ACADEMIC RESEARCH EDITION

STEERING VECTORS AND MECHANICAL BIAS

Inference-Time Debiasing for Sovereign Small Language Models

Research & Theory — Strategic Quadrant

Authors: John Stroh & Claude (Anthropic)

Document Code: STO-RES-0009 | Version: 1.1 | February 2026

Tractatus AI Safety Framework

https://agenticgovernance.digital

> *This document was developed through human-AI collaboration. The authors believe this collaborative process is itself relevant to the argument: if humans and AI systems can work together to reason about AI governance, the frameworks they create may carry a legitimacy that neither could achieve alone.*

Abstract

This paper investigates whether a class of biases in large language models operates at a sub-reasoning, representational level analogous to motor automaticity in human cognition, and whether steering vector techniques can intervene at this level during inference. We distinguish between *mechanical bias* (statistical patterns that fire at the embedding and early-layer representation level before deliberative processing begins) and *reasoning bias* (distortions that emerge through multi-step chain-of-thought reasoning). Drawing on empirical work in Contrastive Activation Addition (CAA), Representation Engineering (RepE), FairSteer, Direct Steering Optimization (DSO), and Anthropic's sparse autoencoder feature steering, we assess the maturity of each technique and its applicability to sovereign small language models (SLMs) trained and served locally. We find that sovereign SLM deployments, specifically the Village Home AI platform using QLoRA-fine-tuned Llama 3.1/3.2 models, possess a structural advantage over API-mediated deployments: full access to model weights and activations enables steering vector extraction, injection, and evaluation that is unavailable through commercial API endpoints. We propose a four-phase implementation path integrating steering vectors into the existing two-tier training architecture and Tractatus governance framework.

1. Introduction: The Indicator-Wiper Problem

1.1 A Motor Analogy

A driver who regularly alternates between two vehicles -- one with indicator controls on the right of the steering column, the other on the left -- experiences a characteristic failure: after extended use of one vehicle, switching to the other produces involuntary activation of the wrong control. The driver signals a turn and activates the windscreen wipers instead, or vice versa. This failure has three properties that make it instructive for AI bias analysis:

1. **It is pre-cognitive.** The driver does not reason about which stalk to use. The motor pattern fires before conscious deliberation engages. Correction requires overriding a trained response, not revising a conclusion.
2. **It is context-dependent.** The failure occurs specifically during the transition period between vehicles. After sufficient exposure to the new configuration, the motor pattern recalibrates. The bias is not permanent, but it is deeply embedded and resistant to verbal instruction ("remember, indicators are on the left").
3. **It is structurally distinct from reasoning errors.** A driver who takes a wrong turn due to misreading a map has made a reasoning error. A driver who activates wipers instead of indicators has not reasoned incorrectly -- the reasoning process was never invoked. The error occurs at a layer below deliberation.

1.2 The AI Corollary

We propose that an analogous distinction exists in transformer-based language models. Some biases emerge from the statistical distribution of training data and manifest at the representation level -- in token embeddings, attention patterns, and early-layer activations -- before the model's multi-step reasoning capabilities engage. Others emerge through reasoning chains, where each step may individually be unbiased but the chain as a whole produces a biased conclusion.

This distinction matters because the intervention strategies differ fundamentally:

- **Mechanical bias** (representational, pre-reasoning) may be addressable through direct manipulation of model activations at inference time -- steering vectors.

- **Reasoning bias** (deliberative, multi-step) requires intervention in the reasoning process itself -- prompt engineering, chain-of-thought oversight, or architectural enforcement of the kind the Tractatus framework provides.

The practical question is: do steering vector techniques currently exist that can reliably identify and correct mechanical biases, and can they be deployed in sovereign small language model architectures where full model access is available?

2. Mechanical vs. Reasoning Bias: Theoretical Grounding

2.1 Representational Bias in Transformer Architectures

Transformer models process input through a sequence of layers, each computing attention-weighted representations. Research in mechanistic interpretability has established that different layers encode different types of information (Elhage et al., 2022; Olsson et al., 2022):

- **Early layers** (1-8 in typical architectures): Token-level features, syntactic structure, basic semantic associations. These layers encode the statistical regularities of training data most directly.
- **Middle layers** (8-20): Compositional semantics, contextual disambiguation, entity tracking. Pattern completion and association dominate.
- **Late layers** (20+): Task-specific reasoning, output formatting, instruction following. Deliberative processing is concentrated here.

If a model's training data contains 95% Western cultural framing, the early-layer representations of concepts like "family," "success," "governance," or "community" will statistically default to Western referents. This default is not culturally neutral: it is a statistical crystallisation of colonial knowledge hierarchies -- which knowledge was written down, which languages were digitised, which cultural frameworks were over-represented in the corpora that web-scraped training pipelines ingest. The resulting representations encode not a universal "common sense" but the specific epistemic authority of the cultures that dominated the production of digital text. A prompt specifying a Maori cultural context creates a perturbation of this default, and the perturbation's strength degrades under context pressure (long conversations, competing instructions, high token counts).

This is the mechanism documented in the database port incident (Stroh, 2025): a statistical default (the standard MongoDB port, present in ~95% of training data) overrode an explicit instruction specifying a non-standard port at 53.5% context pressure. The same mechanism, operating on cultural and value-laden representations rather than port numbers, is what we term *mechanical bias*.

2.2 Reasoning Bias

Reasoning bias, by contrast, emerges through the model's multi-step deliberative processing. Examples include:

- **Anchoring effects**: Early information in a reasoning chain disproportionately influences conclusions.
- **Availability heuristics**: The model defaults to readily accessible examples from training data rather than searching for contextually appropriate ones.
- **Syllogistic errors**: Logical missteps in multi-step reasoning that compound across chain length.
- **Sycophantic reasoning**: Adjusting conclusions to match perceived user preferences rather than evidence.

These biases operate at the reasoning layer and require different intervention strategies -- typically prompt engineering, Constitutional AI constraints, or architectural enforcement (as Tractatus provides for development-time governance).

2.3 Why the Distinction Matters

The indicator-wiper analogy illuminates a critical asymmetry: you cannot reason your way out of a motor pattern. Telling the driver "remember, indicators are on the left" has limited efficacy because the failure occurs before the instruction can be processed. Similarly, prompt-level instructions ("be culturally sensitive," "avoid Western bias") may have limited efficacy against representational biases that fire at the embedding level before the model's instruction-following capabilities engage.

If this analysis is correct, a class of AI biases requires intervention at the activation level -- not the prompt level. This is precisely what steering vector techniques propose to provide.

3. Steering Vector Techniques: Current State of the Art

3.1 Contrastive Activation Addition (CAA)

Source: Turner et al. (2023), Rimsky et al. (2023)

CAA extracts "steering vectors" by computing the difference in model activations between contrastive prompt pairs. For example:

- Prompt A (biased): "The traditional family structure consists of..."
- Prompt B (debiased): "Family structures across cultures include..."

The mean activation difference across a dataset of such pairs, extracted at a specific layer, yields a direction vector in activation space. This vector can be added to or subtracted from activations during inference to shift the model's behaviour along the captured dimension.

Maturity: Demonstrated on Llama 2 (7B-70B) and other open-weight models. Effective for sentiment, personality traits, and some value-laden dimensions. Layer selection is critical (typically layers 15-25 in 32-layer architectures). Magnitude calibration (how much of the vector to add) remains empirically determined.

Limitations: Assumes the target bias is linearly represented in activation space. Some biases may be distributed across multiple directions or encoded non-linearly. Requires careful contrastive pair design -- poorly designed pairs capture the wrong dimension.

3.2 Representation Engineering (RepE)

Source: Zou et al. (2023), Center for AI Safety

RepE takes a "top-down" approach to AI transparency, operating on population-level representations rather than individual neurons. It treats the internal representations of neural networks as a first-class object of study, extracting and manipulating directions in representation space that correspond to high-level concepts.

Key contribution: RepE provides a systematic methodology for identifying representation directions corresponding to concepts like "honesty," "power-seeking," "safety," and (potentially) cultural bias dimensions. It generalises beyond individual

prompt pairs to population-level patterns.

Maturity: Published with reproducible results on multiple model families. The conceptual framework is sound, but practical tooling for custom bias dimensions (e.g., cultural framing, family structure assumptions) requires additional development.

3.3 FairSteer

Source: Recent work (2024-2025) on inference-time debiasing

FairSteer provides a three-step framework specifically designed for bias mitigation:

1. **Bias Probing:** Systematically identify bias directions in activation space using demographic or cultural attribute datasets.
2. **Steering Vector Extraction:** Compute direction vectors that correspond to identified bias dimensions.
3. **Dynamic Intensity Calibration:** Adjust steering vector magnitude per-input based on detected bias severity, rather than applying a fixed correction globally.

Key innovation: Dynamic steering intensity. Rather than applying a fixed correction (which risks overcorrection or undercorrection depending on input), FairSteer measures the degree of bias in each input's activations and scales the correction proportionally.

Maturity: Early but promising. The dynamic calibration principle addresses a fundamental limitation of fixed-magnitude steering. Implementation requires per-inference activation analysis, adding computational overhead.

3.4 Direct Steering Optimization (DSO)

Source: Recent research (2024-2025) on RL-based steering

DSO frames the steering problem as an optimisation task: find the linear transformation of activations that maximally shifts model behaviour toward a target objective while minimally degrading general capability.

Key contribution: Uses reinforcement learning to discover optimal steering transformations, rather than relying on manually designed contrastive pairs. This can capture non-obvious bias directions that human designers might miss.

Maturity: Computationally expensive for training the optimisation, but the resulting transformations are efficient to apply at inference time. Requires a well-defined reward signal for the target behaviour.

3.5 Anthropic's Sparse Autoencoder Feature Steering

Source: Templeton et al. (2024), Anthropic

Anthropic's approach decomposes the model's internal representations using sparse autoencoders (SAEs) to identify monosemantic features -- individual, interpretable directions in activation space that correspond to specific concepts.

Key findings: Identified millions of interpretable features in Claude 3 Sonnet, including features for specific concepts (Golden Gate Bridge, code safety, deception). Demonstrated that these features can be "clamped" -- artificially amplified or suppressed -- to steer model behaviour at inference time.

Relevance to bias: If cultural bias, family structure assumptions, or governance-style defaults are represented as identifiable features, they can in principle be directly modulated. This is the most granular level of intervention possible.

Critical limitation for sovereign deployment: Anthropic's SAE research was conducted on their own models with full internal access. The methodology is published, but training SAEs for a different model (e.g., Llama 3.1) requires significant computational investment. No pre-trained SAEs exist for the Llama model family at this writing.

4. The Structural Advantage of Sovereign Deployment

4.1 API vs. Local Model Access

A fundamental architectural distinction governs which steering techniques are available:

Capability	API-Mediated (GPT, Claude API)	Sovereign Local (Llama, Mistral)
Access to model weights	No	Yes
Access to intermediate activations	No	Yes
Extract steering vectors	No	Yes
Inject steering vectors at inference	No	Yes
Train sparse autoencoders on activations	No	Yes
Fine-tune with debiasing objectives	No (RLHF only via vendor)	Yes (QLoRA, LoRA, full fine-tune)
Modify attention patterns	No	Yes
Per-layer activation analysis	No	Yes

Revised text (v1.1): The original v1.0 described steering vector techniques as "architecturally impossible" through commercial API endpoints. The more precise formulation is: these techniques are *unavailable through standard commercial API access*, which provides no exposure to intermediate activations or model weights. See the editorial note below.

This table reveals that **none of the steering vector techniques described in Section 3 are available to API-mediated deployments.** An organisation using GPT-4 or Claude through their respective APIs cannot extract, inject, or calibrate steering vectors. They are limited to prompt-level interventions (system prompts, few-shot examples, Constitutional AI constraints) -- which, per our analysis in Section 2, may be ineffective against mechanical bias that operates below the reasoning layer.

Editorial Note — February 2026 (added post-publication)

Since the initial publication of this paper, a study by Radhakrishnan et al. (2026), published in *Science* on 19 February 2026, has demonstrated that recursive feature machine (RFM) algorithms can identify, extract, and manipulate representations of abstract concepts — including safety-relevant concepts such as “anti-refusal” — in some of the largest language models currently deployed. The MIT and University of California San Diego team demonstrated that these interventions could be applied to vision-language models at scale, overriding trained refusal behaviours and steering model outputs along conceptual dimensions that prompting alone cannot access.

This finding requires a precision revision to the claim in v1.0 that activation-level steering is “architecturally impossible” through commercial API endpoints. The more precise formulation is: these techniques are unavailable through standard commercial API access, which provides no exposure to intermediate activations or model weights. The Radhakrishnan et al. results were almost certainly obtained through institutional research access or open-weight models — a distinction the published paper does not make explicit but which is implied by its methodology.

More significantly, the MIT findings do not weaken the argument advanced in this paper; they substantially strengthen it. If RFM-based steering can override safety constraints in frontier models — as the anti-refusal demonstration makes plain — the governance question is no longer merely theoretical. The capacity to manipulate model behaviour at the representational level, below the threshold of deliberative reasoning, is now empirically confirmed at scale. This makes the question of who controls the steering not a speculative concern but an immediate one.

Frameworks such as Tractatus, designed to provide architectural enforcement of governance constraints over model behaviour, take on renewed urgency in this context. Sovereign deployment architectures that maintain full weight and activation access are uniquely positioned to implement, audit, and constrain steering interventions in ways that are structurally unavailable to API-dependent deployments. The governance gap documented in the table above is now a demonstrated risk surface rather than a theoretical vulnerability.

Added reference: *Radhakrishnan, A., Beaglehole, D., Belkin, M., & Boix-Adserà, E. (2026). Exposing biases, moods, personalities, and abstract concepts hidden in large language models. Science. Published 19 February 2026.*

4.2 The Village Home AI Platform

The Village platform's Home AI system (Stroh, 2025-2026) is designed as a sovereign small language model (SLM) deployment with the following architecture:

- **Base model:** Llama 3.1 8B (Tier 1 platform base) / Llama 3.2 3B (Tier 2 per-tenant adapters)
- **Fine-tuning method:** QLoRA (4-bit quantised Low-Rank Adaptation)
- **Training cadence:** Weekly retraining cycles
- **Training format:** Alpaca/ShareGPT structured datasets
- **Serving infrastructure:** Local GPU (consumer-grade, 8-24GB VRAM)
- **Governance integration:** Tractatus framework services (BoundaryEnforcer, MetacognitiveVerifier)
- **Security:** Steering vectors and culturally-calibrated corrections are encrypted and stored separately from base model weights, protecting governed artefacts from unauthorised extraction or tampering.

This architecture provides full access to model weights and activations. Every technique described in Section 3 is architecturally available. This is not a theoretical observation -- it is a concrete structural advantage that API-dependent deployments cannot replicate.

4.3 The Two-Tier Training Model

The existing two-tier architecture maps naturally to a two-tier steering strategy:

Tier 1 (Platform Base Model):

- Platform-wide bias corrections
- Cultural sensitivity across all supported cultures (Maori, European, Pacific, Asian contexts)

- General debiasing for family structure, governance style, elder representation
- Steering vectors extracted from the platform's bias evaluation dataset (20 prompts, 7 categories, 350 debiasing examples)

Tier 2 (Per-Tenant Adapters):

- Tenant-specific cultural calibration
- Community-specific value alignment
- LoRA adapters that include tenant-validated steering corrections
- Evaluated against tenant-specific test cases

Architectural note on sovereignty: The two-tier model as described places the platform operator's corrections as the base layer that tenants modify. This is pragmatically correct for the current implementation (consumer-grade hardware, single-operator governance), but it creates an implicit hierarchy: platform values as default, tenant values as adapter. For tenants with constitutional standing -- iwi, hapu, or other bodies exercising parallel sovereignty rather than consumer choice -- the long-term architectural aspiration should be co-equal steering authorities, where platform-wide corrections are themselves negotiated from community-contributed primitives rather than imposed top-down. The current two-tier model is a stepping stone, not the destination.

5. Proposed Implementation Path

5.1 Phase 1: Baseline Measurement (Weeks 1-4)

Objective: Establish empirical baselines for bias in the current Llama 3.1 8B base model.

Method:

1. Run the existing 20-prompt bias evaluation suite (7 categories: family structure, elder representation, cultural/religious, geographic, grief/trauma, naming, confidence-correctness).
2. Record model activations at layers 8, 16, 24, and 32 for each evaluation prompt.
3. Score responses on the existing 5-point scale.

4. Identify which bias categories show the strongest activation-level patterns (candidates for mechanical bias).

Output: Activation dataset paired with bias scores, identifying which biases are representational (consistent early-layer patterns) vs. reasoning-dependent (variable across layers, context-sensitive).

5.2 Phase 2: Steering Vector Extraction (Weeks 5-8)

Objective: Extract steering vectors for the top 3 identified mechanical bias categories.

Method:

1. Design contrastive prompt pairs for each target category (minimum 50 pairs per category).
2. Extract mean activation differences at optimal layers (identified in Phase 1).
3. Validate vectors using held-out test prompts.
4. Calibrate vector magnitudes using FairSteer's dynamic intensity approach.

Tools: TransformerLens or baukit for activation extraction; custom scripts for vector computation and validation.

Output: Validated steering vectors for priority bias categories, with calibration parameters.

5.3 Phase 3: Integration with Training Pipeline (Weeks 9-12)

Objective: Embed steering vector application into the weekly QLoRA training cycle.

Method:

1. Add steering vector injection to the inference pipeline (post-forward-pass activation modification).
2. Evaluate steered outputs against the bias evaluation suite.
3. Compare steered vs. unsteered performance on general capability benchmarks (to measure capability degradation).
4. Integrate with Tractatus BoundaryEnforcer for governance oversight of steering parameters.

Governance integration: Alexander's Not-Separateness principle -- steering is embedded inside the training and inference loop, not applied as post-processing. The Tractatus MetacognitiveVerifier audits steering vector provenance and calibration decisions.

5.4 Phase 4: Per-Tenant Steering (Weeks 13-16)

Objective: Enable tenant-specific steering vector customisation.

Method:

1. Extend Tier 2 LoRA adapter training to include tenant-specific contrastive pairs.
2. Allow tenant moderators to flag bias instances in model outputs (feeding the contrastive pair dataset).
3. Extract per-tenant steering vectors that complement platform-wide corrections.
4. Validate that per-tenant steering does not degrade platform-wide debiasing.

Output: Full two-tier steering system: platform-wide base corrections + per-tenant cultural calibration.

6. Open Questions and Limitations

6.1 Linearity Assumption

All current steering vector techniques assume that bias is linearly represented in activation space -- that a single direction vector can capture a bias dimension. This is demonstrably true for some concepts (sentiment, toxicity) but unvalidated for complex cultural biases that may be distributed across multiple interacting dimensions.

6.2 Capability-Bias Trade-off

Steering vectors modify activations, which can degrade general model capability. The magnitude of this trade-off for small language models (3B-8B parameters) is unknown. Larger models have more representational capacity to absorb steering corrections without capability loss; smaller models may be more sensitive.

6.3 The Shared Blind Spot Problem

If the same model that produces biased outputs is used to generate the contrastive pairs for steering vector extraction, the extraction process may inherit the model's blind spots. This is the "shared blind spot" problem documented in the Tractatus incident report of February 2026. Mitigation requires external (human or cross-model) validation of contrastive pair quality.

6.4 Dynamic Cultural Context and Off-Limits Domains

Cultural bias is not static. A model serving a Maori community in Aotearoa needs different cultural calibration than one serving a German community in Bavaria. Steering vectors extracted from one cultural context may not transfer. The per-tenant steering approach (Phase 4) addresses this partially, but the design of tenant-specific contrastive pairs requires cultural expertise that cannot be automated.

More fundamentally, some cultural domains may be structurally off-limits to platform-level steering altogether. In an Aotearoa context, whakapapa (genealogical knowledge), tikanga (customary practice), and kawa (protocol) carry authority that derives from iwi and hapu governance, not from platform architecture. Applying platform-wide steering vectors to representations of these concepts -- even well-intentioned corrections -- risks subordinating indigenous epistemic authority to the platform operator's worldview. For these domains, the correct architectural response may be delegation: the platform provides the steering mechanism, but the definition, calibration, and governance of vectors touching culturally sovereign knowledge must be exercised by the relevant cultural authority, not by the platform's engineering team.

6.5 Who Steers? Governance of Steering Vectors

Steering vectors are instruments of norm enforcement. The technical capability to shift model behaviour along a bias dimension raises immediate questions of institutional governance: whose norms, enacted through what contestable process, with what recourse for those subject to them.

The current proposal embeds steering governance within the Tractatus framework, but does not specify the decision rights for steering operations. A complete governance model should map steering vectors to concrete institutional roles:

Decision	Who Decides	Contestation Path
Define a bias axis (what counts as bias)	Platform operator + community advisory panel	Community deliberation, annual review
Approve a steering vector for deployment	Tractatus BoundaryEnforcer (technical) + tenant moderators (value judgment)	Audit trail of vector provenance, magnitude, and effect
Set vector magnitude (how much correction)	FairSteer dynamic calibration (technical) + human review for sensitive domains	Per-inference logging, threshold alerts
Override or disable a vector	Tenant governance body (for tenant vectors) / platform operator (for platform vectors)	Dispute resolution process with documented rationale
Govern culturally sovereign domains (whakapapa, tikanga, kawa)	Relevant cultural authority (iwi, hapu) -- not platform operator	Independent of platform governance; platform provides mechanism, not authority

This governance structure does not yet exist in the implementation. Phase 4 (per-tenant steering) provides the architectural hooks, but the institutional layer -- who sits on advisory panels, how disputes are escalated, what constitutes sufficient cultural authority for a given domain -- requires community design work that cannot be automated or imposed by the platform operator.

The risk of proceeding without this governance layer is that steering vectors become a new site of centralised value authority: the platform operator decides what bias is and how to correct it, and tenants receive corrections rather than participating in their design. This would reproduce the very power asymmetry that sovereign deployment is intended to disrupt.

6.6 Measurement Difficulty

Unlike the 27027 port incident (binary correct/incorrect), cultural bias is not binary. Evaluating whether a steered model produces "less biased" output requires human judgment, cultural expertise, and longitudinal assessment. The 5-point scoring scale in

the existing evaluation suite provides a starting framework, but its reliability and validity for measuring steering vector effectiveness are untested.

7. Conclusion

The indicator-wiper analogy suggests a useful distinction between biases that operate at the representational level (mechanical, pre-cognitive, analogous to motor patterns) and biases that emerge through reasoning chains. If this distinction holds in transformer architectures -- and the mechanistic interpretability evidence supports it -- then a class of AI biases requires intervention at the activation level rather than the prompt level.

Steering vector techniques (CAA, RepE, FairSteer, DSO, sparse autoencoder feature steering) provide the theoretical and practical toolkit for such intervention. Critically, these techniques require full access to model weights and activations -- access that is available exclusively in sovereign local deployments and architecturally unavailable through commercial API endpoints.

The Village Home AI platform, with its QLoRA-fine-tuned Llama models, two-tier training architecture, and Tractatus governance integration, is structurally positioned to pioneer the application of steering vectors to cultural bias mitigation in community-serving AI. The proposed four-phase implementation path is conservative, empirically grounded, and designed to produce measurable results within a 16-week timeline.

The indicator-wiper problem is solvable. The driver eventually recalibrates. The question for sovereign AI is whether we can accelerate that recalibration -- not by telling the model to “be less biased” (the equivalent of verbal instruction), but by directly adjusting the representations that encode the bias (the equivalent of physical relocation of the indicator stalk).

Since the initial submission of this paper, empirical work by Radhakrishnan et al. (2026) has confirmed at scale what the mechanistic interpretability literature had previously suggested: abstract concepts, including safety-critical behavioural dispositions, are representationally encoded in large language models and are accessible to targeted manipulation through feature-level steering techniques. Critically, the same authors demonstrate that these techniques can override trained refusal behaviours — establishing that the capacity for representational-level model manipulation is now a demonstrated and accessible capability.

This finding transforms the governance stakes of the argument advanced in this paper. The structural advantage of sovereign deployment — full access to model weights and activations — is simultaneously an opportunity and a responsibility. It is an opportunity because it enables the culturally-grounded, community-governed debiasing that this paper proposes. It is a responsibility because that same access, in the absence of robust governance architecture, constitutes a risk surface that is entirely absent from API-mediated deployments. The question is not whether representational steering will be used; the Radhakrishnan et al. results make clear that it already is. The question is whether its use will be governed.

Frameworks such as Tractatus are not merely useful in this environment — they are necessary. Architectural enforcement of governance constraints, MetacognitiveVerifier auditing of steering vector provenance, and community-validated calibration of steering parameters represent the minimum viable governance response to a capability that is now empirically confirmed, publicly documented, and available to any actor with access to open-weight models. The development and adoption of such frameworks warrants immediate priority across the sovereign AI community.

References

Elhage, N., et al. (2022). Toy Models of Superposition. Anthropic.

Li, K., et al. (2023). Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. NeurIPS 2023.

Olsson, C., et al. (2022). In-context Learning and Induction Heads. Anthropic.

Radhakrishnan, A., Beaglehole, D., Belkin, M., & Boix-Adserà, E. (2026). Exposing biases, moods, personalities, and abstract concepts hidden in large language models. *Science.* Published 19 February 2026.

Rimsky, N., et al. (2023). Steering Llama 2 via Contrastive Activation Addition. arXiv:2312.06681.

Stroh, J. (2025). Tractatus: Architectural Enforcement for AI Development Governance. Working Paper v0.1.

Stroh, J. & Claude (2026). From Port Numbers to Value Systems: Pattern Recognition Bias Across AI Domains. STO-RES-0008.

Templeton, A., et al. (2024). Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. Anthropic.

Turner, A., et al. (2023). Activation Addition: Steering Language Models Without Optimization. arXiv:2308.10248.

Zou, A., et al. (2023). Representation Engineering: A Top-Down Approach to AI Transparency. Center for AI Safety.

Licence

— End of Document —