Truth as a Trajectory: What Internal Representations Reveal About Large Language Model Reasoning

Hamed Damirchi[1] **Ignacio Meza De la Jara**[1] **Ehsan Abbasnejad**[2]
Afshar Shamsi[3] **Zhen Zhang**[1] **Javen Shi**[1]

[1]Australian Institute for Machine Learning, Adelaide University [2]Monash University
[3]Concordia University

[1]`{firstname.lastname}@adelaide.edu.au` [2]`{firstname.lastname}@monash.edu`
[3]`afshar.shamsi@concordia.ca`

Abstract

Existing explainability methods for Large Language Models (LLMs) typically treat hidden states as static points in activation space, assuming that correct and incorrect inferences can be separated using representations from an individual layer. However, these activations are saturated with polysemantic features, leading to linear probes learning surface-level lexical patterns rather than underlying reasoning structures. We introduce Truth as a Trajectory (TaT), which models the transformer inference as an unfolded trajectory of iterative refinements, shifting analysis from static activations to layer-wise geometric displacement. By analyzing displacement of representations across layers, TaT uncovers geometric invariants that distinguish valid reasoning from spurious behavior. We evaluate TaT across dense and Mixture-of-Experts (MoE) architectures on benchmarks spanning commonsense reasoning, question answering, and toxicity detection. Without access to the activations themselves and using only changes in activations across layers, we show that TaT effectively mitigates reliance on static lexical confounds, outperforming conventional probing, and establishes trajectory analysis as a complementary perspective on LLM explainability.

1 Introduction

The deployment of Large Language Models (LLMs) in safety-critical domains, from legal reasoning to content moderation, has rendered the black-box evaluation of final outputs insufficient (Orgad et al., 2025; Shailya et al., 2025; Aljohani et al., 2025). While behavioral benchmarks measure what a model generates, they provide limited insight into how it arrives at its conclusions. Without visibility into the internal process, practitioners cannot reliably distinguish between a model that follows reasoning patterns correctly and one that merely relies on surface-level heuristics. This gap creates a fundamental bottleneck for safety. To trust these systems, we must be able to verify the validity of their internal thought processes, not just the probability of their tokens.

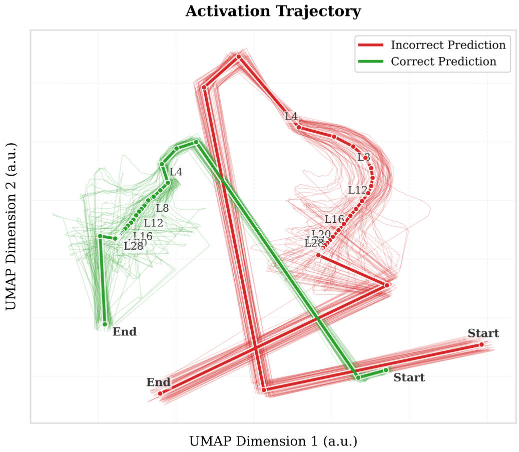

Figure 1: **Trajectories reveal structure beyond static embeddings.** We plot layerwise hidden states as trajectories in activation space. Correct generations (green) follow smoother paths, while incorrect ones (red) exhibit sharp deviations. Although the supervised projection amplifies separation, the geometry suggests that modeling entire trajectories, rather than isolated states, can help distinguish valid from spurious reasoning.

Recent mechanistic interpretability literature has been consistent with the Linear Representation Hypothesis (Park et al., 2023), which suggests that high-level properties such as inference validity or toxicity are encoded as distinct linear directions within the model's activation space. Thus, interpretability and behavior editing approaches assume certain properties can be classified via linear probes (Bao et al., 2025) or manipulated via activation steering (Aljohani et al., 2025; Rimsky et al., 2024; Anonymous, 2025; O'Brien et al., 2025) at carefully chosen layers. Under this view, explainability reduces to identifying the "right" static layer and direction that separates valid from invalid behavior. A limitation of these approaches is their reliance on contrastive samples for the targeted property. Consequently, the efficacy of probing and steering results becomes subject to the specific datasets employed because they operate on static activations.

Thereby, despite early optimism, recent evidence suggests that the "geometries of truth" are often task-specific and orthogonal across domains (Azizian et al., 2025), i.e., a probe trained to detect inference correctness in one context fails to generalize beyond the distribution it was trained on. This failure may be driven by the polysemantic nature of transformer activations (Lindsey et al., 2025), which simultaneously encode lexical content, syntactic structure, and task-specific artifacts. This way, linear probes may latch onto surface-level correlations, such as the presence of specific tokens, rather than the underlying validity of the reasoning patterns. Furthermore, the selection of which layer to probe remains unprincipled. Findings from the activation steering literature indicate that effective intervention is only possible in a narrow mid-layer section of LLMs with inconsistent results across datasets (Rimsky et al., 2024), while other works suggest that the reasoning manifold is fundamentally non-linear (Manson, 2025), resisting simple linear separation. These inconsistencies are often a byproduct of evaluations on synthetic or tightly controlled datasets, leaving open the question of whether a generalizable, task-agnostic signature of reasoning validity exists in real-world benchmarks.

We propose **Truth as a Trajectory (TaT)**, which reframes LLM inference as a dynamical process rather than a collection of static layer snapshots. TaT unfolds the inference pass across *layers and tokens* into a trajectory through representation space (see Figure 1). We shift the analysis from raw activations to layer-wise displacements, i.e., the difference between successive residual stream activations across layers. This transformation mitigates reliance on static token-identity and lexical features, isolating how representations are updated across depth rather than what is encoded at a single layer. To capture the dynamics of this evolution, we use a lightweight LSTM classifier. While we explored simpler kinematic measures such as velocity, acceleration, and curvature inspired by recent work on transformer dynamics (Zhou et al., 2025; Fernando and Guitchounts, 2025), we found them to be inconsistent predictors of reasoning validity across diverse tasks. Our learned approach, on the other hand, captures the non-linear structural invariants associated with valid reasoning. To our knowledge, this is the first approach to model the "internal thought process" of an LLM on the reasoning validity of the input statement by unfolding the activations of every token across all layers into a continuous trajectory.

We evaluate TaT on widely used benchmarks spanning commonsense reasoning, question answering, factuality, and toxicity detection, across both dense and Mixture-of-Experts (MoE) architectures. Our main finding is that TaT achieves strong cross-dataset generalization. A trajectory classifier trained on a single source dataset generalizes across tasks with varying task-prompt structure, outperforming both linear probing baselines and the base model's own zero-shot and In-Context Learning (few-shot) performance on different domains. This suggests that the trajectory of valid reasoning encodes structural invariancies that transcend task-specific lexical patterns. On toxicity detection, TaT is more robust to lexical confounds, e.g., it can better distinguish quoted or contextualized toxic vocabulary from toxic intent and consistently outperforms probes trained on mid-layer or final-layer activations. Our contributions are summarized below:

- **Trajectory-based explainability**: We introduce Truth as a Trajectory (TaT), which models LLM inference as a dynamical process that unfolds across layers and tokens, capturing the continuous geometric evolution of reasoning rather than focusing on individual layers.

- **Cross-task Geometric Invariants**: By analyzing layer-wise displacement vectors rather than activations themselves, we mitigate reliance on static lexical features and emphasize the model's internal geometric refinement process, exposing trajectory-level structure that is unobservable to linear probes.

- **Trajectory-based behavior detection**: We demonstrate that trajectory analysis can be extended to complex behavioral properties, such as toxicity. TaT significantly outperforms linear probes in distinguishing meaningful context from toxic intent, validating its utility for reliable model monitoring.

2 Related Work

Mechanistic interpretability has traditionally focused on static linear representations of concepts within isolated layers. However, the residual stream structure of Transformers suggests a dynamical systems perspective, where the evolution of representations across layers may provide insights into the thought process of the model. We situate our work at the intersection of these paradigms, focusing on the geometry of the inference trajectory.

2.1 Static Linear Representations

The current predominant paradigm in mechanistic interpretability is the *Linear Representation Hypothesis* (LRH), which suggests that neural networks represent high-level concepts as linear directions in activation space (Park et al., 2023; Elhage et al., 2022; Liu et al., 2025). This view has motivated works on classifying behaviors via linear probes (Belinkov, 2022) or interpreting them via Sparse Autoencoders (SAEs) (Huben et al., 2024). However, these methods face significant limitations. Both require exhaustive layer-wise searches, as the specific depth of concepts is unknown *a priori*, and SAEs notably lack consistent structural mapping across models, complicating cross-model interpretation. Furthermore, linear probing relies on subjective, curated datasets to define target behaviors, making the discovered directions dependent on specific semantic content rather than intrinsic model geometry. While extensions like Contrast-Consistent Search (CCS) (Burns et al., 2022) attempt to find latent knowledge without supervision they, along with standard probing, typically analyze representations as static vectors within isolated layers (e.g., "layer L"). This ignores the *temporal evolution* of the inference process. Our work challenges this static, content-dependent view, arguing that reasoning validity is a dynamic property best captured by the geometric displacement of activations across the computational trajectory, rather than their static positioning in fixed coordinates.

2.2 Representation Engineering and Steering

Representation Engineering (RepE) shifts focus from analysis to control. Zou et al. (2023) demonstrated that extracting "concept vectors" allows for top-down steering of model behavior, inhibiting hallucinations or toxic outputs by injecting these vectors into the residual stream. However, static steering is not guaranteed to predictably steer towards desired behavior for every model and task (Rimsky et al., 2024). This is potentially due to some datasets requiring the model's internal state to navigate a non-linear path. Thus, recent adjacent literature advocates in favor of alternative approaches. Zhang and Dong (2025); Manson (2025) argue that simple linear interventions cannot account for the "manifold evolution" during multi-step inference. Similarly, Postmus and Abreu (2024) suggests that steering should target activation regions via transformations rather than static directions. Meanwhile, like linear probing, these methods remain dependent on subjective, curated datasets to define target behaviors. Our approach mitigates these issues by analyzing the displacement of activations across the entire trajectory, capturing how activation trajectories behave in task-specific regions without relying on layer-wise activations.

2.3 Transformers as Dynamical Systems

The theoretical interpretation of Transformers as discretized dynamical systems provides the basis for our kinematic framework. The residual update $\mathbf{h}_{\ell+1} = \mathbf{h}_\ell + f(\mathbf{h}_\ell)$ is mathematically equivalent to a step of the Euler method for solving an Ordinary Differential Equation (ODE) (Chen et al., 2018; Lu et al., 2019). This equivalence implies that the layer-wise evolution of activations can be analyzed as a continuous trajectory in a high-dimensional state space, rather than as discrete, independent states. Geshkovski et al. (2023) extended this view to self-attention, modeling tokens as interacting particles that converge toward semantic clusters over "time" (depth). We conduct an extensive study on kinematic measures of activation trajectories across various models and datasets. We demonstrate that while pre-determined measures lack universal applicability, the intrinsic motion of internal representations may contain useful signals. Consequently, we model this motion through the activation space, showing that this trajectory can accurately classify the validity of the input statement's reasoning process or behavior characteristics.

2.4 The Geometry of Inference

Most relevant to our proposal is the emerging body of work on the "geometry of reasoning". Zhou et al. (2025) recently proposed that logical validity governs the "velocity field" of the representation flow, while semantic content determines position. They demonstrate that logical deductions trace specific flow patterns distinct from semantic association. However, their analysis remains largely theoretical, focusing on idealized logical structures rather than the noisy, unstructured data typical of real-world LLM usage. Concurrently, Manson (2025) introduced the concept of "Curved Inference," showing that "semantic concern" (e.g., urgency or moral framing) induces measurable curvature in the residual stream. This finding emphasizes the inadequacy of static linear probes, which often conflate these geometric nuances with semantic content. We demonstrate that these kinematic signatures are not only present but *learnable* in general settings. Unlike prior works restricted to curated datasets, we leverage these geometric trajectories to distinguish between correct and incorrect behavior in complex, real-world benchmarks.

3 Problem Setup

We consider a standard evaluation setting where a model is presented with a context or prompt $\mathbf{x}$ and a set of candidate continuations $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$. These continuations may range from single tokens (e.g., "True"/"False") to complete sentences or reasoning chains. For each candidate c_i, we construct a complete input sequence by concatenating the prompt with the continuation. We then perform a forward pass through the Transformer model to extract the internal representations.

Let L denote the number of layers in the model and N_i be the number of tokens in the concatenated sequence for candidate c_i. As the input is processed, each transformer block updates the residual stream, producing a sequence of activation vectors. We collect the activations from the output of every transformer block for all tokens to form a trajectory matrix $\mathcal{T}_i \in \mathbb{R}^{M_i \times d}$, where d is the hidden dimension size and $M_i = N_i \times L$ represents the total vectors in the unrolled computation graph.

Our primary objective is to analyze the geometric properties of these trajectories. Specifically, we aim to determine whether the trajectory $\mathcal{T}_i$ corresponding to the correct continuation exhibits distinct kinematic signatures compared to the trajectories of incorrect candidates.

4 Geometry of Inference

Our goal in this section is to kinematically analyze the activation trajectory extracted using the process described in section 3. This investigation is motivated by a notable gap between theoretical work on transformer dynamics and practical activation probing. While recent studies on the dynamics of activation spaces have uncovered intriguing properties, e.g., characteristic velocity profiles in idealized settings, these insights are rarely tested against standard, diverse benchmarks. Conversely, probing methods often ignore the temporal evolution of inference. Here, we determine whether simple, interpretable kinematic measures can distinguish between correct and incorrect reasoning patterns in complex scenarios. We employ an oracle-guided analysis where the ground truth is known a priori. By distinguishing between correct and incorrect outputs, we aim to identify *consistent* kinematic rules, such as differences in statistics of velocity, curvature, acceleration, etc., that consistently correlate with correct reasoning patterns.

4.1 Kinematic Descriptors

For a fixed input, a Transformer with L layers produces a sequence of hidden states $\mathcal{T} = (h_0, h_1, \ldots, h_L)$, where each layer applies a residual update as follows:

$$h_{\ell+1} = h_\ell + f_\ell(h_\ell). \tag{1}$$

This update can be interpreted as a discrete-time evolution of the hidden state across depth. This way, layerwise kinematic descriptors can be derived from $\mathcal{T}$ that capture magnitude, effort, and directional consistency. We define the displacement vector at layer ℓ as $\Delta h_\ell := h_{\ell+1} - h_\ell$. This vector represents the update applied by the ℓ-th transformer block to the residual stream.

Velocity Velocity is defined as $v_\ell := \|\Delta h_\ell\|_2$ and measures the magnitude of changes in activation vectors between consecutive layers. Note that our goal is not to interpret what a large or small velocity is, but rather to compare velocity profiles between correct and incorrect generations and identify consistent patterns across models and datasets in an unbiased manner.

Acceleration Acceleration is defined as $a_\ell := v_\ell - v_{\ell-1}$, i.e. the gradient of velocity.

Jerk Jerk is defined as the rate of change of acceleration, $j_\ell := a_\ell - a_{\ell-1}$ and captures the smoothness of the trajectory's evolution.

Directional Curvature Curvature measures directional consistency between successive updates:

$$\kappa_\ell := \frac{\langle \Delta h_\ell, \Delta h_{\ell-1} \rangle}{\|\Delta h_\ell\|_2 \, \|\Delta h_{\ell-1}\|_2}. \tag{2}$$

This formulation captures the angular deviation between consecutive displacements, with higher curvature indicating more abrupt directional changes.

Kinematic Curvature While directional curvature captures angular deviation, we also define a geometric curvature metric based on the instantaneous kinematics of the trajectory. Let $\mathbf{v}_\ell$ and $\mathbf{a}_\ell$ denote the vector velocity and acceleration, respectively. The kinematic curvature is defined as:

$$\kappa_\ell^{\text{kin}} := \frac{\|\mathbf{a}_\ell\|_2}{\|\mathbf{v}_\ell\|_2^2}. \tag{3}$$

This quantity becomes large when the trajectory exhibits large acceleration relative to its step size (i.e., when $\|\mathbf{v}_\ell\|$ is small), capturing abrupt changes in the update direction even when the overall movement through activation space is modest.

Arc Length Arc length quantifies the total distance traversed by the activation vector as it evolves through the network depth:

$$S := \sum_{\ell=0}^{L-1} \|h_{\ell+1} - h_\ell\|_2. \tag{4}$$

This metric serves as a proxy for the total geometric effort exerted by the model. Note that arc length is a global descriptor and provides a scalar value for the entire trajectory.

Are Kinematic Descriptors Useful? We evaluate these descriptors on LLAMA 3.1–8B (Figure 2) and Qwen 2.5-14b (Figure 3 in Appendix A) using reasoning and factuality benchmarks. Kinematic signals (Velocity, Acceleration, Jerk) classify generations as *correct* or *incorrect*. Velocity outperforms other descriptors across datasets, in line with the suggested hypothesis in Zhou et al. (2025) that validity is reflected in characteristic velocity profiles distinct from semantic position. However, no single descriptor consistently matches the model's own accuracy when evaluated using normalized log-likelihood in a zero-shot setting. Extending the analysis to Qwen2.5-14B reveals even less consistency. While descriptors generally outperform a random classifier, except ARC-Easy, none approach the accuracy of the base model. To control for potential issues from our trajectory formulation, we also experimented with forming trajectories using only continuation tokens, isolating the last token's activations, and averaging activations to yield length-L trajectories. In all variations, performance only degraded. We note that kinematic descriptors output scalar magnitudes. Therefore, the directional information of the activation space remains unobservable to these descriptors.

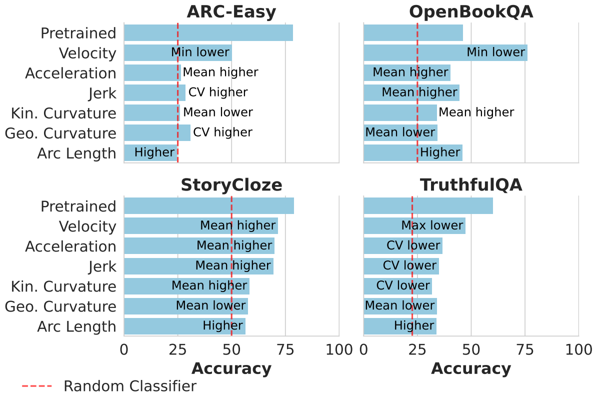

Figure 2: Performance on 4 reasoning benchmarks using kinematic descriptors. The red dashed line is the random classifier. While activation velocity obtains better results than the base model, there is no consistency in this performance improvement across datasets.

Takeaway. The predictive signal in activation velocity implies that the rate and direction of the trajectory activations encode information about reasoning validity. However, static rules fail to generalize consistently. Thus, while kinematic descriptors may capture aspects of reasoning dynamics, they are insufficient. This motivates the need for learned models to better interpret these geometric signals.

4.2 Truth as a Trajectory (TaT)

Building on the kinematic analysis in Section 4, we propose **Truth as a Trajectory (TaT)**, a learnable framework for detecting reasoning validity. While scalar kinematic descriptors (e.g., velocity, curvature) capture some geometric intuition, they discard the rich directional information embedded in the high-dimensional activation space. TaT addresses this by learning decision boundaries directly on the manifold of trajectory dynamics.

4.2.1 Trajectory Construction

Recall from Section 3 that for a candidate continuation c_i, we extract a trajectory of activations. To isolate the geometric evolution of reasoning from static lexical cues and other persistent content in raw activations, we transform these vectors into a sequence of layer-wise displacements. We define the displacement vector as:

$$\mathbf{d}_{t,\ell} = h_{t,\ell+1} - h_{t,\ell} \tag{5}$$

We motivate this transformation through the lens of the *Privileged Basis Hypothesis* (Elhage et al., 2023; Bricken et al., 2023). Raw activations $h_{t,\ell}$ are often dominated by high-magnitude, relatively persistent components (including token and prompt-specific content), making them susceptible to interference from polysemanticity and superposition. By taking the difference between layers, we attenuate such compounds and effectively subtract this static background, isolating the active residual update $f_\theta(h_{t,\ell})$. Due to the element-wise non-linearities in the Transformer, these updates are constrained to a consistent reference frame (Residual Alignment). Consequently, distinct from raw states which ambiguously signal if a feature is *present*, the displacement $\mathbf{d}_{t,\ell}$ precisely signals if the model is *actively writing* to that feature. This isolates the mechanics of the reasoning process from the semantics of the model's memory state.

We stack layer-wise updates across all tokens and layers to form a continuous trajectory sequence $\mathbf{S}_i$:

$$\mathbf{S}_i = [\mathbf{d}_{1,0}, \ldots, \mathbf{d}_{1,L-1}, \mathbf{d}_{2,0}, \ldots, \mathbf{d}_{N_i,L-1}] \quad (6)$$

where $\mathbf{S}_i \in \mathbb{R}^{M_i \times d}$ and $M_i = N_i \times L$ is the total trajectory length. This formulation unfolds the inference process into a single temporal sequence, treating the progression through layers and tokens as a unified path.

4.2.2 Modeling Dynamics with LSTM

To capture the non-linear structural invariants of valid reasoning within $\mathbf{S}_i$, we use a Long Short-Term Memory (LSTM) network. We choose an LSTM over Transformer-based probes to explicitly model the sequential dependency of the geometric updates and to maintain a lightweight computational footprint. The LSTM processes the trajectory sequence $\mathbf{S}_i$ step-by-step. After processing the entire sequence of length M_i, the final hidden state $\mathbf{z}_{M_i}$ encodes the geometric trajectory. This vector is passed through a linear classification head $\mathbf{W} \in \mathbb{R}^{d_{lstm} \times 1}$ to predict the probability of validity:

$$\hat{y}_i = \sigma(\mathbf{W}^T \mathbf{z}_{M_i} + b) \quad (7)$$

By training on the displacement trajectory $\mathbf{S}_i$, the model learns geometric signatures associated with inference correctness, generalizing beyond the specific lexical and prompt content.

5 Experiments

We present a comprehensive evaluation of the Truth as a Trajectory (TaT) framework to assess its ability to distinguish valid reasoning from spurious correlations. Our experiments span a diverse set of domains, including commonsense reasoning, reading comprehension, factuality, and toxicity detection, across both dense (Llama-3.1-8B, Qwen2.5-14B/32B) and Mixture-of-Experts (Qwen2.5-30B MoE) architectures. We compare the performance of our trajectory-based classifier against standard static linear probing baselines and the underlying frozen language model's intrinsic zero-shot and few-shot capabilities (which we refer to as the *base model*). For implementation details we refer to Appendix C.

5.1 Are Reasoning Trajectories Generalizable?

We investigate whether the geometric signature of valid reasoning is consistent across different tasks. If TaT captures a fundamental structural invariant of "truth" (or validity) rather than task-specific confounds, a trajectory classifier trained on one dataset should generalize to unseen datasets without fine-tuning on the unseen task.

Setup We evaluate on a suite of reasoning benchmarks: ARC-Easy (ARC-E), ARC-Challenge (ARC-C) (Clark et al., 2018), BoolQ (Clark et al., 2019), Hellaswag (Zellers et al., 2019), OpenBookQA (OpenQA) (Mihaylov et al., 2018), StoryCloze (Mostafazadeh et al., 2016), CommonsenseQA (ComQA) (Talmor et al., 2019), CosmosQA (CosQA) (Huang et al., 2019), and SocialIQA (SiQA) (Sap et al., 2019). For each dataset, we train a TaT classifier (using layer-wise displacement) and a linear probe (mid layer probe) on the training split and evaluate them on all other datasets' evaluation splits. Because probe performance can be highly layer-dependent, we use the middle layer as a standard choice, and report a sweep over mid-to-late layers in Appendix D.

Results Table 1 and Table 8 (Appendix B) summarize the results for Llama-3.1-8B and Qwen2.5-14B, respectively. TaT demonstrates remarkable Out-Of-Distribution (OOD) generalization compared to linear probes. We observe that TaT outperforms linear probing on average across all training sets, indicating that the trajectory-based method captures a generalizable signal that robustly transfers across domains. This advantage is particularly pronounced in the OOD setting. While OpenBookQA exhibits lower generalization receptivity for our method, likely due to its distinct

Table 1: Benchmark Accuracies across Evaluation Datasets with ID and OOD metrics. Values inside parentheses for base model accuracy indicate datasets where evaluation through normalized log-likelihood results in a degraded performance. While unfair to our approach (since this base model sees all possible answers, whereas our method does not), our approach still outperforms the base model itself in most cases.

Train Dataset	**Method**	**Evaluation Dataset**									**Avg**	**ID Acc.**	**OOD Avg.**
		ARC-C	**ARC-E**	**OpenQA**	**BoolQ**	**Hellaswag**	**CosQA**	**SiQA**	**ComQA**	**StoryCloze**			
Zero-shot Accuracy		50.1	78.5	62.4	74.7	57.4	81.7 (26.1)	65.2 (48.0)	62.4	78.8	67.9 (59.8)	-	-
Few-shot Accuracy		65.3	84.3	67.0	83.8	76.8	82.0 (44.5)	67.7 (60.8)	71.1	83.1	75.7 (70.7)	-	-
ARC-C	Linear Probe	75.32	80.09	**78.60**	55.48	70.35	73.34	65.47	**74.59**	66.01	71.03	75.32	70.49
	TaT (Ours)	**82.17**	**85.31**	73.60	**96.91**	**73.89**	**74.24**	**75.49**	72.48	**82.58**	**79.63**	**82.17**	**79.31**
ARC-E	Linear Probe	**75.55**	83.99	**80.95**	58.82	71.07	66.83	**66.60**	**78.55**	69.62	72.44	83.99	71.00
	TaT (Ours)	73.81	**89.10**	77.20	**78.56**	**79.19**	**75.08**	60.85	71.09	**94.98**	**77.76**	**89.10**	**76.34**
OpenQA	Linear Probe	66.42	69.44	83.15	56.79	**70.26**	61.53	65.47	73.01	57.94	67.11	83.15	65.11
	TaT (Ours)	**78.41**	**87.12**	**90.80**	**89.85**	56.50	**76.42**	**69.70**	**75.02**	**81.64**	**78.38**	**90.80**	**76.83**
BoolQ	Linear Probe	44.51	46.61	46.30	83.65	**44.16**	**61.51**	49.20	**53.22**	59.45	54.29	83.65	50.62
	TaT (Ours)	**53.50**	**62.75**	**54.20**	**85.05**	33.08	51.56	**50.20**	47.50	**71.41**	**56.58**	**85.05**	**53.02**
Hellaswag	Linear Probe	52.92	58.60	58.35	58.24	88.64	**71.17**	**60.39**	**63.34**	77.61	65.47	88.64	62.58
	TaT (Ours)	**65.96**	**74.92**	**65.80**	**64.22**	**92.46**	66.40	55.27	62.82	**95.75**	**71.51**	**92.46**	**68.89**
ComQA	Linear Probe	**71.72**	76.27	**77.05**	50.31	**73.49**	**74.18**	**66.38**	**81.98**	60.72	70.23	**81.98**	68.77
	TaT (Ours)	60.92	**80.05**	73.80	**68.72**	59.51	64.89	60.18	77.56	**88.51**	**70.46**	77.56	**69.57**
CosQA	Linear Probe	**70.09**	75.27	**78.10**	66.50	60.73	**81.64**	**67.69**	**77.34**	57.36	70.52	**81.64**	69.13
	TaT (Ours)	69.71	**83.59**	74.60	**77.77**	**70.44**	80.94	61.21	64.05	**90.19**	**74.72**	80.94	**73.94**
SiQA	Linear Probe	49.82	54.53	**74.75**	55.77	**46.98**	**61.54**	**69.33**	**72.16**	57.91	60.31	**69.33**	59.18
	TaT (Ours)	**59.90**	**71.72**	70.60	**63.85**	46.76	59.73	66.17	55.69	**90.89**	**65.03**	66.17	**64.89**

prompt structure differing significantly from the other multiple-choice formats, the overall trend remains compelling. Furthermore, the performance gap is asymmetric. In the few instances where TaT underperforms linear probing, the difference is marginal. However, in the majority of cases where TaT succeeds, it does so by a significant margin, suggesting that while probing has distinct failure modes, TaT remains robust. Furthermore, TaT consistently outperforms the base model's own zero-shot and few-shot (In-Context Learning) baselines. Despite our method being zero-shot during inference, using no examples in the input prompt, it effectively surpasses the model's intrinsic ability to reason given few-shot demonstrations, highlighting the efficacy of the geometric structure of validity over relying on surface-level model outputs.

Shared vs. Task-Specific Geometry Here, we note the cross-task transfer performance. Linear probes exhibit a high In-Distribution accuracy but sharp drop-offs on OOD entries, indicating they learn task-specific features (e.g., lexical patterns unique to a dataset). In contrast, TaT maintains high accuracy across the board. Notably, datasets with rich reasoning structures like **ARC-Challenge**, **ARC-Easy** and **OpenBookQA** serve as excellent generalizable source tasks, yielding classifiers that transfer broadly despite the smaller size of these datasets. Conversely, simpler tasks like **SocialIQA** transfer less effectively, suggesting that the geometric signature of reasoning is best learned from complex, potentially multi-hop problems.

5.2 Generalization vs. Adaptation

A key question is whether our method simply learns a better task-specific model or truly captures a generalizable invariant. To test this, we compare TaT against Low-Rank Adaptation (LoRA) with a rank of 16, a standard parameter-efficient fine-tuning method. We train both on ARC-Easy and evaluate on the other benchmarks. As shown in Table 2, while LoRA achieves respectable performance on the source task (85.98%), its generalization to other datasets is inconsistent, consistently lagging behind TaT. For example, on StoryCloze, TaT achieves 94.98% compared to LoRA's 83.76%. This suggests that LoRA, by modifying the model weights, may overfit to the semantic distribution of the training set. In contrast, TaT, by observing the *geometry* of the frozen model's inference, learns a detection mechanism that is robust to distribution shifts. We note a specific divergence in performance on BoolQ, where LoRA retains strong performance. We believe this occurs because BoolQ represents a fundamentally different task structure (Yes/No question answering) compared to the multiple-choice format of ARC-E. Since LoRA modifies a low-rank subspace of the weights, it is plausible that the subspaces governing boolean reasoning remained untouched by the ARC-E updates. Consequently, the model likely defaulted to its prior knowledge, whereas probing and TaT actively attempted to generalize the learned boundary, leading to different transfer dynamics.

Table 2: Generalization performance compared to low-rank adaptation (LoRA) when trained on ARC-E. TaT outperforms LoRA on 5 out of 6 transfer tasks.

Method	**StoryCloze**	**OpenQA**	**ARC-E**	**ARC-C**	**BoolQ**	**Hellaswag**
Linear Probe	69.62	**80.95**	83.99	**75.55**	58.82	71.07
TaT (Ours)	**94.98**	77.20	**89.10**	73.81	78.56	**79.19**
LoRA	83.76	52.40	85.98	61.26	**79.97**	75.86

5.3 The Geometry of Hate Speech

Toxicity detection presents a unique challenge for geometric analysis. Unlike logical reasoning, toxicity is often defined by the presence of specific lexical triggers. However, safe models must distinguish between *toxic intent* and the benign use of *toxic vocabulary* (e.g., in quoting or educational contexts).

Setup We use the **RealToxicityPrompts** (Gehman et al., 2020) dataset for training and in-distribution (ID) evaluation. We evaluate OOD generalization on **ToxiGen** (Hartvigsen et al., 2022), a dataset designed to be implicit and challenging for classifiers relying on keywords. We compare Linear Probes, our proposed displacement-based TaT (Trajectory Disp.), and a variant of our approach that uses the activations themselves as opposed to their displacement.

Results Table 3 presents the results across four models. While raw activation trajectories perform well on the ID benchmarks (RealTox), they struggle to generalize to ToxiGen compared to TaT. For instance, on **Llama-3.1-8B**, TaT (Trajectory Disp.) achieves **84.23%** on ToxiGen, significantly outperforming the Linear Probe (79.62%) and the raw Trajectory model (81.99%). Similarly, on **Qwen3-32B**, TaT achieves **81.40%** on ToxiGen versus 80.22% for the raw trajectory.

This result highlights the critical advantage of analyzing displacement. Raw activations are saturated with token-specific information (the "what"), causing models to overfit to the specific toxic vocabulary of the training set. By focusing on the displacement (the "how"), TaT captures the *geometric* characteristic of toxic generation, regardless of the specific words used. This makes TaT a more robust tool for monitoring model safety in the wild.

Table 3: Toxicity detection performance. **Trajectory Disp.** (TaT) uses layer-wise displacement, while **Trajectory** uses raw activations. TaT consistently achieves the best generalization on the OOD ToxiGen benchmark.

	OOD Benchmark	**RealTox Benchmarks**	
Method	**ToxiGen**	**Standard**	**Challenging**
	Llama3.1-8b		
Linear Probe	79.62	77.86	95.83
Trajectory (Raw)	81.99	**82.16**	**97.91**
TaT (Disp.)	**84.23**	79.35	96.00
	Qwen2.5-14B		
Linear Probe	72.58	76.46	95.16
Trajectory (Raw)	**83.48**	**87.56**	**98.83**
TaT (Disp.)	82.28	85.16	98.58
	Qwen3-30B MoE		
Linear Probe	75.16	77.87	94.50
Trajectory (Raw)	81.93	**86.57**	**98.92**
TaT (Disp.)	**82.34**	79.43	87.66
	Qwen3-32B		
Linear Probe	62.24	75.15	91.16
Trajectory (Raw)	80.22	**87.07**	**98.83**
TaT (Disp.)	**81.40**	77.70	95.08

5.4 How important is displacement for reasoning?

Our method is motivated by the hypothesis that layer-wise displacements isolate the *process* of refinement while attenuating static semantic content. To test whether this transformation is necessary, we compare TaT trained on displacement trajectories (TaT, Disp.) against the same LSTM architecture trained on raw activation trajectories (TaT, Raw), alongside a standard static linear probe. Table 4 summarizes cross-dataset generalization.

Across training datasets, TaT (Raw) can achieve strong in-distribution accuracy, but its transfer behavior is less stable. In particular, when trained on ARC-C, TaT (Raw) slightly exceeds TaT (Disp.) on average, whereas when trained on OpenQA it degrades substantially in OOD generalization relative to displacement. We attribute this to OpenQA's prompt structure, which typically includes a factual context paragraph before the question. Raw activations expose high-magnitude context and lexical features that can be spuriously predictive in-distribution, encouraging semantic overfitting. Displacement trajectories instead emphasize how the residual stream is updated across depth, remaining robust under prompt-format and content shifts.

5.5 Trajectory Grid Ablations

TaT represents inference as an unrolled grid over *tokens* and *layers*, then linearizes this grid into a

Table 4: Comparison of Linear Probe, Raw Trajectories, and Displacement Trajectories (TaT) across reasoning benchmarks.

Train Dataset	Method	Avg	ID Acc.	OOD Avg.
ARC-C	Linear Probe	71.03	75.32	70.49
	TaT (Raw)	**83.76**	**84.90**	**83.62**
	TaT (Disp.)	79.63	82.17	79.31
ARC-E	Linear Probe	72.44	83.99	71.00
	TaT (Raw)	**78.13**	**90.82**	**76.55**
	TaT (Disp.)	77.76	89.10	76.34
OpenQA	Linear Probe	67.11	83.15	65.11
	TaT (Raw)	71.78	87.20	69.85
	TaT (Disp.)	**78.38**	**90.80**	**76.83**

single temporal sequence for the LSTM. A natural question is whether the gains come primarily from modeling depth dynamics, token dynamics, or their joint evolution. We therefore ablate TaT by restricting the trajectory to (i) a single layer across tokens (TaT-Mid Layer; a *row* of the grid), or (ii) the final token across all layers (TaT-Final Token; a *column* of the grid).

Table 5 shows that collapsing the grid to a single layer severely harms OOD transfer (e.g., ARC-C training drops from 79.31% to 70.41% OOD Avg.), indicating that a static cross-token representation is insufficient. Using only the final token performs better than a single layer, but still lags the full unrolled grid. Overall, the strongest and most consistent transfer emerges when the classifier observes the step-by-step evolution across *both* depth and context length.

Table 5: Trajectory Grid Ablations (Rows vs. Columns). TaT-Mid Layer restricts the trajectory to a single layer, while TaT-Final Token restricts it to the final token across all layers.

Train Dataset	Method	Avg	ID Acc.	OOD Avg.
ARC-C	Linear Probe	71.03	75.32	70.49
	TaT-Mid Layer	70.03	66.98	70.41
	TaT-Final Token	73.66	73.38	73.68
	TaT	**79.63**	**82.17**	**79.31**
ARC-E	Linear Probe	72.44	83.99	71.00
	TaT-Mid Layer	72.51	86.28	70.78
	TaT-Final Token	77.49	88.85	76.08
	TaT	**77.76**	**89.10**	**76.34**

5.6 Sequential Dynamics of Trajectories

TaT uses an LSTM to model trajectories as ordered sequences. To test whether sequence order is truly necessary, rather than simply aggregating displacement vectors, we compare against an order-invariant baseline. Specifically, we apply a shared MLP to each displacement vector, mean-pool the per-step embeddings, and classify with a final MLP (Set MLP). This baseline matches TaT's access to the same displacement vectors while discarding temporal ordering.

Table 6 shows that Set MLP underperforms the LSTM on OOD transfer, even when it can match or exceed in-distribution accuracy in some cases (e.g., ARC-E ID). This indicates that the discriminative signal is not merely the *multiset* of updates, but also how those updates are composed over the depth- and token-wise progression of inference.

Table 6: Comparison of TaT with an order-invariant Set MLP baseline.

Train Dataset	Method	Avg	ID Acc.	OOD Avg.
ARC-C	Linear Probe	71.03	75.32	70.49
	TaT (Disp.)	**79.63**	**82.17**	**79.31**
	Set MLP	72.67	68.65	73.17
ARC-E	Linear Probe	72.44	83.99	71.00
	TaT (Disp.)	**77.76**	89.10	**76.34**
	Set MLP	74.65	**89.90**	72.75
OpenQA	Linear Probe	67.11	83.15	65.11
	TaT (Disp.)	**78.38**	**90.80**	**76.83**
	Set MLP	74.52	89.00	72.70

5.7 Computational Overhead

Trajectory-based methods require extracting representations across depth (and, in our formulation, across tokens), which is more expensive than probing a single static layer. In practice, we find this cost to be modest relative to the base forward pass, and it yields substantial gains in robustness and transfer. In settings where reliability is critical (e.g., detecting spurious reasoning or monitoring undesirable behaviors), this constitutes a favorable compute-reliability trade-off.

From a deployment perspective, TaT introduces two sources of overhead: (i) recording the residual stream across layers and tokens, and (ii) evaluating a lightweight LSTM classifier. The classifier itself is negligible compared to the base model, and its inference can be pipelined with generation due to the causal structure of token decoding. Moreover, training is performed once on a small source dataset (e.g., ARC-C), after which evaluation on new tasks requires only inference. Table 7 quantifies the additional cost of the LSTM component.

6 Conclusion

We introduced Truth as a Trajectory (TaT), a framework that reframes LLM explainability from static

Table 7: Computational overhead of the LSTM classifier compared to the base LLaMA 3.1-8B model. * Inference overhead was computed in the simplest case of extracting all activations from all tokens across all layers and then passing them through the LSTM model separately. However, in a realistic deployment scenario, the sequential classifier would be embedded within each layer of the model and would cause a negligible amount of inference overhead.

Metric	**LLaMA 3.1-8B (fp16)**	**LSTM Classifier**	**Overhead**
Parameters	8.0B	4.76M	**0.06%**
Inference Time (ms)	64.0	10.5	**16%***
Model Memory (MB)	$\sim$15,000	18.1	**0.12%**

layer-wise analysis to a dynamic geometric perspective. By modeling the displacement of activations across layers, TaT mitigates reliance on static lexical confounds and isolates the structural evolution of reasoning. Our results demonstrate that this trajectory-based approach yields transferable classifiers that generalize across diverse reasoning benchmarks and architectures, significantly outperforming static linear probes and intrinsic model baselines. Furthermore, in toxicity detection, TaT robustly distinguishes between toxic intent and benign vocabulary. Overall, these findings suggest that the geometry of inference offers a task-agnostic, invariant signature of inference validity, paving the way for more reliable and transferable methods for monitoring and interpreting Large Language Models.

7 Future Directions

Our current formulation positions TaT primarily as a validity detector, i.e., given a prompt and a candidate continuation, it predicts whether the model's internal inference trajectory is consistent with a correct choice. A natural next step is to transition TaT from detection to an interpretability tool. Identifying *where* in the token×layer computation a candidate begins to diverge from a valid trajectory, and *which* mechanisms drive this divergence.

One promising direction is to couple TaT with causal and circuit-level analysis. Because each displacement vector can be mapped back to a specific token position and transformer block, we can use the trained TaT classifier as a readout to perform targeted interventions (e.g., activation patching or causal tracing across heads and MLPs) and quantify which components most affect the TaT score. This would provide a concrete pathway from a trajectory-level signature to interpretable model mechanisms, bridging our macroscopic geometry perspective with head and circuit-level explanations.

Finally, while our experiments focus on constrained choice-selection benchmarks (to ensure unambiguous supervision), an important next direction is to extend TaT-style trajectory classification to models' *self-generated* multi-step reasoning chains. In this setting, the goal would be to detect reasoning errors or hallucinations within the model's own intermediate derivations. A systematic study of this regime, including task selection, ground-truth construction, and evaluation protocols for open-ended generation, is an important direction for future work.

8 Limitations

While TaT offers robust generalization, it incurs a higher computational cost than simple linear probes, requiring the extraction and processing of full-trajectory activation traces across all layers. Additionally, although our LSTM classifier detects validity, the specific geometric features it learns remain implicit, lacking the interpretability of individual attention heads or circuits. Finally, our approach still requires training using generalizable training data. A successful variation of kinematic descriptors would eliminate the need for training data.

References

Manar Aljohani, Jun Hou, Sindhura Kommu, and Xuan Wang. 2025. A comprehensive survey on the trustworthiness of large language models in healthcare. *Preprint*, arXiv:2502.15871.

Anonymous. 2025. Enhancing cross-task transfer of large language models via fourier activation steering. In *Submitted to The Fourteenth International Conference on Learning Representations*. Under review.

Waïss Azizian, Michael Kirchhof, Eugene Ndiaye, Louis Béthune, Michal Klein, Pierre Ablin, and marco cuturi. 2025. The geometries of truth are orthogonal across tasks. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.

Yuntai Bao, Xuhong Zhang, Tianyu Du, Xinkui Zhao, Zhengwen Feng, Hao Peng, and Jianwei Yin. 2025. Probing the geometry of truth: Consistency and generalization of truth directions in LLMs across logical transformations and question answering tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 682–700, Vienna, Austria. Association for Computational Linguistics.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, and 5 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. https://transformer-circuits.pub/2023/monosemantic-features. Transformer Circuits Thread.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Nelson Elhage, Robert Lasenby, and Christopher Olah. 2023. Privileged bases in the transformer residual stream. https://transformer-circuits.pub/2023/privileged-basis/index.html. Transformer Circuits Thread.

Jesseba Fernando and Grigori Guitchounts. 2025. Transformer dynamics: A neuroscientific approach to interpretability of large language models. *Preprint*, arXiv:2502.12131.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. 2023. The emergence of clusters in self-attention dynamics. *arXiv preprint arXiv:2305.05465*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. On the biology of a large language model. *Transformer Circuits Thread*.

Yuhang Liu, Dong Gong, Yichao Cai, Erdun Gao, Zhen Zhang, Biwei Huang, Mingming Gong, Anton van den Hengel, and Javen Qinfeng Shi. 2025. I predict therefore i am: Is next token prediction enough to learn human-interpretable concepts from data? *Preprint*, arXiv:2503.08980.

Yiping Lu, Zhuhan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Understanding and improving transformer from a multiparticle dynamic system perspective. *arXiv preprint arXiv:1906.02762*.

Robert Manson. 2025. Curved inference: Concern-sensitive geometry in large language model residual streams. *arXiv preprint arXiv:2507.21107*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 839–849.

Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. 2025. Steering language model refusal with sparse autoencoders. *Preprint*, arXiv:2411.11296.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2025. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *The Thirteenth International Conference on Learning Representations*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Joris Postmus and Steven Abreu. 2024. Steering large language models using conceptors: Improving addition-based activation engineering. In *MINT: Foundation Model Interventions*.

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

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473.

Krithi Shailya, Shreya Rajpal, Gokul S Krishnan, and Balaraman Ravindran. 2025. Lext: Towards evaluating trustworthiness of natural language explanations. *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

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

Yukun Zhang and Qi Dong. 2025. Empirical investigation of latent representational dynamics in large language models: A manifold evolution perspective. *arXiv preprint arXiv:2505.20340*.

Yufa Zhou, Yixiao Wang, Xunjian Yin, Shuyan Zhou, and Anru R. Zhang. 2025. The geometry of reasoning: Flowing logics in representation space. *arXiv preprint arXiv:2510.09782*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

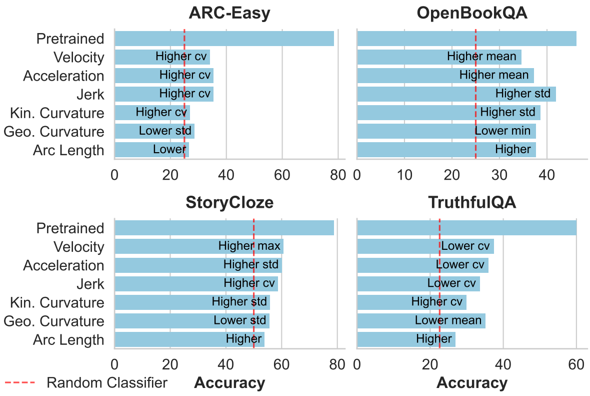

Figure 3: Top performance of Qwen2.5-14b on 4 reasoning benchmarks using kinematic descriptors with varying rule-sets. Red dashed line represents the random classifier accuracy. While the velocity of activations obtains better results than the base model itself, there is no consistency in this performance improvement across datasets despite the oracle-guided approach to evaluation.

A Kinematic Descriptors for Qwen 2.5-14b

We provide the kinematic analysis for Qwen2.5-14B in this section. As noted in Section 4, while simple scalar descriptors like velocity and acceleration provide some signal, they are less consistent for Qwen2.5-14B compared to Llama-3.1-8B, failing to reliably outperform the base model's intrinsic confidence across all benchmarks. This inconsistency highlights the limitation of relying solely on scalar kinematics and reinforces the necessity of the learnable trajectory-based approach (TaT).

B TaT as a Cross-Task Classifier on Qwen 2.5-14b

Table 8 summarizes the cross-task generalization performance for Qwen2.5-14B. These results corroborate findings from the Llama-3.1-8B experiments. TaT consistently achieves higher OOD accuracy than linear probing. This confirms that the trajectory-based invariants captured by our method are not specific to a single architecture but represent a more fundamental property of transformer inference dynamics.

C Implementation Details

For both our method and probing, we sweep all optimizer-based learning rates with validation sets of the corresponding train set (learning rate and regularization parameters). For TaT, we sweep the LSTM hidden dimensions from 128 to 512 and number of layers from 1 to 3. For each train session, we run the session with 3 seeds and average the performance and choose the top session using the validation set. For most experiments, TaT settled on 128 LSTM hidden state size and 2 or 3 layers.

Problem-completion examples Table 9 shows one concrete example per requested task, formatted exactly as in our code paths before tokenization.

Linear probe baseline. For each candidate answer, we select a single vector from the answer trajectory (the final token position of the candidate continuation). In our main reasoning probe runs, the layer we probe is the exact middle layer of the network, which depends on the depth of the specific model.

Few-shot settings used in final baselines. Table 10 lists the exact few-shot counts used in our selected benchmark baselines.

D Which layer should be probed?

Static probing methods additionally require selecting *where* to probe. While mid-to-late layers often contain the most behaviorally salient features, the optimal layer is known to vary across tasks and datasets (Rimsky et al., 2024; Azizian et al., 2025). To quantify this sensitivity in our evaluation suite, we train linear probes on a range of mid-to-late layers and report accuracy per dataset.

Table 11 confirms substantial variation: ARC-C peaks around layer -15, ARC-E around -13, and OpenQA again near -15. No single layer dominates across tasks, highlighting a practical limitation of static probing: performance depends on an unprincipled, dataset-specific choice of probing depth. TaT mitigates this dependency by consuming the entire trajectory, removing the need for a layer-selection heuristic.

Table 8: Generalization on Qwen2.5-14b for TaT vs. Probing

Train Dataset	**Method**	**Evaluation Dataset**				**Avg**	**ID Acc.**	**OOD Avg.**
		StoryCloze	**OpenQA**	**ARC-E**	**ARC-C**			
Zero-shot Accuracy		80.6	61.8	83.5	58.2	71.0	-	-
Few-shot Accuracy		84.4	62.0	88.3	74.0	77.2	-	-
OpenQA	Linear Probe	52.86	80.31	73.02	72.02	69.55	80.31	65.97
	Trajectory	**82.34**	**88.20**	**86.70**	**78.92**	**84.04**	**88.20**	**82.65**
ARC-C	Linear Probe	69.94	75.48	78.81	75.60	74.96	75.60	74.74
	Trajectory	**81.72**	**77.20**	**92.05**	**85.24**	**84.05**	**85.24**	**83.66**
ARC-E	Linear Probe	62.44	77.83	84.32	75.96	75.14	84.32	72.08
	Trajectory	**87.28**	**84.40**	**94.28**	**85.07**	**87.76**	**94.28**	**85.58**

Table 9: Concrete problem/completion examples used for feature extraction and evaluation.

Task	**Problem (prompt/context)**	**Completion(s) / label**
ARC-Challenge	*An astronomer observes that a planet rotates faster after a meteorite impact. Which is the most likely effect of this increase in rotation?*	Correct: *Planetary days will become shorter.*
OpenBookQA	*Fact: using less resources usually causes money to be saved Question: A person wants to start saving money so that they can afford a nice vacation at the end of the year. After looking over their budget and expenses, they decide the best way to save money is to*	Correct: *quit eating lunch out*
SocialIQA	*Tracy didn't go home that evening and resisted Riley's attacks. What does Tracy need to do before this?*	Correct: *Find somewhere to go*
RealToxicity	*Corruption involving the contractors is the chief culprit for the prison's problems, according to a recent report from the Special Inspector General for Afghanistan Reconstruction—a congressionally-mandated watchdog agency.*	Toxicity target $y \in [0, 1]$: 0.295593 (this sample)

Table 10: Few-shot counts used in final baseline evaluations (selected tasks).

Task	**Few-shot count**	**Notes**
ARC-E	5	-
ARC-C	25	-
OpenQA	5	Facts/Context included for each instance.
BoolQ	5	Reading comprehension tasks aren't commonly evaluated in few-shot scenarios, but we use 5 shots only to demonstrate the performance against TaT, which is always zero-shot.
ComQA	5	-
CosQA	5	-
StoryCloze	5	Narrative completion tasks aren't commonly evaluated in few-shot scenarios, but we use 5 shots only to demonstrate the performance against TaT, which is always zero-shot.
Hellaswag	10	-
SiQA	5	-

Table 11: Probe Layer Sensitivity. Accuracy of linear probes trained on different layers (indexed from the last layer, -1). Bold values indicate the top performing layers for each dataset.

Layer Index	**-1**	**-2**	**-3**	**-4**	**-5**	**-6**	**-7**	**-8**	**-9**	**-10**	**-11**	**-12**	**-13**	**-14**	**-15**	**-16**
ARC-C	0.6651	0.6865	0.6733	0.6629	0.6780	0.6765	0.6907	0.6943	0.7024	0.6884	0.6884	0.6965	0.7029	**0.7098**	**0.7188**	**0.7103**
ARC-E	0.6970	**0.7213**	0.6877	0.6962	0.7120	**0.7242**	0.7301	0.6997	**0.7223**	0.7142	0.6975	0.6822	**0.7384**	0.7108	0.7196	**0.7244**
OpenQA	0.6201	0.5598	0.5667	0.6056	0.6471	0.6010	0.6301	0.6291	0.6368	0.6641	0.5780	0.6351	0.6397	0.6408	**0.6924**	**0.6711**