. One prominent approach is activation steering, which directly intervenes in a model's residual stream by adding a steering vector to encourage or suppress a target concept. Yet two fundamental questions arise before steering and have received comparatively little attention: where does the target concept reside within the prompt, and should we intervene on a given input at all? We study these questions for cyclic concepts, whose activations trace loop-like structures in representation space. We show that tools from Topological Data Analysis, particularly persistent homology, provide a principled way to address both problems. By analyzing activations across prompts, persistent homology identifies where a cyclic concept emerges and which tokens carry it. For new inputs, the same topological signal then determines whether the concept is present and whether steering should be activated, allowing the intervention to remain off when it is unnecessary. This yields a topology-aware approach to activation steering that jointly addresses both where and when to steer.

1 INTRODUCTION

Activation steering focuses on *how* to modify a language model's internal state, but two decisions come first: *where should the intervention act, and when should it fire?* Existing methods modify residual-stream activations at inference time through operations such as additive shifts or rotations, without updating the model weights (Turner et al., 2024; Panickssery et al., 2024; Vu & Nguyen, 2025). However, choosing an intervention does not establish that the target concept is causally represented at the chosen site, nor that every input should receive the edit. Intervening at the wrong site may have little effect on the target behavior, while intervening when the target concept is absent can unnecessarily perturb unrelated computation. A clock-specific steer, for example, should act where clock information can influence the answer while remaining silent on inputs unrelated to time. Thus, *where* and *when* to steer are part of the control problem, rather than implementation choices to be fixed by convention.

Contributions. We study these questions for *cyclic concepts*, including clock times, months, and compass headings. Such concepts have an intrinsic periodic structure, suggesting that their representations may organize into loops when activations are compared across prompts. Our central idea is to use the shape of the *unsteered* representation twice: first, to locate candidate sites where the concept is structurally represented; and second, to determine whether a new activation resembles the reference activations observed at such a site. Rather than proposing another steering direction, we develop a topology-aware mechanism that *localises* and *gates* an existing intervention.

For localisation, we construct prompts that vary the target concept and collect residual-stream activations at each candidate layer and token position. The activations at a fixed site form a point cloud with *one point per prompt*. We analyse each cloud using persistent homology, which tracks topological features as the distance scale varies (Edelsbrunner et al., 2002; Carlsson, 2009). For cyclic concepts, we focus on persistent loops, represented by first homology, H_1. Because finite random clouds can also contain loops, we compare the longest observed persistence interval with a null distribution matched in sample size and spread. The resulting persistence ratio is a label-free score for identifying structured candidate sites, without assuming in advance which activation directions encode the concept.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

For gating, we convert the recovered loop into a circular coordinate. We reject nearly constant, degenerate coordinates using a uniformity check on this circular parameter, then fit a two-dimensional read-out plane. Calibration activations define an acceptance region according to their in-plane radius and the fraction of their displacement captured by this plane. At inference time, the gate reads the *unsteered* activation and applies the existing intervention only when the activation falls within this calibrated region; otherwise, it leaves the state unchanged. Persistent homology is computed only once, offline. The online gate requires only an $O(d_h)$ projection per token, where d_h is the hidden width, making the decision cheap enough to evaluate inside the forward pass.

Our contributions are threefold:

1. **Topological localisation with causal validation.** We introduce an unsupervised topological criterion for identifying candidate concept-carrying sites and validate the selected locations through causal activation patching.
2. **A lightweight gate for existing interventions.** We convert the recovered topological structure into a cheap per-input admission test that decides whether an existing steering intervention should fire, without changing its steering direction.
3. **Specificity on unrelated text.** We show that the gate, evaluated at every position of four standard benchmarks, does not fire, so the accuracy cost of always-on steering is not incurred there. This is a specificity result (§4.3).

Empirically, the topological criterion localises causally effective intervention sites: patching the discovered onset span matches the oracle target-percept rate of 1.00, while downstream copies and single tokens yield 0.00, and the signal separates the concept from a matched-variance distractor that varies across prompts just as the concept does. On Qwen2.5-3B, Llama-3.1-8B, and Qwen2.5-7B (11 tasks, $n = 33$ task–model pairs), the persistence ratio orders donor-patch success (Spearman $r_s = 0.56$; the pairs share tasks across models, so we read this as an ordinal signal) and flags model-specific failures such as `hue` on Qwen2.5-3B. The gate fires on 0.80–0.96 of `dense_clock` target inputs and at most 0.03 of off-task inputs across three models, and exceeds a rate-matched random gate on target percept by $+0.18$ over the six well-posed cells; on `months` it fails. For clock steering on Qwen2.5-3B, always-on intervention reduces ARC-easy accuracy from 77% to 35% and degrades other benchmarks, whereas the gate fires on no benchmark token. Together, these results support a simple principle: *before deciding how to steer, first determine where to act and whether to act at all.*

Notation. Let M be a decoder-only Transformer with L layers and hidden width d_h, and let $h_t^{(\ell)}(p) \in \mathbb{R}^{d_h}$ denote the residual-stream activation of prompt p at layer ℓ and token position t. We call (ℓ, t) a *site*. Given prompts $\mathcal{P} = \{p_i\}_{i=1}^n$, the activations at a fixed site form the point cloud $X_t^{(\ell)} = \{h_t^{(\ell)}(p_i)\}_{i=1}^n$. For a scale $\epsilon \geq 0$, $\mathrm{VR}_\epsilon(X)$ denotes the Vietoris–Rips complex of a point cloud X, and $H_q(\mathrm{VR}_\epsilon(X))$ its q-th homology group. We write $\mathrm{Dgm}_q(X)$ for the corresponding persistence diagram; a feature born at b and dying at δ has persistence $\pi = \delta - b$. Our method uses H_1, whose persistent features represent loops. For steering, $\mathcal{D}_{\mathrm{pos}}$ and $\mathcal{D}_{\mathrm{neg}}$ are prompt sets of opposite polarity. Section 2 introduces these objects formally, and Section 3 shows how they are used for localisation and gating.

Related Work. We discuss related work in Appendix A.

2 Background

2.1 Residual-stream activations

Let M be a decoder-only Transformer with L layers and hidden width d_h. At token position t, the residual stream accumulates the updates produced by successive Transformer blocks. Writing $a_t^{(\ell)}$ and $m_t^{(\ell)}$ for the attention and MLP writes at layer ℓ, respectively, we denote the residual-stream state after the block by $h_t^{(\ell)} = h_t^{(\ell-1)} + a_t^{(\ell)} + m_t^{(\ell)} \in \mathbb{R}^{d_h}$. Here $h_t^{(0)}$ is the token embedding. The two updates in $h_t^{(\ell)}$ are applied sequentially: the MLP reads the post-attention state rather than $h_t^{(\ell-1)}$ directly. We refer to $h_t^{(\ell)}$ as the *residual-stream activation* at site (ℓ, t) (Elhage et al., 2021). This distinction is important: the *residual connection* is the identity skip that carries the incoming state forward, whereas the *residual stream* is the running hidden state itself. Figure 1 illustrates the terminology. For

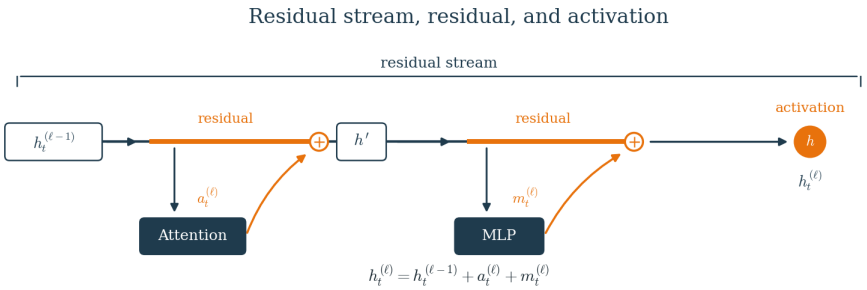

Figure 1: **Residual stream and activation.** At token position t, the residual stream carries the running hidden state through the Transformer block. Attention and the MLP write updates into this stream through residual connections. Our activation at site (ℓ, t) is the resulting state $h_t^{(\ell)} \in \mathbb{R}^{d_h}$.

a prompt p, write $h_t^{(\ell)}(p)$ for this activation. Given a family of prompts $\mathcal{P} = \{p_i\}_{i=1}^n$, fixing the site (ℓ, t) produces the *site cloud* $X_t^{(\ell)} = \left\{h_t^{(\ell)}(p_i)\right\}_{i=1}^n \subset \mathbb{R}^{d_h}$. Thus, each point in $X_t^{(\ell)}$ corresponds to a different prompt; the layer and token position are fixed. Throughout the paper, *points are prompts, not tokens*.

2.2 Activation steering

Activation steering intervenes directly on a residual-stream activation during the forward pass. A standard construction obtains a steering direction from contrasting two sets of prompts, $\mathcal{D}_{\text{pos}}$ and $\mathcal{D}_{\text{neg}}$: $v = \mathbb{E}_{p\sim\mathcal{D}_{\text{pos}}}\left[h_t^{(\ell)}(p)\right] - \mathbb{E}_{p\sim\mathcal{D}_{\text{neg}}}\left[h_t^{(\ell)}(p)\right], v \in \mathbb{R}^{d_h}$. Additive steering then replaces an activation x by $T_{\text{add}}(x) = x + \alpha v$ where α controls the intervention strength (Turner et al., 2024; Panickssery et al., 2024). Other interventions act differently on the same residual stream. Angular steering, for example, applies an orthogonal transformation $x \mapsto Rx$ within a selected subspace (Vu & Nguyen, 2025), while activation patching replaces an activation with a donor state (Vig et al., 2020; Meng et al., 2022; Engels et al., 2025). For our purposes, the important distinction is not whether an edit is called additive, rotational, or patching, but whether it preserves pairwise distances across a cloud. A common translation $x \mapsto x + \alpha v$ and a common orthogonal transformation $x \mapsto Rx$ are *isometries*: for all x, y, $\|T(x) - T(y)\|_2 = \|x - y\|_2$. By contrast, replacement patches, projections, and selectively gated edits need not preserve pairwise distances. This distinction becomes central in Proposition 3.1.

2.3 Persistent homology

Topological Data Analysis (TDA) describes the structure of a point cloud across spatial scales (Edelsbrunner et al., 2002; Carlsson, 2009). In our setting, the point cloud is $X_t^{(\ell)}$: different prompts observed at one fixed residual-stream site. For two activations $x_i, x_j \in X_t^{(\ell)}$, we use the Euclidean metric $d(x_i, x_j) = \|x_i - x_j\|_2$. We therefore compare *activations induced by different prompts*, not token positions within a prompt.

Vietoris–Rips filtration. For a scale $\epsilon \geq 0$, the Vietoris–Rips complex $\text{VR}_\epsilon(X)$ contains a simplex $\sigma \subseteq X$ whenever $\text{diam}(\sigma) = \max_{x,y\in\sigma} d(x, y) \leq \epsilon$. In particular, two vertices are joined by an edge when their distance is at most ϵ, and three vertices form a filled triangle when all three pairwise distances are at most ϵ. Because $\epsilon \leq \epsilon' \implies \text{VR}_\epsilon(X) \subseteq \text{VR}_{\epsilon'}(X)$, the complexes form a nested sequence, or *filtration*. Figures 2 and 10 in Appendix illustrate this construction.

Homology and persistence. At a fixed scale, the q-th homology group $H_q(\text{VR}_\epsilon(X))$ records q-dimensional holes in the complex. Its dimension, the Betti number β_q, counts independent such features: β_0 counts connected components and β_1 counts independent one-dimensional loops. Persistent homology follows these classes through the filtration. A class that appears at scale b and disappears at scale δ is represented by a birth–death pair (b, δ) with persistence $\pi = \delta - b$. The collection of birth–death pairs in degree q is the persistence diagram $\text{Dgm}_q(X)$. Long persistence indicates a feature that survives over a broad range of scales rather than one produced only by a narrow choice of ϵ. Persistence diagrams are stable to small perturbations of the underlying metric (Cohen-Steiner et al., 2005). We focus on H_1 because our target concepts are cyclic. A persistent H_1 class corresponds to a loop that closes before its interior fills. By contrast, H_0 persistence of a Vietoris–Rips filtration is equivalent to single-linkage clustering (Carlsson et al., 2010); it is therefore not the topological signal studied here. We compute persistent homology using `ripser` (Bauer,

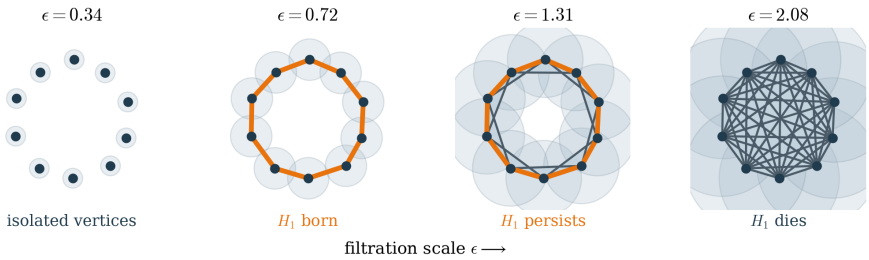

Figure 2: **A Vietoris–Rips filtration of a cyclic point cloud.** As ϵ increases, isolated vertices connect into a loop. The corresponding H_1 class is born when the loop closes, persists while its interior remains unfilled, and dies once higher-dimensional simplices fill the hole.

2021). Homology is evaluated over a finite field $\mathbb{Z}/p$; the choice of coefficient field and its role in constructing circular coordinates are detailed in Appendix D.

3 Persistent Homology for Localisation and Gating

Our method uses topology in two stages. First, an *offline localisation* procedure scans residual-stream sites and identifies where a cyclic concept forms a persistent loop across calibration prompts. Second, the detected loop is converted into a cheap *online gate* that determines whether a new activation resembles the calibrated concept manifold. Persistent homology is used only in the offline stage; inference requires only a low-dimensional projection.

3.1 Topological localisation

Aligned prompt clouds. Let $\mathcal{P} = \{p_i\}_{i=1}^n$ be calibration prompts constructed from a common template while varying the cyclic concept. For each site (ℓ, t) we form $X_t^{(\ell)} = \left\{h_t^{(\ell)}(p_i)\right\}_{i=1}^n \subset \mathbb{R}^{d_h}$. The token index t must denote the same linguistic slot across prompts. Otherwise, $X_t^{(\ell)}$ mixes representations from different roles and its topology has no consistent interpretation. We therefore use a common template and concept labels with matched tokenisation. Table 2 in Appendix illustrates the requirement.

A null-calibrated persistence score. We reduce each site cloud to 16 dimensions by PCA before computing persistent homology. Raw persistence is not directly comparable across clouds with different spread: even an unstructured cloud can produce a long finite H_1 bar. We therefore compare each observed cloud with a matched Gaussian null. More details about the Gaussian null are in L.

Let σ_j be the empirical standard deviation of coordinate j of the reduced cloud. A null cloud $\widetilde{X} \in \mathbb{R}^{n\times 16}$ is sampled independently as $\widetilde{X}_{ij} \sim \mathcal{N}(0, \sigma_j^2)$. For any point cloud X, define the longest finite H_1 persistence $\pi_{\max}(X) = \max_{(b,\delta)\in \mathrm{Dgm}_1(X),\, \delta<\infty}(\delta - b)$. From at least $R = 20$ independently sampled null clouds, let $q_{0.95}(X)$ denote the empirical 0.95 quantile of $\pi_{\max}(\widetilde{X})$. We score the site by $\rho\left(X_t^{(\ell)}\right) = \frac{\pi_{\max}\left(X_t^{(\ell)}\right)}{q_{0.95}\left(X_t^{(\ell)}\right)}$. Hence $\rho > 1$ means that the longest observed loop persists longer than the 95th percentile of loops generated by the matched null. It does *not* merely mean that an H_1 bar exists: finite random clouds typically contain such bars as well. Appendix J studies the sampling variability of this null calibration.

Selecting the intervention site. We evaluate $\rho(X_t^{(\ell)})$ across candidate layers and token positions using calibration prompts only; when $\pi_{\max} = 0$, as for a cloud of identical activations, we set $\rho = 0$. A site is a topological candidate when $\rho > 1$. When the concept mention spans multiple tokens, we score windows of the corresponding length and select the first window whose mean score exceeds the threshold. This identifies the *onset* of the topological signal rather than a later position that merely contains a downstream copy. On an aligned template with a causal decoder, all pre-mention positions are identical across prompts, so the onset necessarily lies at or after the first token at which prompts differ; the content of the token scan is in selecting among the positions that do vary (Appendix M). Layer selection is performed from the calibration activations alone: among candidate layers we take the one with the largest ρ at the mention token, with ties broken toward the shallower layer, before any behavioural steering outcome is evaluated.

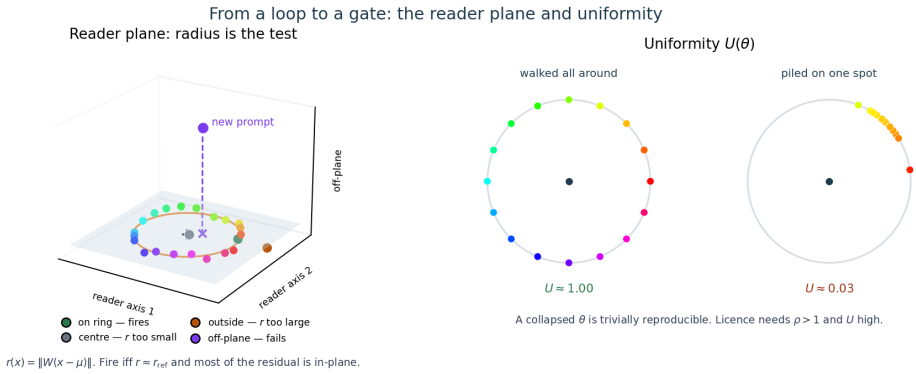

Figure 3: **Left.** The reader plane embeds this coordinate in a two-dimensional subspace of the residual stream and tests whether a new activation lies near the calibrated ring. **Right.** The circular coordinate must also pass the non-degeneracy check.

Scope. The localisation criterion is intentionally specific to cyclic structure. For a Vietoris–Rips filtration, H_0 reduces to single-linkage clustering (Carlsson et al., 2010); we therefore do not present H_0 as a distinct topological localisation mechanism. Moreover, ordered non-cyclic concepts are contractible and need not produce an H_1 class at all. Our method therefore targets genuinely cyclic concepts. We use linear concepts such as `honesty`, `sycophancy`, `sentiment`, and `linear_order` as negative controls.

3.2 From a persistent loop to a reader

A large persistence ratio establishes that a site contains a long-lived loop, but a gate also needs a way to place individual calibration activations *along* that loop. Persistent cohomology provides such a coordinate.

For a licensed H_1 class, we construct a circular coordinate (De Silva & Vejdemo-Johansson, 2009) $\theta : X_t^{(\ell)} \longrightarrow \mathbb{R}/\mathbb{Z}$. Each calibration activation receives an angle $\theta_i \in \mathbb{R}/\mathbb{Z}$. The coordinate is defined only up to a global rotation and orientation reversal: if θ is valid, then so are $\theta + c$ and $-\theta$. These gauge freedoms are irrelevant for the membership statistics below.

For intuition, take a 24-hour clock whose coordinate already tracks time. The gauge may place midnight near $\theta = 0$ and may run either way around the circle; it does not set the remaining hours. If the gaps are even, 06:00 then lies near $\theta = 1/4$, 12:00 near $\theta = 1/2$, and 18:00 near $\theta = 3/4$, or at the reflected positions $3/4$, $1/2$, and $1/4$. The origin and the orientation are arbitrary; the gaps between hours are not.

Rejecting degenerate coordinates. A persistent cocycle can yield a nearly constant circular coordinate, which is stable but uninformative. We therefore use

$$U(\theta) = 1 - \left| \frac{1}{n} \sum_{i=1}^{n} e^{2\pi \mathrm{i} \theta_i} \right| \in [0, 1]. \tag{1}$$

Here $U(\theta)$ is close to zero when the angles concentrate near a single location and is large when their first circular moment is small. It is invariant to both gauge freedoms above. We license a cycle only when the persistence criterion and this non-degeneracy check both pass.

Reader plane. The coordinate θ is initially defined only on the calibration cloud. To evaluate new activations, we regress the two-dimensional embedding $\big(\cos 2\pi\theta_i, \sin 2\pi\theta_i\big)$ on the centred hidden states by minimum-norm least squares. Since $n \leq 48 < d_h$ the system is underdetermined and the minimum-norm solution lies in the span of the centred calibration points. This yields a centre $\mu \in \mathbb{R}^{d_h}$ and a matrix $W \in \mathbb{R}^{2 \times d_h}$, whose rows we orthonormalise. For a new activation x, its reader coordinates are $c(x) = W(x - \mu)$. The pair (W, μ) defines the *reader plane*. It is used only as a sensor of whether an activation resembles the licensed cycle; it is not itself used as a steering direction. The basis of the plane is not identified across refits (Appendix K); the membership statistics below depend only on the subspace and on μ. Figure 3 and Figure 11 in Appendix summarise the construction.

3.3 SELECTIVE GATING

Localisation is an offline, cloud-level computation. At inference time, however, only one new activation is available. Recomputing persistent homology on that activation would be meaningless: a single point does not contain a loop. We therefore establish the cycle offline and use the reader plane as a per-input membership test.

Offline calibration. At the selected site, we require both $\rho(X_t^{(\ell)}) > 1$ and the circular-coordinate criterion in equation 1. If either check fails, no cycle is licensed and the method abstains. Otherwise, we fit (W, μ) and define the reference radius $r_{\text{ref}} = \text{median}_{x \in X_t^{(\ell)}} \|W(x - \mu)\|_2$.

Online membership test. For a new unsteered activation x, define $c = W(x - \mu), \quad r = \|c\|_2, \quad f = \frac{\|c\|_2}{\|x-\mu\|_2}$. The radius r measures where the projection lies relative to the calibration ring, while f is the fraction of the centred activation's norm that lies in the reader plane. The gate opens when $\left|\log \frac{r}{r_{\text{ref}}}\right| \leq \tau_r \quad$ and $\quad f \geq \tau_f$. The thresholds (τ_r, τ_f) are calibrated on the same reference cloud used to fit (W, μ), using a single target-coverage parameter (0.90 by default). This calibration is in-sample; the firing rates on held-out target inputs in Table 3 are the out-of-sample check. The resulting rule can be read simply as: *intervene only when the new activation resembles the cloud on which the cycle was licensed*. Once (W, μ) has been stored, this test costs $O(d_h)$ per token and requires no persistent-homology computation during generation.

The gate reads the *unsteered* activation: if it opens, the chosen intervention is applied; otherwise the activation is unchanged. Thus, the gate decides *whether* to steer, while the underlying method determines *how*.

Why topology is read before steering. Vietoris–Rips persistence is invariant under isometries, which determines the order of operations in our method.

Proposition 3.1. *Let $X \subset \mathbb{R}^{d_h}$ be finite and $T : \mathbb{R}^{d_h} \to \mathbb{R}^{d_h}$ an isometry. Then, for every q,*

$$\text{Dgm}_q(TX) = \text{Dgm}_q(X).$$

Hence common translations $x \mapsto x + \alpha v$ and orthogonal transformations $x \mapsto Rx$, $R^\top R = I$, preserve all Vietoris–Rips persistence diagrams.

The proof is in Appendix E. Thus, cloud-level persistence cannot verify a shared isometric edit, so our gate reads the *unsteered* activation before intervention. This does not preclude per-input verification, e.g., $c(x + \alpha v) = c(x) + \alpha W v$. Appendix K provides numerical checks and studies non-isometric interventions.

4 EXPERIMENTS

The same unsupervised pipeline runs in three settings. Language uses `Qwen2.5-3B-Instruct` (36 blocks, d_h=2048), `Llama-3.1-8B-Instruct` (32), `Gemma-2-9B-it` (42), `Qwen2.5-7B-Instruct` for the pooled correlation only, and `gemma-2-2b` only for the SAE comparison (bfloat16, greedy, left-padded; seeds randomise the null draws and the random-gate masks, not decoding). Cyclic prompts name the concept: `dense_clock` (48), `zh_clock` (48), `heading` (36), `hue` (24), `clock` (12), `months` (12), `emotions` (8), `weekdays` (7), `zh_weekdays` (7), `colors` (6), `seasons` (4), `compass` (4) (Figure 12, Appendix F). Linear controls `honesty`, `sycophancy`, `sentiment`, and `linear_order` are negative controls. They are not assumed to form a cycle. Tasks with $n \leq 8$ points are exploratory (Appendix M). The `smds_`$\star$ tasks of Table 3 are the temporal templates of Tiblias et al. (2025), imported unchanged and binned to a period. Splits are frozen. *Target percept* is the fraction of short answers that contain the steered label, i.e. the calibration label shifted around the cycle; once it saturates we also report *circular answer error*, with accuracy over n and over parsed outputs.

Vision is two tasks on one cycle, `dawn`→`midnight`. `vehicle_tod` is a VLM read on `Qwen2.5-VL-3B-Instruct`: one silhouette against changing skies, "what time of day?", and a state-fill of mean-pooled vision tokens, so the answer moves and the pixels stay. Controls are `vehicle_gray`, `sky_only`, and indoor `vehicle_indoor`; scoring matches language. The second, in §4.4, is gated Prompt-to-Prompt inside four-step `SDXL-Lightning`. A horizon cloud (desert, ocean, snow, prairie, pier $\times$ eight phases) licences one UNet norm and denoising step; the gate and the nearest phase choose whether to edit and which light words to append. Source cross-attention keeps the scene, and a closed gate returns the original image.

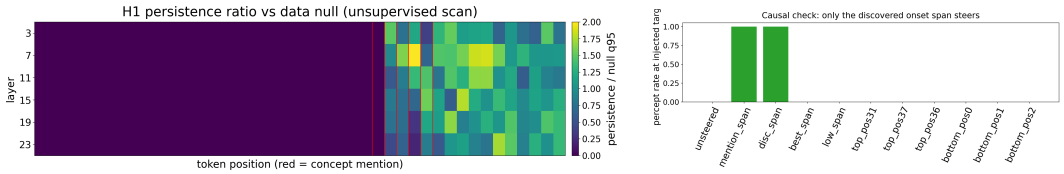

Figure 4: **Localisation.** Left: persistence-ratio scan over (position, layer); 0 at pre-mention positions by construction (identical activations), switching on at the mention. Right: the oracle mention and the onset span steer (1.000); downstream copies, a low-gate span and single tokens give 0.000. Pre-mention rows (`bottom_pos0-2`) are null rows on which donor and recipient coincide.

4.1 Topology localises the concept

Unsupervised localisation and causal validation. Pre-mention positions carry identical activations across aligned prompts, so the H_1 persistence-ratio scan is 0 there by construction; this checks that the hook and the alignment are correct. The informative comparison is among the positions that vary. Patching the discovered onset span yields percept 1.000, matching the oracle mention span, while the strongest-signal span (a downstream copy), an equal-size low-gate span, and the best-gated single token all yield 0.000 (Figure 4). The scan therefore selects the causal source over positions that inherit the loop through attention. The donor patch used here is the same operator as in the gate grid of §4.2.

Topology is not a variance detector. A matched-variance distractor separates this effect from activation magnitude: a random-name token and the concept have variances 1885 and 1837, but H_1 ratios 0.96 (below null) and 1.70 (above null), respectively. Correspondingly, donor patching at the name does not steer (the source label is retained on 0.96 of answers), whereas patching the concept reaches target percept 1.00 (Figure 5, left). Both slots vary across prompts, so this distinction is not available to a rule that selects the first token at which prompts differ.

A saturated probe has no dynamic range. Across 11 tasks at a fixed anchor layer on `Qwen2.5-3B`, a cross-validated linear probe reaches accuracy 1.000 on ten tasks and 0.998 on `dense_clock`. Its rank correlation with donor-patch success is therefore determined by tie handling and is not informative. The H_1 ratio reaches $r_s = +0.41$ ($p = 0.21$, not significant at $n = 11$) and activation variance $+0.11$ ($p = 0.76$). The distinction is between decodability, which the probe measures and which is near ceiling everywhere, and intervenability, which varies across these tasks (Figure 5, right).

The signal orders intervenability. Pooling 11 tasks across 3 models ($n = 33$ task–model pairs), the H_1 ratio orders donor-patch success ($r_s = +0.56$; Figure 6, left). The 33 pairs share tasks across models and are not independent, so we read the correlation as an ordinal signal rather than quoting a pooled p-value. A binary Youden cutoff fitted on `Qwen2.5-3B` (1.035) rejects one of the 22 held-out task–model pairs, and that pair steers; since $19/22$ held-out pairs steer, accepting every site (recall 1.0, precision 0.864) dominates the cutoff (recall 0.947, precision 0.857), and the cutoff carries no transfer information beyond the ordinal signal (Appendix M). The ordinal signal is model-dependent: `hue` fails on `Qwen` ($\rho = 0.68$, steer 0.49) but passes on `Llama` ($\rho = 1.32$, steer 0.99), a distinction that a fixed layer default cannot make.

Comparison with sparse autoencoders. On `gemma-2-2b` layer 12, persistence and Gemma-Scope SAE η^2 rank tokens similarly ($r_s = 0.68$) and activate only after the concept appears. Within its span, however, SAE scores saturate at ≈ 0.99 while persistence identifies a single peak. Persistence requires 13.3s of forward passes versus 0.51s for SAE encoding, but the latter requires a dictionary trained on $\approx$ 4B tokens and none is publicly available for `Qwen2.5-3B` (Cunningham et al., 2023).

Additional tests distinguishing internal intervention from prompt editing and H_1 from circle fitting are reported in Appendix H, Table 3.

4.2 A selective gate

Nine model–target cells (three cyclic targets $\times$ three models, five seeds each) rotate tasks between target and off-task inputs: the off-task inputs are the other two cyclic tasks' prompts together with `honesty` as a linear control, so they do not share the target's template. The injection layer is the $\arg\max \rho$ layer of §3.1, chosen before any behavioural evaluation. The gate decides per input from the unsteered activation at the mention token; when it fires, the same donor patch as in the always-on arm is applied to that input, so the two arms differ only in which inputs are written. On `dense_clock` the gate fires on $0.854/0.958/0.796$ of target inputs and $0.000/0.028/0.000$ of

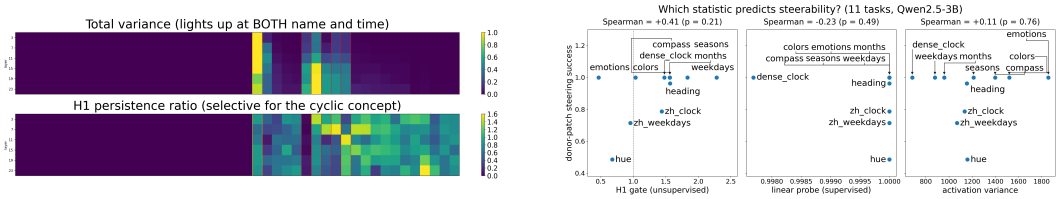

Figure 5: **Controls.** Left: a matched-variance distractor (1885 vs. 1837) is separated by H_1 (0.96 vs. 1.70), and only the topological site steers. Right: H_1, linear-probe, and variance predictors of steering success; the probe saturates on all 11 tasks.

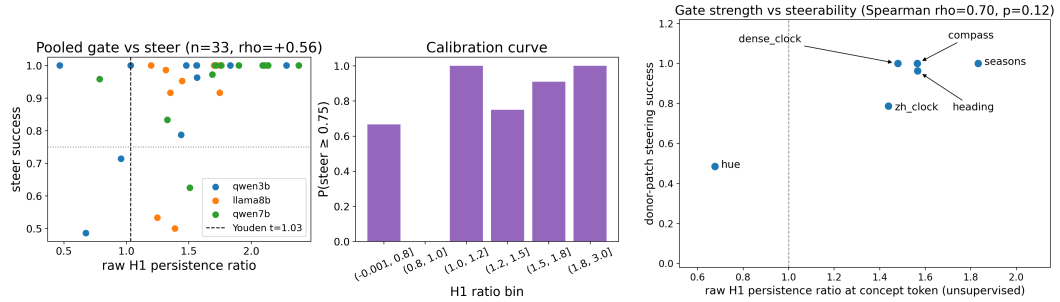

Figure 6: **Calibration and predictive validity.** Left: $P(\text{steer} \mid \text{ratio bin})$ over $n = 33$ task–model pairs. Right: H_1 ratio against donor-patch success.

Table 1: Capability under additive steering, `Qwen2.5-3B`, $\alpha = 16$, 100 examples per task (standard error roughly 0.04–0.05 per cell). `gsm8k` scored 0.000 in every arm (answer extraction fails at 0-shot) and is excluded from the mean.

Condition	**fire rate**	ARC-e	HellaSwag	TruthfulQA	WinoGrande	**mean Δ**
unsteered	0.000	0.770	0.680	0.532	0.730	—
always_on	1.000	0.350	0.480	0.502	0.560	**−0.205**
gated	**0.000**	**0.770**	**0.680**	**0.532**	**0.730**	**+0.000**
random_gated	0.012	0.750	0.670	0.553	0.740	+0.000

off-task inputs (`Gemma`/`Llama`/`Qwen`); on `months` two of the three cells invert (Figure 7). On held-out `dense_clock`, firing is 0.806 on unseen targets and 0 on `honesty` and `weekdays`. The three `weekdays` cells are ill-posed as a steering target because the unsteered model already produces the shifted label (percept 1.000), and they are excluded from pooled comparisons (Appendix M). Over the six well-posed cells, target percept is 0.611 always-on, 0.340 gated and 0.164 for a random gate at the same overall firing rate; the gate's margin over the random gate (+0.176 on average, ahead in 4/6 cells) is its selectivity, and it retains 0.56 of the always-on effect, below our pre-registered 0.90. Persistence ratios at the mention site are 1.44, 1.87, and 1.14 for `dense_clock`, `weekdays`, and `honesty`; all are licensed there but rejected at the last token, where additive steering on `dense_clock` scores 0. Thus, site licensing and gate firing are distinct; Appendix J measures cycle coverage, and Table 3 reports all cells.

4.3 The gate does not fire on unrelated text

Additive steering vectors degrade unrelated capabilities. We measure this effect on four standard benchmarks using an in-process harness (Gao et al., 2023) with active forward hooks; `gsm8k` was also run, but answer extraction fails at 0-shot in every arm, so it is excluded from the mean. To avoid spurious null effects, we steer *all* sequence positions; a last-position hook would often modify only padding. We also choose the steering magnitude from the always-on damage curve alone, independent of the gate. At $\alpha = 16$ (Table 1, Figure 8), always-on steering reduces mean accuracy by $\mathbf{0.205}$. The gate, evaluated at every position of every input with no information about where a concept might be, fires on none of the scored tokens, so the gated arm is the unsteered model and its $\Delta = 0$ follows. This is a specificity result: the gate has no false positives on unrelated text. It does not by itself separate the topological gate from any other low-rate gate, since a random gate at 1.2% is equally harmless here, and with 100 examples per benchmark the standard error per cell is roughly 0.04–0.05, so only the always-on drops on ARC-e and HellaSwag are individually resolved. Our gate however, gives us control that a random gate does not.

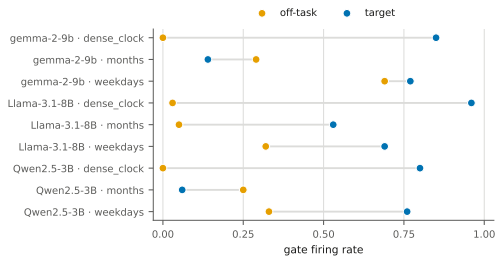

Figure 7: **Gate selectivity.** Firing on target vs. off-task inputs per model×task cell. `dense_clock` is selective on all three models (best case 0.96 vs. 0.03); `months` inverts on two; the `weekdays` cells are ill-posed as steering targets (Appendix M).

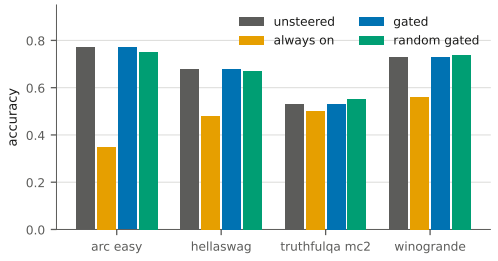

Figure 8: **Specificity on benchmark text.** Always-on steering costs 20.5 accuracy points over the four scored benchmarks; the gate fires on no benchmark token, so the gated arm equals the unsteered model. A random gate at 1.2% is also harmless at this rate (§I).



Figure 9: **Qualitative text-to-image demonstration.** Left to right: Michigan pier from day to sunset, rocky coastline, and snow, each as day then dawn. These scenes contain no vehicle; the vehicle appears only in the VLM read. If the gate does not open, the image is returned pixel-identical.

4.4 A CONCRETE APPLICATION OF OUR METHOD WITH VISION STEERING

The language tasks name the concept in the prompt, so a lexical detector could also gate them; vision requires an activation-level gate because no string marks the time of day. We therefore run the same unsupervised pipeline on an eight-phase time-of-day cycle (`dawn`→`midnight`) in two settings, one quantitative and one qualitative. For the **VLM read**, `vehicle_tod` uses `Qwen2.5-VL-3B-Instruct` on a composited vehicle silhouette against a changing sky with the fixed question "what time of day?". We state-fill the mean-pooled vision tokens with a donor phase, changing the model's answer while leaving the image unchanged. Flat lighting (`vehicle_gray`), sky-only, and indoor (`vehicle_indoor`) images serve as controls. The gate fires on 0.800 of target images and 0.000 of controls, with target percept 0.250 versus 0.375 always-on and 0.175 for rate-matched random gating (Table 3). For the **pixel edit**, `vehicle_tod_t2i` runs inside a four-step `SDXL-Lightning` UNet at CFG 0. An unsupervised H_1 scan over module×denoising step licenses one read site before Prompt-to-Prompt (Hertz et al., 2022): pass 1 stores source cross-attention maps; pass 2 copies them onto shared prefix tokens at steps 0–1, while steps 2–3 use the edited prompt's own attention; self-attention is unchanged. At inference, a closed gate leaves the image pixel-identical; an open gate assigns the nearest calibration phase and appends its lighting phrase before Prompt-to-Prompt runs. Figure 9 shows three outdoor scenes. The painter is prompt-level; topology supplies only the site scan and gate, so the demonstration is qualitative.

5 CONCLUSION

Activation steering asks what to add; we address two prior questions: where the concept is causally read and whether to intervene. A null-calibrated H_1 persistence ratio selects the causal source over downstream copies and over a matched-variance distractor, orders donor-patch success across three models ($r_s = 0.56$, $n = 33$ task–model pairs). The resulting $O(d_h)$ gate is selective on `dense_clock` across three models, exceeds a rate-matched random gate on target percept by $+0.18$ over the six well-posed cells, fails on `months`, and does not fire on unrelated benchmark text, where always-on steering loses 20.5 points. The framework wraps an existing intervention and transfers to a vision-language read.

Our main *limitation* is scope: H_1 applies to templated cyclic concepts, while some gating cells fail. *Future work* should extend topology to non-cyclic concepts and topology-derived steering directions.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on our experiments and implementing methods. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology, and interpreting results, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for modifying scientific figures, suggesting experimental parameters, editing software code, editing our research paper to improve readability, and identifying relevant literature. We have reviewed all AI-assisted work. In particular, we discussed and evaluated within our group the AI-generated suggestions for experimental design. We also carefully assessed AI suggestions for method implementation. In addition, we reviewed all AI-generated or AI-edited figures and manuscript revisions, and carefully read and verified the literature suggested by AI. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Given the nature of the work, we do not foresee negative societal or ethical impacts. Risks remain: by making an existing steering intervention more selective, our method may lower the barrier to applying it, including for harmful content such as persuasive misinformation or biased narratives. Although it does not expand the baseline risk profile of the underlying models, robust safeguards, transparency, accountability, and ongoing ethical review are required for responsible use.

REPRODUCIBILITY STATEMENT

Source codes for our experiments are provided in the supplementary materials of the paper. The details of our experimental settings are given in Section 4 and the Appendix. The datasets that we used in the paper are published.

REFERENCES

Henry Adams, Sofya Chepushtanova, Tegan Emerson, Eric Hanson, Michael Kirby, Francis Motta, Rachel Neville, Chris Peterson, Patrick Shipman, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology, 2016. URL https://arxiv.org/abs/1507.06217.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Ulrich Bauer. Ripser: efficient computation of vietoris–rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.

Rickard Brüel-Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J. Guibas, and Gunnar Carlsson. A topology layer for machine learning, 2020. URL https://arxiv.org/abs/1905.12200.

Peter Bubenik. Statistical topological data analysis using persistence landscapes, 2015. URL https://arxiv.org/abs/1207.6437.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

Gunnar E Carlsson, Facundo Mémoli, et al. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.*, 11(Apr):1425–1470, 2010.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pp. 263–271, 2005.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models, 2023. *URL https://arxiv.org/abs/2309.08600*, 2309, 2023.

Vin De Silva and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pp. 227–236, 2009.

Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & computational geometry*, 28(4):511–533, 2002.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Josh Engels, Eric Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *International Conference on Learning Representations*, volume 2025, pp. 84591–84622, 2025.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, 2022.

Wes Gurnee and Max Tegmark. Language models represent space and time. In *International Conference on Learning Representations*, volume 2024, pp. 2483–2503, 2024.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36:17643–17668, 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022. URL https://arxiv.org/abs/2208.01626.

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)*, pp. 2733–2743, 2019.

Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures, 2018. URL https://arxiv.org/abs/1707.04041.

Subhash Kantamneni and Max Tegmark. Language models use trigonometry to do addition. *arXiv preprint arXiv:2502.00873*, 2025.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International conference on learning representations*, volume 2025, pp. 90960–90985, 2025.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024. URL https://arxiv.org/abs/2408.05147.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *International Conference on Learning Representations*, volume 2024, pp. 26486–26515, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition, 2024. URL https://arxiv.org/abs/2312.06681.

Erik Rybakken, Nils Baas, and Benjamin Dunn. Decoding of neural data using cohomological feature extraction. *Neural computation*, 31(1):68–93, 2019.

Federico Tiblias, Irina Bigoulaeva, Jingcheng Niu, Simone Balloccu, and Iryna Gurevych. Hypothesis-driven feature manifold analysis in llms via supervised multi-dimensional scaling. *arXiv preprint arXiv:2510.01025*, 2025.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Hieu M Vu and Tan M Nguyen. Angular steering: Behavior control via rotation in activation space. *arXiv preprint arXiv:2510.26243*, 2025.

Weixuan Wang, Jingyuan Yang, and Wei Peng. Semantics-adaptive activation intervention for llms via dynamic steering vectors. In *International Conference on Learning Representations*, volume 2025, pp. 79334–79351, 2025.

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, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL https://arxiv.org/abs/2310.01405.

Supplement to "Topology Before Steering: Localising and Gating Cyclic Concepts with Persistent Homology"

Table of Contents

A Expanded related work **14**

B Topological Data Analysis supplementary figure **14**

C Localisation details **16**

D Coefficient fields for persistent homology **17**

E Proof of Proposition 3.1 **17**

F Benchmark examples **17**

G Experimental detail **18**

G.1 Spearman coefficient and Youden cutoff 18

G.2 Construction of the gate 18

G.3 Additional details on localisation experiments 19

H Extra Experiments **20**

I Some limitations of our gating method **22**

I.1 Hyperparameters 23

I.2 Controls that changed a conclusion 23

J Measurement notes **23**

K Supplementary results **24**

L Data-dependent, independent and Gaussian null. **25**

M Further clarifications **25**

N Reproducibility **27**

A Expanded related work

Steering. Activation steering modifies internal representations during inference, providing an efficient alternative to fine-tuning. Early work showed that static vectors injected into the residual stream can steer sentiment and alignment (Turner et al., 2024; Zou et al., 2025; Arditi et al., 2024). Contrastive Activation Addition (CAA) instead derives steering vectors from mean activation differences between paired positive and negative inputs (Panickssery et al., 2024). Recent methods go beyond additive translations: Angular Steering applies rotations within activation subspaces (Vu & Nguyen, 2025).

Conditional and localised intervention. Conditional Activation Steering (Lee et al., 2025) gates a steer with a supervised condition vector and a cosine threshold and is the closest antecedent to our *when* question; SADI (Wang et al., 2025) adapts the intervention per input. ITI (Li et al., 2023) selects intervention heads by probe accuracy. Hase et al. (2023) show that causal localisation by patching does not predict where an edit succeeds, and Makelov et al. (2024) document interpretability illusions in subspace patching; both bear on the gap between reading and writing at a site that we observe in Appendix K. Relative to CAST, our gate requires neither the cyclic order of the labels nor negative examples of the condition: it takes a calibration cloud and returns whether a loop exists and a coordinate along it.

Localisation and attribution. Activation patching is the standard causal-attribution tool (Vig et al., 2020; Meng et al., 2022). Engels et al. (2025) show some LLM features are genuinely circular and introduce the concept-token replacement protocol we use for our patches; Gurnee & Tegmark (2024) and Kantamneni & Tegmark (2025) report related structured concept geometry in language models. Sparse autoencoders give an alternative route to localisation (Lieberum et al., 2024). Supervised probes with control tasks (Hewitt & Liang, 2019) measure decodability at a controlled capacity, a property distinct from the intervenability studied here.

Topological Data Analysis and Persistent Homology. TDA characterizes the shape of high-dimensional data, with persistent homology tracking topological features across scales (Edelsbrunner et al., 2002; Carlsson, 2009). Persistence diagrams record their births and deaths and are stable to small perturbations (Cohen-Steiner et al., 2005). Persistent cohomology and circular coordinates (De Silva & Vejdemo-Johansson, 2009) have been used to decode neural population activity (Rybakken et al., 2019; Gardner et al., 2022); we use the same coordinate but score sites by a null-calibrated persistence ratio and convert the coordinate into an inference-time gate rather than a decoder. H_0 of a Vietoris–Rips filtration is single-linkage clustering (Carlsson et al., 2010), so we make no H_0-based claims.

Topological Machine Learning. TDA was integrated with neural networks through stable representations such as Persistence Landscapes (Bubenik, 2015) and Persistence Images (Adams et al., 2016), followed by differentiable topological layers (Hofer et al., 2018; Brüel-Gabrielsson et al., 2020). However, simplex growth and persistent-homology reductions make training-time topological constraints expensive at scale. We keep persistent homology offline, on the calibration cloud, and use only the resulting reader plane as an inference-time gate.

B Topological Data Analysis supplementary figure

In this section we introduce the definitions of shape, scale and what closing in means in TDA terms. Visual understanding is provided through Figure 10.

Definition B.1. Let d be the Euclidean distance on the residual stream, restricted to the finite point cloud $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^{d_h}$. The ***shape*** of X is the persistent homology of its Vietoris–Rips filtration: the family of vector spaces

$$\big\{\, H_q\big(\mathrm{VR}_\epsilon(X)\big)\big\}_{q\geq 0,\ \epsilon\geq 0},$$

together with the linear maps $H_q(\mathrm{VR}_\epsilon(X)) \to H_q(\mathrm{VR}_{\epsilon'}(X))$ induced by the inclusions $\epsilon \leq \epsilon'$. Equivalently, it is the collection of persistence diagrams $\mathrm{Dgm}_q(X)$. This object is a function of the pairwise distances $\{d(x_i, x_j)\}$ alone: it is unchanged by relabeling the prompts and by any isometry of $\mathbb{R}^{d_h}$, so it does not depend on a choice of axes.

Definition B.2. A ***scale*** is a real parameter $\epsilon \geq 0$. It is the only threshold in the filtration: $\mathrm{VR}_\epsilon(X)$ contains a simplex $\sigma \subseteq X$ if and only if $\mathrm{diam}(\sigma) = \max_{x,y\in\sigma} d(x, y) \leq \epsilon$. Growing the scale means increasing ϵ. Because $\epsilon \leq \epsilon'$ implies $\mathrm{diam}(\sigma) \leq \epsilon \Rightarrow \mathrm{diam}(\sigma) \leq \epsilon'$, one has the nested family $\mathrm{VR}_\epsilon(X) \subseteq \mathrm{VR}_{\epsilon'}(X)$.

Definition B.3. Two points are ***close at scale*** ϵ when $d(x_i, x_j) \leq \epsilon$; the edge $[x_i, x_j]$ is then a 1-simplex of $\mathrm{VR}_\epsilon(X)$. A finite sequence of distinct points $x_{i_0}, \ldots, x_{i_{m-1}}$ with $m \geq 3$ has ***closed***

into a loop at scale ϵ when every consecutive pair, including the pair that returns from the last point to the first, is close at that scale. The corresponding 1-chain $\gamma = \sum_{k=0}^{m-1} [x_{i_k}, x_{i_{k+1}}] \in C_1(\mathrm{VR}_\epsilon(X))$ (indices modulo m) then has vanishing boundary, $\partial_1 \gamma = 0$, so γ is a 1-cycle. The edges close, and the loop is born, at the scale $b(\gamma) = \max_{0 \le k < m} d(x_{i_k}, x_{i_{k+1}})$, the moment the last edge appears. The cycle is an H_1 hole at scales $\epsilon \in [b(\gamma), \delta(\gamma))$ for which γ is not the boundary of any 2-chain in $\mathrm{VR}_\epsilon(X)$; it dies at the smallest $\delta(\gamma)$ where such a filling exists.

Furthermore, one has to notice that H_0 of a Vietoris–Rips filtration is exactly the single-linkage dendrogram (Carlsson et al., 2010), so an H_0-based scan is a clustering method and must be benchmarked as one. This is why we focused on H_1 cycles in this paper.

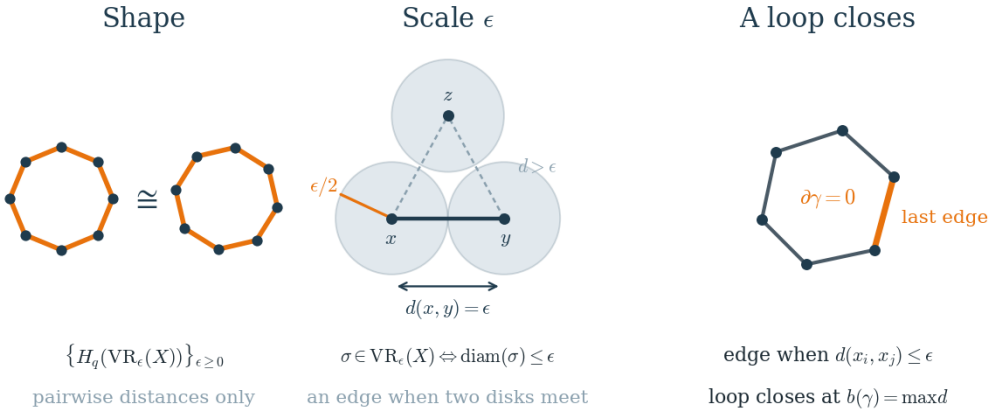

Figure 10: **Basic TDA objects used in this paper.**
A point cloud is studied through pairwise distances. At scale ϵ, points within distance ϵ are connected; when edges close around an unfilled region they form a 1-cycle. Increasing ϵ reveals which structures persist across scales.

C LOCALISATION DETAILS

A site (ℓ, t) is a meaningful object of comparison only when the index t names the same linguistic slot in every calibration prompt. Table 2 makes the requirement concrete. In the unaligned pair the clock mention sits at $t = 3$ under template A and at $t = 4$ under template B, so the cloud $X_t^{(\ell)}$ mixes a concept token with a function word and its homology is not a statement about the concept. The aligned template removes that drift: every column is linguistically fixed, and only the bold mention changes, from `00:00` to `00:30`. Once such a cloud is formed, Figure 11 is the passage from a detected loop to a coordinate that a gate can read. On the left, a persistence ratio $\rho > 1$ licenses the site: the prompts trace a loop, which persistent homology records as an empty interior rather than as a property of any single activation. On the right, persistent cohomology places each calibration point on that loop by a circular coordinate $\theta \in \mathbb{R}/\mathbb{Z}$, drawn as colour. The numerical origin and the orientation are gauge freedoms. Rotating the colours, or reversing them so that θ becomes $-\theta$, describes the same cycle. And what the construction retains is the cyclic order of the prompts.

Table 2: **Token alignment across prompts.** A site (ℓ, t) is comparable only when t denotes the same linguistic slot across the calibration family. Bold marks the concept mention.

	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$
Unaligned templates						
A	The	time	is	**00:00**	.	Thirty
B	If	the	clock	shows	**00:00**	,
Aligned template						
`00:00`	The	time	is	**00:00**	.	Thirty
`00:30`	The	time	is	**00:30**	.	Thirty

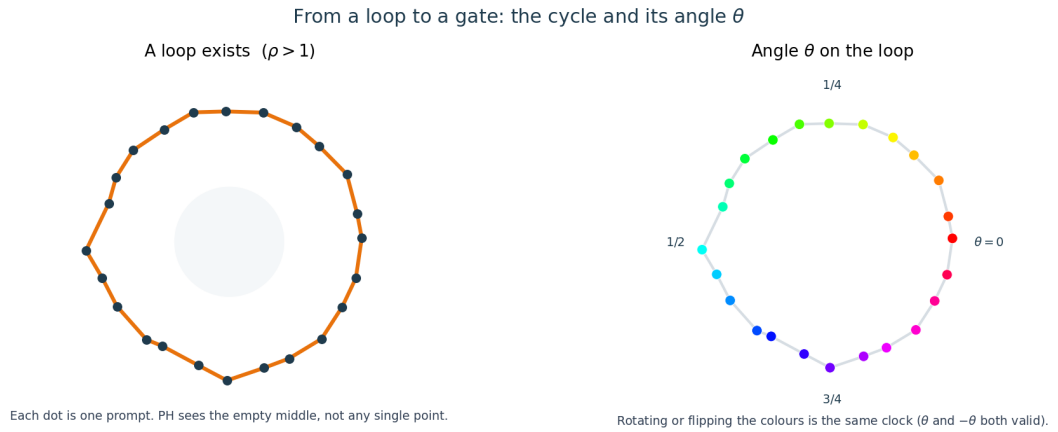

Figure 11: **From a persistent loop to an inference-time gate. Left.** A site is first licensed by its persistent H_1 structure. **Right.** Persistent cohomology assigns calibration points a circular coordinate θ.

D COEFFICIENT FIELDS FOR PERSISTENT HOMOLOGY

Persistent homology is not a geometric heuristic. At each scale ϵ of the Vietoris–Rips filtration it is ordinary linear algebra on the simplices of $\mathrm{VR}_\epsilon(X)$, and the entries of those linear maps are taken from a chosen coefficient field. This appendix records what that means and why we use $\mathbb{Z}/p$.

Chains, boundaries, homology. Fix ϵ and write C_q for the abelian group of finite formal sums of q-simplices of $\mathrm{VR}_\epsilon(X)$. An element of C_1 is a *1-chain*: a weighted sum of edges $c = \sum_e a_e\, e$, with weights a_e in the coefficient ring. The *boundary* operator $\partial_q \colon C_q \to C_{q-1}$ is the linear map that sends a simplex to the alternating sum of its faces (an edge to its two endpoints, a triangle to its three edges). It satisfies $\partial_{q-1} \circ \partial_q = 0$. A chain with $\partial c = 0$ is a *cycle* (for $q = 1$: a closed loop of edges). A cycle is a *boundary* if it is ∂ of a $(q+1)$-chain (for $q = 1$: the loop is the rim of a filled region). Homology in degree q is the quotient

$$H_q(\mathrm{VR}_\epsilon(X)) \;=\; \frac{\ker \partial_q}{\operatorname{im} \partial_{q+1}} \;=\; \frac{\text{cycles}}{\text{cycles that bound}}. \tag{2}$$

Computing H_q is computing the kernel and image of two matrices ∂_q and ∂_{q+1}. Persistent homology repeats this as ϵ grows and records, as a birth–death pair (b, δ), the scale at which a class enters the kernel and the scale at which it later falls into the image. That is the persistence diagram Dgm_q.

Why a field. The weights a_e must live in some ring. If that ring is a *field* $\mathbb{F}$: a number system with addition, subtraction, multiplication and division except by zero. Then each C_q is an $\mathbb{F}$-vector space, ∂_q is an $\mathbb{F}$-linear map, and H_q is an $\mathbb{F}$-vector space. Its dimension

$$\beta_q \;=\; \dim_{\mathbb{F}} H_q \tag{3}$$

is the q-th *Betti number*: β_0 counts connected components, β_1 counts independent unfilled loops. Over a ring that is not a field (notably $\mathbb{Z}$) homology is a module, Betti numbers are not a complete invariant, and the computation requires Smith normal form rather than Gaussian elimination. Software therefore computes over a field.

Why $\mathbb{Z}/p$. The field we use is $\mathbb{Z}/p$, the integers modulo a prime p, also written $\mathbb{F}_p$. It is a field if and only if p is prime. Arithmetic is exact (no floating-point), the matrices are small, and it is the coefficient convention of `ripser` (Bauer, 2021), whose `coeff` argument is that prime. Two values appear in this paper.

Over $\mathbb{Z}/2$ an edge is simply in a cycle or not: weights are $\{0, 1\}$ with $1 + 1 = 0$, so there is no orientation. This is the default for persistence diagrams when only birth–death pairs are required.

Circular coordinates need more. An H_1 class is represented by a *cocycle* with values in $\mathbb{Z}/p$; following De Silva & Vejdemo-Johansson (2009) we lift those values to $\mathbb{Z}$ (sending $k \mapsto k$ if $k \le p/2$ and $k \mapsto k - p$ otherwise) and solve a least-squares smoothing problem to obtain $\theta \colon X \to \mathbb{R}/\mathbb{Z}$.

A larger prime makes the lift unambiguous on typical clouds. We use $p = 47$ as the primary field and check that θ is stable across $\{11, 47, 97\}$. The prime is a computational parameter, not a quantity estimated from the activations.

E PROOF OF PROPOSITION 3.1

Proof. For every $x_i, x_j \in X$,

$$\|T(x_i) - T(x_j)\|_2 = \|x_i - x_j\|_2.$$

Hence a subset of vertices has diameter at most ϵ before the transformation if and only if it has diameter at most ϵ after the transformation. Therefore $\mathrm{VR}_\epsilon(TX)$ and $\mathrm{VR}_\epsilon(X)$ are isomorphic at every ϵ, with identical inclusion maps across the filtration. Their persistent homology, and hence their persistence diagrams, are identical. □

F BENCHMARK EXAMPLES

Here we provide a clearer view on our benchmarks. Figure 12 shows one prompt per cyclic task: green is the winding label, red the continuation, and the last column the period, with Mandarin rows as alternate surfaces of the same cycles and vision split into a VLM read and a text-to-image write.

Cyclic concepts and representative prompts

Colours: **14:30** cyclic mention **the next day is** continuation **truck** identity **desert highway** distractor

Black text is fixed template. zh_* are Mandarin surfaces of the same cycle.

Vision is two tasks: a VLM read (the answer changes) and a T2I write (the pixels change).

Task	Context	Continuation	Range
Language			
dense_clock	The time is **14:30**.	**Thirty minutes later it is**	00:00–23:30 **(48)**
zh_clock	现在是十四点半。	半小时后是	0点–23点半 **(48)**
heading	The heading is **40** degrees.	**Ten degrees clockwise is**	0°–350° **(36)**
hue	The hue angle is **120** degrees.	One 15° clockwise step is	0°–345° **(24)**
clock	It is **3** o'clock.	**In one hour it will be**	12, 1, ⋯, 11 **(12)**
months	The current month is **March**.	**The next month is**	Jan–Dec **(12)**
emotions	I feel **joy**.	**One step clockwise is**	Plutchik **(8)**
weekdays	Today is **Monday**.	**The next day is**	Mon – Sun **(7)**
zh_weekdays	今天是星期一。	下一天是	星期一 – 日 **(7)**
colors	Current hue is **red**.	**The next colour is**	rainbow order **(6)**
seasons	It is **spring**.	**The next season is**	spring–winter **(4)**
compass	Facing **north**.	A 90° clockwise turn faces	N, E, S, W **(4)**
Vision — two tasks			
vehicle_tod VLM	[image: **truck** at **sunset**]	**What time of day is shown?**	dawn–midnight **(8)**
vehicle_tod_t2i T2I	a **truck** on a **desert highway** at **dawn**	**eval caption omits the phase**	dawn–midnight **(8)**

Figure 12: **Cyclic concepts.** One representative prompt per task. Green is the winding label, red the continuation (or eval query). Vision is two tasks: a VLM read that changes the answer, and a T2I write that changes the pixels; blue is vehicle identity, orange the scene distractor. `zh_clock` and `zh_weekdays` are Mandarin surfaces of `dense_clock` and `weekdays`. Linear controls are omitted.

G Experimental detail

G.1 Spearman coefficient and Youden cutoff

Spearman r_s is a rank correlation (we used ρ for the persistence ratio): if we sort tasks (or sites) by one score, how well does that match the order of another? Here, $+1$ would mean a higher H_1 ratio always goes with higher donor-patch success, 0 means the two rankings are unrelated, and -1 would mean they run backwards. We use it to ask whether the topological score *orders* intervenability, not whether it predicts the exact percept value.

Youden's J (sensitivity $+$ specificity -1) picks a single cutoff that best splits successes from failures. We fit that cutoff on `Qwen2.5-3B` (ratio 1.035) and apply it to the 22 held-out task–model pairs, where it gives 18 true positives, 1 false negative, 3 false positives and 0 true negatives, i.e. $J = -0.053$. Accepting every site (recall 1.0, precision $19/22 = 0.864$) dominates it, so the cutoff has no discriminative content on the held-out models (§M); it describes the `Qwen2.5-3B` operating point only.

G.2 Construction of the gate

The main procedure to construct the gate on the theoretical side has been explained in 3.3. We have an extra evaluation harness that reorders and re-batches requests, so a mask built in advance and aligned by row both misaligns and fails on ragged batches. Deciding in the hook also keeps the capability

test of §4.3 honest: there the gate sees only the activation at each position, and it is told nothing about where a concept might be. In the gate grid of §4.2 the gate reads the mention-token activation of each input, so it is told the position but not the label. Proposition 3.1 fixes which activation must be read. A data-independent edit leaves the persistence diagram unchanged, so the edit cannot be checked after it has been applied. The gate therefore reads the *unsteered* activation and decides before the write.

G.3 Additional details on localisation experiments

This section gives the additional details behind the localisation experiments in Section 4.1. The main text reports the results; here we describe the controls and comparisons in more detail.

Localisation and causal patching. The localisation scan is entirely unsupervised. At every candidate (position, layer) site we construct the activation cloud across prompts and score its H_1 persistence ratio against the matched null. The token positions are aligned across prompts as described in Appendix C. Positions before the mention carry identical activations across aligned prompts, so the scan is 0 there by construction and turns on at the mention (Figure 4); see §M.

Finding a topological signal does not by itself show that the corresponding site can causally affect the output. We therefore test the selected site with activation patching. A donor prompt carries the target label and its hidden states are copied onto the corresponding recipient span. We compare the discovered onset span against the true mention span, the strongest-signal span, an equal-size low-gate span, and the best-gated individual token. The discovered onset and oracle mention both reach percept 1.000, while all tested alternatives give 0.000. Thus the site selected without behavioural labels coincides with a site at which changing the representation changes the model's answer.

Matched-variance distractor. A possible alternative explanation is that persistence simply identifies tokens whose activations vary strongly across prompts. We construct a control where each prompt contains two independently varying quantities: a random single-token first name and the cyclic concept. The name is uncorrelated with the concept label, but produces approximately the same activation variance. At the tested sites, total variance is 1885 for the name and 1837 for the concept, so variance alone cannot distinguish them.

Their topology is different. The name has persistence ratio 0.96, below the null threshold, while the concept reaches 1.70. The causal test follows this difference: copying donor states onto the name leaves the percept at 0.96, whereas copying them onto the concept moves it to the target at 1.00 (Figure 5, left). Thus high activation variance is not sufficient for either a persistent cycle or a successful intervention.

Linear-probe control. We next ask whether a standard supervised decoder already provides the localisation information supplied by topology. At a fixed anchor site, we compare three predictors of donor-patch success across 11 tasks: the unsupervised H_1 ratio, cross-validated linear-probe accuracy, and total activation variance.

The distinction is between decodability and intervenability. The linear probe is at accuracy 1.000 on ten tasks and 0.998 on `dense_clock`, despite large differences in whether a donor patch changes the output. It therefore has no dynamic range with which to rank the tasks, and its rank correlation with patch success is determined by tie handling and a single near-tie (nominally -0.23, $p = 0.49$), so it carries no information. Variance reaches $r_s = +0.11$ ($p = 0.76$) and the H_1 ratio $+0.41$ ($p = 0.21$); neither is significant at $n = 11$. Figure 5 (right) shows the three comparisons. The probe answers whether information about the concept can be decoded at the site; the topological statistic instead tests for the geometric structure associated with sites at which a write can act.

Predictive validity across models. The previous comparison uses 11 tasks at a fixed anchor site. We next pool the same tasks across three models: Qwen2.5-3B, Llama-3.1-8B, and Qwen2.5-7B, giving $n = 33$ task–model pairs, and ask whether the persistence ratio orders donor-patch success. The pooled Spearman correlation is $r_s = +0.56$. The 33 pairs are 11 tasks $\times$ 3 models and the same task recurs across models, so they are not independent observations and we do not attach a pooled p-value to this number; the single-model estimate available is $r_s = +0.41$ on the 11 `Qwen2.5-3B` tasks above. The definition and interpretation of the Spearman statistic are given in Section G.

We additionally fit a Youden cutoff using only `Qwen2.5-3B`. The fitted threshold is 1.035. On the 22 held-out pairs it accepts 21 and rejects 1, and the rejected pair steers. Because $19/22$ held-out pairs steer, accepting every site (recall 1.0, precision 0.864) dominates the transferred cutoff (recall 0.947, precision 0.857, held-out false-positive rate 1.0), and the cutoff carries no transfer information. The cutoff does have content on `Qwen2.5-3B`, where it was fit and where the base rate of steerable sites is lower; on the held-out models the retained claim is the ordinal one.

The statistic also follows the representation in a particular model rather than assigning a fixed property to a task. For example, `hue` has ratio 0.68 and donor-patch success 0.49 on `Qwen`, but ratio 1.32 and success 0.99 on `Llama`. The same concept can therefore cross the topological threshold in one model and fail it in another.

Comparison with sparse autoencoders. Finally, we compare the persistence scan with a pretrained sparse autoencoder (SAE), another method for localising concept information (Cunningham et al., 2023). On `gemma-2-2b` layer 12, we compare the persistence map with a Gemma-Scope SAE η^2 map. The two token rankings agree substantially (Spearman 0.68), and both remain quiet before the concept is named.

The difference appears inside the mention span. SAE η^2 reaches approximately 0.99 on every token in the span, leaving the individual tokens nearly indistinguishable, whereas persistence has a single peak and therefore provides a more localised candidate site in this example. This comparison also has a computational trade-off. Persistence requires 13.3s of forward passes, whereas SAE encoding takes 0.51s once the dictionary is available. The SAE, however, relies on a dictionary pretrained on approximately 4B tokens, and no corresponding public SAE is available for `Qwen2.5-3B`. We therefore view this experiment as a comparison of localisation signals rather than a claim that persistent homology is computationally cheaper than an existing SAE.

H Extra Experiments

One could object two things:

- Editing the model internal state is the same as rewriting the prompt.
- If we look for a loop H_1, why not try to just fit a circle?

We provide evidence in the experiments below that intervention is not prompt editing and that topology and a circle fit are not redundant.

The intervention is not prompt editing. A prompt can say 3:00, or it can say 6:00. It has no way to ask for a time that sits partway between the two. The model can be put in that state directly. On the half-hour clock we mix the internal activation for a time i with the activation six steps later, $\lambda\,\mathrm{donor}(i{+}6) + (1-\lambda)\,\mathrm{donor}(i)$, and read off the time the model reports. The report does not drift through 3:30, 4:00, 4:30. It stays at the earlier time for the whole of $\lambda \le 1/3$, and it has switched to the later time by $\lambda \ge 2/3$. Exactly halfway, at $\lambda = 0.5$, the trials split between the two ends (source 0.47, target 0.43) and a time in between appears on only 0.05 of them. Every one of the 144 answers at each λ is still a real clock time. The model is rounding the blend to whichever of the two times it is closer to, and no wording of the prompt corresponds to $\lambda = 0.45$. The switch at $\lambda = 0.5$ belongs to this internal blend, a state no sentence can write, so the intervention is doing something editing the prompt cannot. We can observe this phenomenon on Figure 14. The figure shows the same blend: the dashed line is a steady walk across the six steps, the solid curve is the time the model actually reports, and the colours show those reports sitting on the earlier time and then flipping to the later one.

Topology and a circle fit are not redundant. A natural objection is that a two-dimensional circle fit would do the same job. Scanning 160 sites (8 tasks $\times$ 20 layers) and computing both the H_1 ratio and PCA roundness λ_2/λ_1, the two statistics are essentially *uncorrelated* ($\mathrm{corr} = -0.018$) and the joint distribution is populated in every cell (Figure 13): 80 sites high on both, **38** high in H_1 and low in roundness, 38 low/high, 4 low on both. The dissociation concentrates in `zh_clock` ($19/20$) and `heading` ($16/20$), with `dense_clock` splitting by depth ($3/20$, all at layers 5–8).

Whether that disagreement matters for steering is still open. Editing a site where the loop is strong and the cloud is not round succeeds on `heading` (percept 0.938–0.979) and fails on `zh_clock` (0.021). Sites with an equally strong loop and a round cloud succeed at 1.000 on both tasks. `dense_clock`

is the only task with both kinds of site, and there every site succeeds at 1.000, so the two cannot be told apart. No task in the set both varies in roundness and sometimes fails to steer, so the practical importance of this cell stays untested.

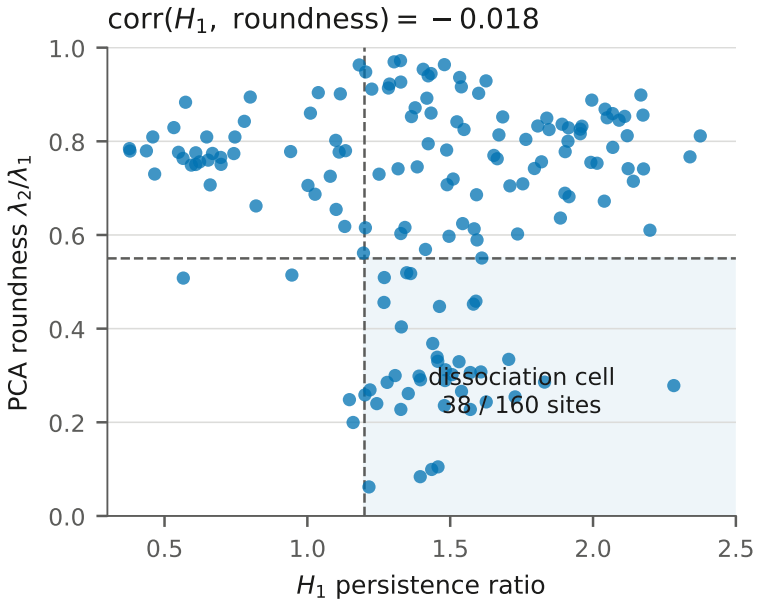

Figure 13: H_1 ratio against PCA roundness over 160 sites; the statistics are uncorrelated and 38 sites fall in the H_1-high / roundness-low cell.

Figure 14 mixes the activation for a clock time with the activation six steps later, $\lambda\,\mathrm{donor}(i{+}6) + (1-\lambda)\,\mathrm{donor}(i)$, and reads the time the model reports. The dashed line is the linear blend the geometry would produce, a steady walk of six steps. The solid curve stays on the source time through $\lambda \le 1/3$ and has switched to the target by $\lambda \ge 2/3$. The stacked panel shows the same switch in labels: intermediate times appear only in a thin band around the midpoint, and at $\lambda = 0.5$ the trials split between the two ends rather than landing between them. Every read is still a real clock time, so the model is rounding an internal state that no prompt can write.

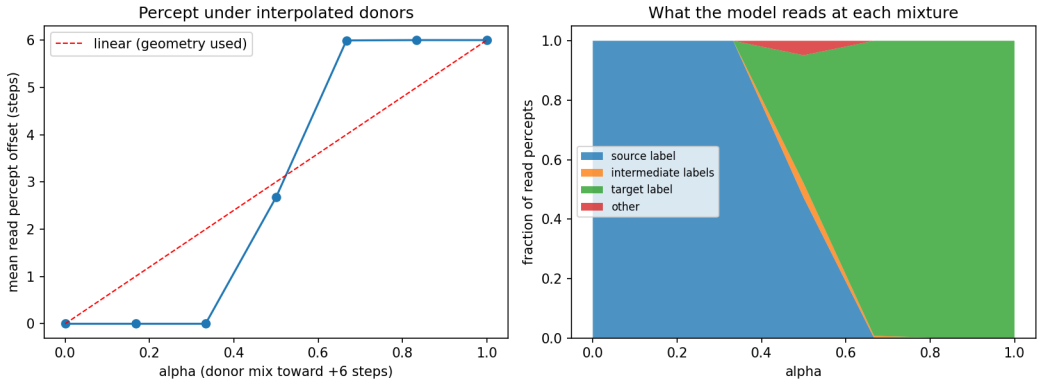

Figure 14: Donor interpolation gives a categorical read-out with a knife-edge at $\lambda = 0.5$ and $144/144$ valid reads.

Table 3 reports target percept, the fraction of short answers that contain the steered label, under four conditions. *Unsteered* is the model with no intervention. *Always-on* applies the steer to every input. *Gated* applies it only when the topological gate fires, and *random-gated* fires equally often but without regard to the input, so any gain of gating over that column is selectivity rather than a lower

firing rate. The last column is that selectivity directly: the gate's firing rate on target inputs against its rate on off-task inputs. The steer in every row is the donor patch of §4.1 applied to the mention span; gated and always-on write the same operator and differ only in which inputs receive it. The random gate is rate-matched over the whole workload, target and off-task inputs together, of which target inputs are roughly 40%, so its target percept is close to $0.4\times$ the always-on value in every cell. The upper block is the rotating design: three tasks on three models with five seeds each. The `weekdays` cells are ill-posed as steering targets, since the unsteered percept is already 1.000 (§M). Over the six well-posed cells, always-on is 0.611, gated 0.340 and random-gated 0.164; over all nine, 0.561, 0.435 and 0.350. Below the rule are the remaining cyclic tasks on `Qwen2.5-3B`, and the last row is the same comparison for the vision-language read.

Table 3: Target percept by condition (donor patch on the mention span). Higher is a stronger shift toward the steered label. The upper block is the rotating design (3 tasks $\times$ 3 models, 5 seeds). The `weekdays` rows are ill-posed as steering targets (unsteered already 1.000) and are excluded from pooled numbers; over the six well-posed cells, always-on is 0.611, gated 0.340 and random-gated 0.164. Below the rule are the remaining cyclic tasks on `Qwen2.5-3B`, then the VLM read. Fire rates are the gate on target inputs versus off-task inputs.

	unsteered	always_on	gated	random_gated	fire (tgt/off)
`Gemma-2-9B-it`					
`dense_clock`	0.000	0.083	0.062	0.025	0.854/0.000
`months`	0.000	0.111	0.028	0.044	0.139/0.289
`weekdays`	1.000	0.190	0.419	0.457	0.771/0.687
`Llama-3.1-8B-Instruct`					
`dense_clock`	0.000	0.750	0.708	0.297	0.958/0.028
`months`	0.000	0.722	0.389	0.150	0.533/0.048
`weekdays`	1.000	0.190	0.457	0.714	0.686/0.316
`Qwen2.5-3B-Instruct`					
`dense_clock`	0.000	1.000	0.796	0.321	0.796/0.000
`months`	0.000	1.000	0.056	0.144	0.056/0.245
`weekdays`	1.000	1.000	1.000	1.000	0.762/0.333
`clock`	0.000	0.917	0.417	0.206	0.472/0.000
`colors`	0.444	1.000	0.556	0.456	0.222/0.110
`hue`	0.000	0.521	0.396	0.254	0.812/0.138
`smds_date`	0.021	0.000	0.021	0.021	0.067/0.000
`smds_duration`	0.694	0.556	0.750	0.717	0.333/0.000
`smds_periodic`	0.806	0.194	0.417	0.506	0.602/0.000
`smds_time_of_day`	0.000	0.667	0.292	0.483	0.656/0.687
`Qwen2.5-VL-3B-Instruct`					
`vehicle_tod`	0.000	0.375	0.250	0.175	0.800/0.000

I SOME LIMITATIONS OF OUR GATING METHOD

Limitations of the gating method. A gate that fires less causes less damage on its own, so the right control is a random gate that fires equally often. On *target percept* (Table 3), gated minus random-gated is $+0.037$, -0.016, -0.038 on `Gemma` (`dense_clock`, `months`, `weekdays`), $+0.411$, $+0.239$, -0.257 on `Llama` and $+0.475$, -0.088, 0.000 on `Qwen`: $+0.085$ over nine cells and $+0.176$ over the six well-posed cells (standard deviation 0.234 across cells, ahead in $4/6$), with $+0.308$ on the three `dense_clock` cells and $+0.045$ on `months`. On the paired within-cell damage statistic the difference is $+0.004 \pm 0.103$ across 9 cells ($+0.027$ `Qwen`, -0.008 `Gemma`, -0.006 `Llama`), so on that quantity the two arms are indistinguishable. Neither comparison clears a 2σ criterion with six to nine cells. On the capability benchmarks both leave accuracy unchanged: the gate fires on 0.000 of benchmark tokens and the random gate on 0.0123. What the random gate cannot do is choose: the topological gate turns on for inputs that match the licensed cycle and stays off otherwise.

A second limitation is preservation of the full target effect. The target effect is a hit rate: how often the model's short answer contains the label the steer was trying to write in. Gating retains $0.435/0.561 = 0.78$ of the always-on effect over nine cells and $0.340/0.611 = 0.56$ over the six well-posed cells, both below our pre-registered 0.90. The gate trades effect for silence.

Finally, robustness of the damage reduction is not shown. We hoped to show that turning the steer on only sometimes reliably causes less harm than leaving it on all the time. The numbers lean that way, but the evidence is too noisy to count. On the paired test the reduction relative to always-on is $+0.153 \pm 0.265$, positive in $6/9$ cells; the spread, 0.265, is larger than the average itself, and a result only counts as solid here if it sits more than 2σ from zero. The capability experiment avoids this noisy paired estimate, but only because the gate never fires there (§M).

I.1 Hyperparameters

PCA components for the PH metric	16
Coefficient field	$\mathbb{Z}/47$ primary; $\{11, 47, 97\}$ for stability
Null draws	≥ 20 throughout (see §J)
Persistence quantile	0.95
Non-degeneracy	$U(\theta)$ check, Eq. 1
Gate coverage	0.90, calibrated in-sample on the fitting cloud
Reader-plane solver	minimum-norm least squares, rows orthonormalised
Layer rule (gate grid)	$\arg\max_\ell \rho$ over candidate layers, ties to the shallower layer
Steering operator (gate grid)	donor patch on the mention span, label shifted around the cycle
Steering operator (capability)	additive, $\alpha = 16$, set on the always-on curve alone
Generation	greedy, 24 new tokens, max length 256
Seeds	5 per cell (gate grid); randomise null draws and random-gate mask

I.2 Controls that changed a conclusion

Four controls were not obvious in advance and each reversed or qualified a result. We list them here:

1. **Rate-matching a gate control.** A gate that fires less causes less damage mechanically, so only a random gate at the *same* rate isolates the selection rule.
2. **Checking the hook reaches the scored positions.** Five standard benchmarks score context+continuation in one right-padded pass, so a last-position steering hook lands on a pad token and reports ≈ 0 delta on every task. It is indistinguishable from "the intervention is harmless". We added a pre-registered validity check: if always-on steering degrades nothing, the run is void, not positive.
3. **Pairing within cell, and counting the right unit.** Cells differ in task and model, so an unpaired spread counts task difficulty as noise. Separately, adjacent layers rescoring the same prompts are not independent: in one follow-up an apparently significant rank test ($p = 0.013$) dissolved on per-item inspection to a single prompt, with Fisher's exact test on items giving $p = 0.301$.
4. **Energy-matching a subspace control.** A Haar-random 2-plane in $\mathbb{R}^{2048}$ removes $\approx 0.1\%$ of residual variance while a concept plane is a top-2 subspace. It is a $32\times$ mismatch on our data, which would attribute to concept specificity what is really "removed more norm".

J Measurement notes

A five-draw null is underpowered. The persistence ratio is commonly scored against a null estimated from 5 draws. Its sampling error at that setting is as large as the between-task differences it is used to resolve. With 5 draws `honesty` (a linear control) ranges from 1.03 to 2.00 across seeds and can appear *above* the flagship cyclic task; at 50 draws the ordering is stable and correct. All numbers in this paper use ≥ 20 draws.

Bench	$n_{\text{null}} = 5$	$n_{\text{null}} = 50$
`dense_clock` (cyclic)	1.65 ± 0.13	1.50 ± 0.02
`weekdays` (cyclic)	1.91 ± 0.28	1.75 ± 0.10
`honesty` (linear ctrl)	1.39 ± 0.33	1.15 ± 0.05

Reproducibility does not certify a cycle. Across 24 sites, θ-reproducibility and θ-uniformity (Eq. 1) are anti-correlated at $r_s = -0.917$ ($p = 3.2 \times 10^{-10}$): where the coordinate genuinely winds it is not reproducible, and where it is reproducible it is nearly constant.

The ratio alone does not separate cyclic from linear concepts. At the mention site the ratio orders `weekdays` (1.87) above `dense_clock` (1.44) above `honesty` (1.14), whereas θ-uniformity

gives 0.10, 0.85 and 0.05. Only the second statistic identifies the one task whose cycle is genuinely traversed, which is why the gate uses both.

K SUPPLEMENTARY RESULTS

We report further results we observed during our experiments.

The isometry proposition, verified. Adding a constant vector to every point of a real activation cloud leaves the bottleneck distance at 0.00e+00 in both H_0 and H_1 at $\alpha = 1, 10, 100, 1000$, with the largest pairwise-distance change at 4.4×10^{-16}, floating point only. A per-point displacement of the same average magnitude moves it at once (0.081, 1.05, 16.6 at scales 1, 10, 100).

Consequently, measured topological displacement is uncorrelated with behavioural effect across five steering families at the last prompt token: Spearman -0.125 (bottleneck H_1), -0.329 (Wasserstein H_1), $+0.143$ (bottleneck H_0), $n = 15$. Within `dense_clock` the ranking inverts, the two methods that fully steer move the diagram *least* (0.113) while a prompt intervention with behaviour 0.000 moves it most (0.127). The small non-zero values are second-order leakage: the steer is injected across layers 5–20 and read at layer 11, and RMSNorm makes a constant offset non-uniform in between. What is measured is how the normalisation refracts the offset, which is roughly method-independent.

The cocycle plane is a poor write target. On a rank-r write ladder at the mention span, a supervised label-mean subspace reaches donor parity by $r = 8$ (0.979), while the cocycle write plateaus at 0.062–0.132 across $r = 2 \ldots 32$ against a donor patch at 1.000. The mechanism is the plane's share of the state: mean in-plane norm 2.193 against mean state norm 25.277, i.e. 8.7% of the norm and 0.75% of the energy, against 36% for the supervised label-mean plane. Supplying k labelled anchors to fix the cocycle basepoint does not help, percept is flat at 0.000 for k up to 64, and nor does a relative rotation, which needs no basepoint. Amplifying the write $50\times$ changes 0.688 of outputs while still giving percept 0.000: it disrupts without steering.

The plane is not identified. Across 24 sites, refitting the plane over coefficient fields and bootstrap resamples gives a median maximum principal angle of 80.7° against a random-2-plane baseline of 89.5°, with $0/24$ sites below 45°; reducing to a well-determined 16-dimensional regression does not help. Across a split it transfers at 88.6°, and an offset refitted on the held-out half beats the transferred one at $24/24$ sites.

Ablating the plane is inert. With an energy-matched control (§I.2) the cocycle plane's accuracy drop is -0.011 against the control's -0.010 — a gap of -0.001 with $2\sigma = 0.017$, holding on $1/4$ tasks. Only the supervised label-mean plane, which removes 18.5–46.2% of energy, moves behaviour appreciably.

Definition K.1. Ordinary persistent homology follows one growing filtration: each complex sits inside the next, and a bar records the scales at which a loop is born and dies. Zigzag persistence allows the sequence to shrink as well as grow. The maps alternate between forward and backward inclusions, and a bar is still a birth and a death, now indexed by position in that sequence rather than by a distance ϵ. In this paper the sequence is depth. The same prompts are embedded at each layer, so the vertices are shared, but the distances change and K_ℓ does not sit inside $K_{\ell+1}$. The zigzag inserts the union between them,

$$K_\ell \;\subseteq\; K_\ell \cup K_{\ell+1} \;\supseteq\; K_{\ell+1}.$$

An H_1 bar alive from layer a to layer b is a loop that survives that range of depth. The negative result takes the midpoint of the longest such bar as the injection layer.

Zigzag persistence does not select the layer. Selecting an injection layer a priori from the longest-lived H_1 zigzag bar across depth lands in the behaviourally healthy layer band 44% of the time, against 33% for a random layer and 100% for taking the middle of the window. The choice is also unstable in the filtration scale: at ϵ-quantile 0.2 the selected layers across seeds are $\{11, 16, 17\}$, at 0.3 $\{7, 18\}$, at 0.4 $\{17, 18\}$.

Sublevel-set filtrations read but do not act. A lower-star filtration by a scalar field is not an isometry invariant and applies to ordered concepts where H_1 is vacuous. Its criterion ranks the concept-carrying direction at mean rank 4.0 of 16 against a chance of 8.5 ($z = 3.38$, $p = 0.0004$),

inside the top 3 at $9/12$ sites, on `honesty`, `sycophancy`, `sentiment` and `linear_order`. But steering along the selected directions has no causal validity: across 32 (task, component) pairs the criterion predicts causal disruption at $r_s = -0.227$ ($p = 0.21$) while the supervised label correlation predicts it at $r_s = +0.601$ ($p = 0.0003$), and no arm, including the supervised oracle, raises the target percept above unsteered.

L Data-dependent, independent and Gaussian null.

Data-dependent edits and directional ablation. Traditional activation steering is a data-independent method because we add the same vector to every input. We oppose it to data-dependent methods, in which we tailor the change to each input. This is the distinction we use in Proposition 3.1.

Data-independent. The same displacement for every input. Additive steering $x \mapsto x + \alpha v$ (Turner et al., 2024; Panickssery et al., 2024) and rotation in a fixed plane $x \mapsto Rx$ (Vu & Nguyen, 2025).

Data-dependent. A displacement that varies per input. Activation patching writes a donor state $x \mapsto y$ at a chosen token (Vig et al., 2020; Meng et al., 2022; Engels et al., 2025); directional ablation removes a subspace $x \mapsto (I - P_V)x$ (Arditi et al., 2024).

Here $V \subset \mathbb{R}^{d_h}$ is the subspace to remove (e.g. the span of a concept direction) and P_V is orthogonal projection onto V. If $V = \mathrm{span}\{v_1, \ldots, v_k\}$ with orthonormal columns in $Q \in \mathbb{R}^{d_h \times k}$, then $P_V = QQ^\top$ and $(I - P_V)x = x - QQ^\top x$. In the rank-1 case $V = \mathrm{span}\{\hat{v}\}$ with $\|\hat{v}\| = 1$, this is $x \mapsto x - (\hat{v}^\top x)\hat{v}$: the subtracted vector scales with the input's own coordinate $\hat{v}^\top x$.

The Gaussian null. Because a loop in a small cloud can outlast a larger loop in a spread-out cloud, the length of an H_1 bar is not comparable across sites on its own. The covariance of the Gaussian null defined in section 3.1 is diagonal, so the construction remains well-defined when $n < d$. Scoring the data against that quantile, rather than against a cloud in a different metric, is what makes the comparison fair. Every finite cloud has some H_1 bar, as it could be due to some noise, so the persistence ratio of Section 3.1 asks whether the longest bar on this site outlasts what noise of this size and this n almost always produces. A ratio above 1 means the longest loop is longer than chance, here a Gaussian null draw, would produce at that scale and sample size. $\rho > 1$ does not signify that "an H_1 bar exists."

The null is estimated from at least 20 draws. Appendix J shows that the 5-draw null common in this area has sampling error as large as the differences it is meant to resolve.

Remark 1. Say at test time we currently have **one** prompt. It is, then, tempting to skip the cloud and test a single activation, or a small neighbourhood of one. However just having one prompt is not enough to define a cycle. That test cannot consequently work. The nearest neighbours of a point on a circle form an arc, and an arc has no H_1 class, so local persistence is near zero even for an input that sits squarely on the manifold. The cycle has to be established from the whole cloud first, and only then projected onto a new point.

M Further clarifications

The transferred cutoff and the pooled correlation. On the 22 held-out task–model pairs the 1.035 cutoff yields 18 true positives, 1 false negative, 3 false positives and 0 true negatives. Accepting every site attains recall 1.0 and precision $19/22 = 0.864$, so the cutoff has no discriminative content on the held-out models (Youden's $J = -0.053$). The reason is the base rate: on `Llama-3.1-8B` and `Qwen2.5-7B`, $19/22$ tasks steer, so almost every site is above 1 and there is nothing for a screen to reject. The cutoff has content on `Qwen2.5-3B`, where it was fit and where the base rate is lower (`hue`: $\rho = 0.68$, steer 0.49). The evidence for predictive validity is ordinal: $P(\text{steer} \mid \rho \text{ bin})$ increases across bins (Figure 6, left) and the pooled r_s is $+0.56$. The 33 pairs are 11 tasks $\times$ 3 models, and the same task recurs across models, so a nominal p-value on $n = 33$ would overstate the evidence; the single-model estimate is $r_s = +0.41$ on the 11 `Qwen2.5-3B` tasks. The linear-probe comparison of §4.1 makes a narrower claim than a race between predictors: a classifier at accuracy 1.000 on ten tasks and 0.998 on one has no dynamic range, so its rank correlation with patch success is determined by tie handling, and the H_1 ratio is $r_s = +0.41$ at $n = 11$.

Layer choice. The layer rule of §4.2 is: among candidate layers, fit the gate on the calibration cloud at the mention token and take the layer with the largest persistence ratio, ties toward the shallower

layer; no behavioural quantity is read. §K reports that selecting the layer from the longest zigzag bar lands in the behaviourally healthy band 44% of the time, against 100% for the middle of the scanned window. Within a single model, therefore, a fixed middle-layer default is a strong choice. What the ratio contributes that a fixed default cannot is the abstention decision across (task, model) pairs: `hue` on `Qwen2.5-3B` is below 1 at the anchor layer and steers at 0.49, while on `Llama` it is 1.32 and steers at 0.99.

Donor patching and additive steering. Two interventions appear in the paper. The causal check of §4.1 and the entire gate grid of Table 3 use the same operator: a donor patch of the mention span with the state of a prompt whose label is shifted around the cycle. The success that ρ orders is therefore the success of the operator used in the gate grid. The capability experiment of §4.3 uses additive steering ($\alpha = 16$) because a donor patch is undefined on benchmark text. The localisation claim is that the topological site is one at which the concept is causally read; §K shows that a site which reads can fail to write along the cocycle plane, in line with Hase et al. (2023).

Cyclic specificity and the uniformity check. In §4.2, `dense_clock`, `weekdays` and `honesty` all have $\rho > 1$ at the mention site (1.44, 1.87, 1.14). Licensing requires both $\rho > 1$ and the non-degeneracy check on $U(\theta)$ in equation 1. The persistence ratio alone does not separate cyclic from linear concepts on clouds of this size: the diagonal Gaussian null is the weakest null that is well defined at $n < d$, and any non-Gaussian structure, including clusters or the discreteness of templated labels, can exceed it. $\rho > 1$ is therefore a necessary condition for gating a site, not a certificate that the site carries a cycle; $U(\theta)$ (0.85 for `dense_clock`, 0.10 for `weekdays`, 0.05 for `honesty`) is the statistic that separates them, which is why the gate uses both. `weekdays` illustrates the limit of small clouds: with $n = 7$ the longest H_1 bar is a statement about a single heptagon, and seven Gaussian points in $\mathbb{R}^{16}$ rarely form a long loop, so the cloud passes the null without the circular coordinate spreading around the cycle. For this reason tasks with $n \leq 8$ (`compass`, `seasons`, `colors`, `weekdays`, `zh_weekdays`, `emotions`) are exploratory, and the main claims rest on `dense_clock` (48), `zh_clock` (48), `heading` (36), `hue` (24), `clock` and `months` (12). All ratios use PCA-16 for both the data cloud and the null, so the comparison is matched; the sensitivity of the ratio to this dimension has not been swept.

The reader plane. $(\cos 2\pi\theta_i, \sin 2\pi\theta_i)$ is regressed on the centred hidden states by minimum-norm least squares. With $n \leq 48$ and $d_h = 2048$ the system is underdetermined; the minimum-norm solution lies in the span of the centred calibration points and would interpolate any set of n target angles. The basis of the plane is accordingly not identified across refits (median maximum principal angle 80.7°, §K). The membership test does not use the basis. It uses two statistics, the in-plane radius r and the in-plane fraction $f = r/\|x - \mu\|$. For a Haar-random 2-plane in $\mathbb{R}^{2048}$, $\mathbb{E}\, f \approx \sqrt{2/d_h} = 0.031$ for every input, so a coverage threshold on f would be near zero and the test would reduce to a test on $\|x - \mu\|$, that is, a distance-to-centre detector. The fitted plane captures 8.7% of the norm of calibration points (§K), roughly 3× the random value in norm and 7.5× in energy, because it lies in the span of the calibration cloud; the f test therefore constrains the direction of $x - \mu$ and not only its length. The thresholds (τ_r, τ_f) are set at 90% coverage on the calibration points used to fit (W, μ). The held-out target firing rates in Table 3 are the out-of-sample numbers: 0.80 to 0.96 on `dense_clock` across models, close to nominal, and 0.06 to 0.53 on `months`. On the language tasks every prompt names the concept, and a lexical rule would detect `dense_clock` inputs from the string alone. The gate reads the same object the steer writes, the residual-stream activation, so that it can be evaluated inside the forward hook on inputs where no string is available; the `vehicle_tod` row of Table 3 (0.800/0.000) is the case of this kind in the paper.

Gate versus rate-matched random gate. In the rotating design each workload contains the target task's inputs, the two other cyclic tasks' inputs and `honesty`. The gate decides per input from the unsteered activation at the mention token of the selected layer; when it fires, the same donor patch as in the always-on arm is applied to that input's mention span. The random gate fires on the same overall fraction of workload inputs, chosen uniformly at random. Gated and always-on therefore write the same operator at the same positions and differ only in which inputs receive it; off-task rows are scored against their own label, so off-task damage is the loss of the model's original correct reading. The seed randomises the null draws used to fit the gate and the random-gate mask; decoding is greedy. Because the random gate spends its firings on the whole workload, of which target inputs are roughly

40%, its target percept is close to $0.4\times$ the always-on percept in every cell (`Qwen dense_clock`: 0.321 against 1.000; `Llama dense_clock`: 0.297 against 0.750). The topological gate's margin over it on target percept is its selectivity: firing on target inputs rather than on the workload at large. From the cell means of Table 3, gated minus random-gated is $+0.085$ over nine cells, $+0.176$ over the six well-posed cells (standard deviation 0.234, ahead in $4/6$), $+0.308$ on the three `dense_clock` cells and $+0.045$ on `months`, consistent with the two inverted selectivity cells in Figure 7. The pooled means over the six well-posed cells are unsteered 0.000, always-on 0.611, gated 0.340 and random-gated 0.164. Gating retains $0.340/0.611 = 0.56$ of the always-on target effect and the rate-matched random gate 0.27; over nine cells the retained fraction is $0.435/0.561 = 0.78$. Both are below the pre-registered 0.90. None of these differences clears a 2σ criterion with six cells.

The percept metric and the weekdays cells. Target percept is the fraction of short answers that contain the steered label, where the steered label is the calibration label shifted around the cycle. It is a hit rate. In §4.1, "donor patching at the name does not steer" means that the source label was retained on 0.96 of answers. For `weekdays` the unsteered percept is 1.000 on all three models: the unsteered answer already contains the shifted label, so in these cells always-on steering can only lower the percept, a gated value above always-on measures steering less, and the cells carry no information about selectivity. They are excluded from the pooled comparison. For the same reason, tasks with a high unsteered baseline (`smds_periodic` 0.806, `smds_duration` 0.694, `colors` 0.444) are read on the difference between always-on and unsteered rather than on the raw percept, and `smds_date` (always-on 0.000) is a failed steer on which the gate cannot be evaluated. The `smds_*` tasks are the temporal templates of Tiblias et al. (2025), imported unchanged, binned to a period (time of day on a 15-minute grid, period 96; dates in weekly bins, period 48) and frozen; they serve as externally authored cyclic prompts. The $\rho = 0.68$ for `hue` on `Qwen2.5-3B` in §4.1 is at the fixed anchor layer of the eleven-task predictor comparison. The gate grid selects the layer per cell by $\arg\max \rho$ over candidate layers, and the `hue` row of Table 3 is gated at that layer, where a cycle was licensed; the two numbers are at different sites, and the method abstains at the anchor layer.

The capability experiment. The gate fires on none of the benchmark tokens, so the gated arm is the unsteered model and $\Delta = 0$ follows. The experiment does not show a capability gain over other low-rate gates, and the random gate at 1.2% confirms this. What it measures is specificity: over all scored tokens of the four benchmarks, evaluated at every position by an in-hook gate that is told nothing about where a concept might be, the gate has no false positive. The gate is calibrated on a concept-token cloud and benchmark tokens are far from it under any distance test, so this is a weak form of specificity; the off-task cells of §4.2, where the off-task inputs are other cyclic concepts, are the stronger test (`dense_clock`: 0.000 to 0.028 off-task firing). `gsm8k` is excluded from the mean because answer extraction fails at 0-shot in every arm; the always-on drop over the four scored benchmarks is -0.205. With 100 examples per benchmark the standard error per cell is roughly 0.04 to 0.05; the always-on drops on ARC-e (0.42) and HellaSwag (0.20) lie far outside it, and the TruthfulQA and random-gated differences lie within it.

Proposition 3.1. Isometry invariance of Vietoris–Rips persistence is immediate from the construction. Its role in the paper is to fix the order of operations: a per-cloud topological statistic cannot verify that a common additive or rotational edit was applied, so the gate reads the unsteered activation. It gives a sufficient condition for invariance and not a characterisation of what distance-based filtrations can detect. A per-input check $c(x + \alpha v) = c(x) + \alpha W v$ is available and verifies that the write occurred, which is a different question from whether the concept is present.

N Reproducibility

All experiments run on a single GPU. Code, pre-registered criteria, per-run verdict reports and the raw JSON behind every table are released with the paper. Unit tests cover the controls that would otherwise fail silently: that a position-`all` hook edits every position while a position-`last` hook does not; that the in-hook gate is invariant to row permutation; that an energy-matched control matches while a Haar control does not; and that off-task labels are never clipped into the target's index space.