# Predicting Chain-of-Thought Correctness from Trajectory Geometry

**Anonymous authors**
**Paper under double-blind review**

## Abstract

We ask whether the geometry of a reasoning trajectory, that is, how a chain-of-thought (CoT) trace moves through semantic space independent of its length, predicts whether the final answer is correct, and whether that prediction is useful in practice. Across 2,800 CoT traces spanning three reasoning benchmarks (FOLIO, GSM8K, and PrOntoQA) and five language models, we extract interpretable trajectory-level features (adjacent-step transition energy, path entropy, semantic drift, loopiness, discourse-graph spectra, and direction-sensitive drift) and predict per-trace correctness. Under problem-grouped cross-validation, in which every model's traces for a given problem are held out together, the engineered features reach 0.748 balanced accuracy (95% confidence interval [0.733, 0.764]), against 0.609 for a length-only baseline and 0.687 for a mean-pooled sentence-embedding classifier; the margin over both is significant under a paired bootstrap ($p < 0.001$). A purely geometric subset of the features reaches 0.72 balanced accuracy on its own, well above the lexical-entropy and discourse-marker feature groups (both below 0.64), which isolates a specifically geometric signal rather than a generic interpretable-feature effect. The result survives label permutation, length-feature ablation, final-answer removal, and prefix truncation, and generalizes leave-one-dataset-out with a measurable discount. Used as a verifier, the classifier reranks the four to five model traces for each problem to a correct answer 88.3% of the time in distribution, against 79.8% for majority voting; out of distribution, however, the ranking inverts, and majority voting wins on two of three held-out benchmarks. The verifier is thus useful when trained on the target distribution but does not transfer across benchmarks. We report one further qualifying result, namely that on the synthetic PrOntoQA benchmark a generic embedding baseline matches the engineered features.

## 1 Introduction

Chain-of-thought (CoT) prompting (Wei et al., 2022) has made the intermediate reasoning trace a standard artifact of how large language models solve hard problems. A natural question follows: given a trace alone, can we tell whether its final answer is correct? Most existing answers either read the model's internal state or pay for extra samples. We ask a narrower and cheaper question. Does the *geometry* of the trace, the path it follows through semantic space, carry a correctness signal, and is that signal strong enough to act on?

The motivating intuition is simple. A trace that wanders, loops back on itself, or drifts erratically through semantic space may be likelier to be wrong than one that advances steadily toward a conclusion. We operationalize this intuition with a set of interpretable trajectory features computed from sentence embeddings of the trace, and we test, on 2,800 traces from three reasoning benchmarks and five models, whether those features predict per-trace correctness. We then test whether the resulting classifier is useful, by applying it as a post-hoc verifier for selective prediction, for reranking candidate traces, and for early stopping.

Two confounds dominate the design. The first is **length**: longer traces may simply be more (or less) error-prone, so any length-correlated feature inherits that signal trivially. We ablate all length features. The second is **problem leakage**: each problem in our corpus is solved by several models, so a classifier evaluated under ordinary cross-validation can memorize problem identity rather than learn a trace-level signal (Kaufman

Table 1: Corpus summary. Each problem is solved by several models, which motivates problem-grouped cross-validation.

| **Property** | **Value** |
|---|---|
| Traces | 2,800 (1,624 correct / 1,176 incorrect) |
| Problems | 600, each solved by 4–5 models |
| Benchmarks | FOLIO, GSM8K, PrOntoQA |
| Models | Claude Sonnet, Gemini 2.5 Pro, GPT-OSS-120B, Llama-3.1-8B, Mistral-7B |
| Decoding | temperature 0 (one trace per model and problem) |

et al., 2012). We use problem-grouped cross-validation throughout, holding out every model's traces for a problem together.

Our contributions are as follows. (i) Under leakage-controlled evaluation, interpretable trajectory-geometry features predict CoT correctness significantly above length, lexical, discourse-graph, and generic sentence-embedding baselines. (ii) A purely geometric subset of the features beats the lexical and discourse-marker groups on its own, which isolates a specifically geometric signal rather than a generic interpretable-feature effect. (iii) The result survives label permutation, length-feature ablation, final-answer removal, and prefix truncation, and it generalizes leave-one-dataset-out with a discount. (iv) Used as a verifier the classifier reranks candidate traces to a correct answer 88.3% of the time in distribution, 8.5 points above majority voting, but out of distribution majority voting wins on two of three held-out benchmarks, so the verifier helps only when trained on the target distribution. (v) We report one further qualifying result, namely that on the synthetic PrOntoQA benchmark a generic embedding baseline matches the engineered features.

## 2 Related work

Predicting CoT correctness sits among several lines of work. *Verifier and process reward models* train a separate network to score reasoning steps or final answers (Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2024). *Self-consistency* samples multiple traces and selects the majority answer (Wang et al., 2023). *Uncertainty and confidence estimation* reads calibrated probabilities or verbalized confidence from the generating model itself (Kadavath et al., 2022; Tian et al., 2023). *Activation probing* predicts correctness or truthfulness from a model's hidden states (Azaria & Mitchell, 2023; Burns et al., 2023). A related line studies whether stated reasoning is faithful to the computation that produced the answer (Turpin et al., 2023; Lanham et al., 2023).

Our setting is deliberately weaker and cheaper than most of these. We use a post-hoc, text-only, model-agnostic classifier over interpretable trajectory features, with no access to logits, hidden states, or extra samples. This paper measures how much correctness signal the bare geometry of a trace contains, under controls that rule out the obvious confounds. We position the engineered features against the natural cheap alternatives, namely length, lexical, and discourse-graph features and a generic sentence-embedding classifier, and we report where an ensemble answer-agreement signal beats all of them.

## 3 Data

Table 1 summarizes the corpus. We use 2,800 CoT traces covering 600 problems drawn from three reasoning benchmarks: FOLIO (Han et al., 2022), a first-order-logic natural-language inference set; GSM8K (Cobbe et al., 2021), grade-school math word problems; and PrOntoQA (Saparov & He, 2023), synthetically generated deduction problems. Each problem is solved by four to five language models (Claude Sonnet, Gemini 2.5 Pro, GPT-OSS-120B, Llama-3.1-8B, and Mistral-7B) at temperature 0, giving one trace per model and problem.

Each trace is normalized to a common schema, segmented into reasoning sentences, and embedded sentence by sentence with the `all-MiniLM-L6-v2` sentence-transformer model (Reimers & Gurevych, 2019; Wang

et al., 2020), which produces a 384-dimensional embedding per sentence. Because each problem recurs across models, every cross-validation split in this paper is grouped by the key `dataset::problem_idx`, so a problem never appears in both the training and the test fold of a split.

# 4 Method

## 4.1 Trace representation

A trace is segmented into reasoning sentences and embedded, giving a sequence $\mathbf{e}_1, \ldots, \mathbf{e}_n \in R^{384}$. We treat this sequence as a discrete trajectory in semantic space and compute features of its shape. The basic local quantity is the *transition energy* of step $t$, the cosine distance between consecutive sentence embeddings,

$$d_t = 1 - \cos(\mathbf{e}_t, \mathbf{e}_{t+1}), \qquad t = 1, \ldots, n-1. \tag{1}$$

## 4.2 Feature groups

We compute 52 features in total, organized into named groups.

- **`length_only`** (6 features): character, word, and sentence counts, mean sentence length, newline count, and an explicit reasoning-step count.
- **`lexical_entropy`** (5 features): type-token ratio, repetition rate, unigram and bigram entropy, and the entropy of the sentence-length distribution.
- **`marker_features`** (17 features): counts of discourse connectives (for example *because*, *therefore*, *wait*, *verify*) and forward and reverse marker-progression indicators.
- **`graph_features`** (8 features): the trace is turned into a discourse graph whose nodes are sentences and whose edges join sentence pairs with cosine similarity above 0.5; we record density, average degree, clustering, connected components, the Laplacian spectral entropy, and the algebraic connectivity.
- **`trajectory_features`** (15 features): transition-energy statistics (mean, standard deviation, maximum, minimum), total path energy, direct start-to-end distance, path efficiency, loopiness, semantic-drift slope, path entropy, conclusion compression, two direction-sensitive features and a transition-predictability feature, all described in Section 4.3.

The combined set is **`all_features`**. We also evaluate the ablations `all_minus_length` (every feature except the length group) and `all_minus_trajectory`, and the partial sets `trajectory_plus_graph` and `trajectory_plus_entropy`. Appendix A gives the full definition of every trajectory feature.

## 4.3 Direction-sensitive features

An earlier version of this work included forward-versus-reverse "energy gap" features. Under symmetric cosine distance these are degenerate. A reversed sentence sequence has the same multiset of adjacent distances, so reverse path energy equals forward path energy and the gap is identically zero. Projection of steps onto the start-to-end axis is likewise invariant under reversal, because reversing the sequence negates both the steps and the axis. We removed these degenerate features and added two that genuinely depend on order. Both are anchored on the *opening* sentence, which is not a symmetric reference point:

- **`drift_monotonicity`**: the fraction of consecutive steps that move farther from the opening sentence, distinguishing steady outward drift from oscillation.
- **`energy_front_back_ratio`**: the share of total step displacement that occurs in the first half of the trace, distinguishing front-loaded from back-loaded reasoning effort.

Table 2: Correctness prediction, overall, under problem-grouped cross-validation. Balanced accuracy and ROC-AUC are reported for the best of three classifiers per feature group.

| **Feature group** | **Balanced accuracy** | **ROC-AUC** |
|---|---|---|
| majority-class baseline | 0.500 | 0.500 |
| `length_only` | 0.609 | 0.655 |
| `lexical_entropy` | 0.625 | 0.680 |
| `graph_features` | 0.629 | 0.680 |
| `embedding_meanpool` (baseline) | 0.687 | 0.751 |
| `trajectory_features` | 0.713 | 0.805 |
| `all_minus_length` | 0.744 | 0.844 |
| **`all_features`** | **0.748** | **0.845** |

We additionally tested one feature, `path_irreversibility`, that measures whether a trace's step-to-step dynamics are more predictable forward than backward, by fitting affine transition maps in a per-trace principal-component space in both directions and taking the log-ratio of their residual variances. It carries no measurable signal: per-trace estimates from traces of roughly ten steps are too noisy to resolve any asymmetry (Sections 5.2 and 7). We retain it inside `all_features` for continuity with the released artifacts, and Section 5.2 confirms that the headline result does not depend on it.

### 4.4 Classifiers, evaluation, and baselines

We train three classifiers, namely logistic regression, a random forest (Breiman, 2001), and gradient-boosted trees (Chen & Guestrin, 2016), using the scikit-learn implementations (Pedregosa et al., 2011). All evaluation uses 5-fold stratified, problem-grouped cross-validation with median imputation of missing values and class-balanced weighting. We report balanced accuracy and the area under the ROC curve (ROC-AUC). Confidence intervals and paired significance tests use a percentile bootstrap over out-of-fold predictions (Efron & Tibshirani, 1994) with 1,000 resamples; for the paired comparisons we fix a random forest, so that the comparison reflects the feature set rather than an interaction between the feature set and the classifier family.

Baselines come in two kinds. The cheap text baselines are the `length_only`, `lexical_entropy`, and `graph_features` groups above. The key comparison is **`embedding_meanpool`**: a classifier trained on the 384-dimensional mean-pooled MiniLM sentence embedding of the whole trace. If a generic embedding matches the engineered features, then the feature engineering adds little. We also report **`cross_model_answer_agreement`**, the fraction of the other models on the same problem that produced the same final answer, which is an ensemble-consistency signal rather than a single-trace one.

## 5 Results

### 5.1 Correctness prediction

Table 2 reports overall performance under problem-grouped cross-validation. The engineered features reach 0.748 balanced accuracy, well above the 0.609 of the length-only baseline and the 0.687 of the generic embedding baseline. The `trajectory_features` group alone reaches 0.713, and removing the length features (`all_minus_length`, 0.744) barely changes the headline. For reference, ordinary (non-grouped) cross-validation gives `all_features` 0.744, so the result is not an artifact of problem-identity leakage.

Table 3 breaks the result down by benchmark. The engineered features beat the embedding baseline clearly on FOLIO and GSM8K. On PrOntoQA the embedding baseline is the strongest single method, a point we return to in Section 8.

Table 3: Balanced accuracy per benchmark, best classifier per cell.

| **Benchmark** | `length` | `trajectory` | `all_minus_length` | `all_features` | `embedding` |
|---|---|---|---|---|---|
| FOLIO | 0.558 | 0.636 | 0.683 | 0.684 | 0.619 |
| GSM8K | 0.601 | 0.718 | 0.729 | 0.735 | 0.634 |
| PrOntoQA | 0.703 | 0.836 | 0.860 | 0.855 | 0.871 |

Table 4: Isolating the geometric signal: balanced accuracy of a purely geometric feature subset against the lexical and discourse-marker groups, under problem-grouped cross-validation, best of three classifiers. Computed by the supplementary script `scripts/10_review_experiments.py` on the same environment as the main pipeline.

| **Feature group** | $n$ | **Balanced accuracy** | **ROC-AUC** |
|---|---|---|---|
| `lexical_entropy` | 5 | 0.625 | 0.680 |
| `marker_features` | 17 | 0.634 | 0.690 |
| `geometry_only` | 14 | 0.714 | 0.807 |
| `geometry_only` without `path_irreversibility` | 13 | 0.722 | 0.809 |

### 5.2 Isolating the geometric signal

A predictor built from interpretable trace features could succeed because trajectory *geometry* carries the signal, or simply because interpretable features of any kind do. To separate the two we evaluate a purely geometric subset, `geometry_only`: the trajectory features that depend only on the embedding path (transition energy, total path energy, path efficiency, loopiness, semantic drift, conclusion compression, path entropy, and the direction-sensitive features), excluding the single discourse-marker feature that the trajectory group contains. Table 4 compares it with the lexical and discourse-marker groups.

The geometric subset reaches 0.714 balanced accuracy on its own, and 0.722 once the inert `path_irreversibility` feature is removed. Both are roughly nine points above the lexical-entropy and discourse-marker groups, which sit below 0.64, and they recover most of the 0.748 of the full feature set (Table 2). The correctness signal is therefore specifically geometric: it is not reproduced by interpretable lexical or discourse features of comparable count. The comparison also isolates `path_irreversibility` as inert: removing it slightly improves the geometric subset, and `all_features` scores 0.748 with it and without it.

### 5.3 Statistical significance

Table 5 reports paired bootstrap comparisons on out-of-fold predictions with a fixed random forest. The engineered features beat both the length baseline and the generic embedding baseline by margins whose 95% confidence intervals exclude zero, with $p < 0.001$ in every case.

### 5.4 Robustness

We run four controls; Appendix B gives the full table.

**Shuffled labels.** Permuting the correctness labels collapses performance to chance (`trajectory_features` $0.500 \pm 0.007$, `all_features` $0.500 \pm 0.009$ over repeats), which rules out leakage or artifactual separability.

**Length control.** `all_minus_length` (0.744) is statistically indistinguishable from `all_features` (0.748), and both are far above `length_only` (0.609). The signal is not verbosity.

**Final-answer removal.** Stripping the final-answer span from each trace and recomputing all features, `all_features` holds at 0.690. Even with the final answer removed, the classifier matches the generic embedding baseline computed on the full trace (0.687), so it is not merely reading answer formatting.

Table 5: Paired bootstrap on out-of-fold predictions (random forest, 1,000 resamples).

| **Comparison** | **Δ balanced acc.** | **95% CI** | $p$ |
|---|---|---|---|
| `trajectory_features` − `length_only` | +0.107 | $[0.086, 0.130]$ | $< 0.001$ |
| `all_features` − `length_only` | +0.142 | $[0.122, 0.164]$ | $< 0.001$ |
| `trajectory_features` − `embedding_meanpool` | +0.053 | $[0.034, 0.073]$ | $< 0.001$ |
| `all_features` − `embedding_meanpool` | +0.088 | $[0.069, 0.107]$ | $< 0.001$ |

Table 6: Leave-one-dataset-out balanced accuracy, best classifier per cell.

| **Held-out benchmark** | `length_only` | `trajectory` | `all_minus_length` | `all_features` |
|---|---|---|---|---|
| FOLIO | 0.580 | 0.642 | 0.663 | 0.669 |
| GSM8K | 0.507 | 0.583 | 0.576 | 0.547 |
| PrOntoQA | 0.552 | 0.663 | 0.684 | 0.686 |

**Prefix only.** Computed from the first 50% or 75% of each trace, `all_features` scores 0.636 and 0.656 respectively, against 0.741 for the full trace on the same length-filtered subset. Correctness is partially predictable before the trace ends.

### 5.5 Cross-dataset generalization

Table 6 reports leave-one-dataset-out generalization: train on two benchmarks, test on the third. The engineered features beat the length baseline on every held-out split, so the signal is not purely a per-dataset artifact. However, held-out performance (0.55 to 0.69) is well below the in-distribution result (0.70 to 0.86 in Table 3). A substantial part of the signal is benchmark-specific, and GSM8K generalizes worst. On held-out GSM8K, adding the lexical, marker, and discourse-graph feature groups slightly hurts generalization (`all_features` 0.547 against 0.583 for `trajectory_features` alone), which suggests that some of the auxiliary features are benchmark-specific.

## 6 The verifier use-case

A correctness predictor is only interesting if it can be acted on. We use the classifier (random forest, `all_features`) as a post-hoc verifier for selective prediction, for reranking candidate traces, and for early stopping, and we then test whether the reranking result transfers to an unseen benchmark.

### 6.1 Selective prediction

Ranking traces by classifier confidence and abstaining on the least-confident ones raises accuracy on the accepted set from 0.764 at full coverage to 0.818 at 80% coverage and 0.895 at 50% coverage. The area under the risk-coverage curve is 0.107. Appendix C gives the full curve.

### 6.2 Cross-model reranking, in distribution

For each of the 600 problems we select, among its four to five model traces, the one the verifier scores highest, and we record whether the selected trace has the correct answer. Predictions here are out-of-fold under problem-grouped cross-validation: the verifier never scored a trace whose problem it trained on, but it did train on other problems from the same three benchmarks, so this is an in-distribution measurement. Table 7 compares the policy with random selection, with majority voting over the models' final answers, and with an oracle that succeeds whenever any model produced a correct trace.

Table 7: Cross-model reranking, in distribution: probability of selecting a correct answer among the 4–5 model traces per problem ($n = 600$), with verifier scores out-of-fold under problem-grouped cross-validation.

| **Selection policy** | **Accuracy** |
|---|---|
| random model | 0.582 |
| majority vote (most common final answer) | 0.798 |
| **verifier rerank** | **0.883** |
| oracle (any correct trace) | 0.953 |

Table 8: Out-of-distribution reranking: the verifier is trained on two benchmarks and reranks the held-out third (200 problems each). The final column repeats the in-distribution reranking accuracy for the same benchmark. Computed by `scripts/10_review_experiments.py`; see the Reproducibility Statement.

| **Held-out benchmark** | **OOD verifier** | **majority** | **random** | **oracle** | **in-dist. verifier** |
|---|---|---|---|---|---|
| FOLIO | 0.765 | 0.740 | 0.624 | 0.950 | 0.840 |
| GSM8K | 0.710 | 0.820 | 0.604 | 0.925 | 0.845 |
| PrOntoQA | 0.580 | 0.835 | 0.517 | 0.985 | 0.965 |

In distribution the verifier reranker selects a correct answer 88.3% of the time, 8.5 points above majority voting and most of the way to the oracle ceiling of 95.3%. Section 6.3 tests whether this advantage survives a shift to an unseen benchmark.

### 6.3 Out-of-distribution reranking

The reranking result above is in-distribution. The deployment scenario that the Broader Impact statement cares about, applying the verifier to a benchmark it was not trained on, is a different regime, and we test it directly: we train the verifier on two benchmarks and rerank the held-out third. Table 8 reports the result.

Out of distribution the verifier beats majority voting on only one of the three held-out benchmarks, FOLIO, and there by 2.5 points. On GSM8K it trails majority voting by 11 points, and on PrOntoQA it falls to 0.580, close to random selection (0.517) and 26 points behind majority voting. Averaged over the held-out benchmarks majority voting leads the verifier by roughly 11 points, the reverse of the in-distribution ordering. Leaner feature sets do not rescue transfer: reranking with `trajectory_features` or with the geometric subset alone fails to beat majority voting on any of the three held-out benchmarks. The reranking advantage is therefore in-distribution. The verifier improves answer selection when it is trained on the target distribution, and does not transfer across benchmarks.

### 6.4 Early stopping

We also test committing to a prediction from a trace prefix once a prefix classifier is sufficiently confident. In distribution, at a moderate confidence threshold, the policy reaches 0.756 accuracy against 0.762 for the full trace, while reading 84.5% of each trace on average and stopping early on 35.6% of traces. The compute saving is real but modest.

## 7 Interpretability and a causal probe

**Permutation importance.** Computed on held-out folds, permutation importance ranks `conclusion_compression` first by a wide margin (a 0.060 drop in balanced accuracy when permuted), followed by `however_count`, `symbolic_token_count`, and `direct_start_end_distance`. The top of the ranking is a mix of trajectory, marker, and lexical features, so no single group carries the result alone. The

transition-predictability feature ranked 18th, with importance indistinguishable from noise, consistent with Section 4.3.

**Intervention test.** To check that the classifier responds to trace *structure* rather than to an order-invariant summary, we perturb each trace and recompute its features. Shuffling sentence order shifts the predicted correctness probability by 0.29 on average and flips 46% of predictions; injecting duplicated sentences shifts it by 0.26 and flips 34%. The classifier is strongly sensitive to reasoning structure. This is a sensitivity probe rather than a formal causal claim, but it is evidence that the trajectory signal is real and not a repackaging of bag-of-features content.

## 8 Negative results

We report two findings that qualify the contribution.

**The embedding baseline wins on PrOntoQA.** On the most regular benchmark, the synthetically generated PrOntoQA, `embedding_meanpool` reaches 0.871 and exceeds `all_features` at 0.855 (Table 3). The advantage of the engineered features is concentrated on the more naturalistic FOLIO and GSM8K, where raw embeddings do markedly worse. The engineering earns its keep on naturalistic reasoning, not on synthetic deduction.

**Ensemble answer-agreement is a strong signal.** As a classifier feature, the fraction of other models on a problem that agree on an answer reaches roughly 0.84 balanced accuracy on its own, above any trajectory feature. It is not a competitor in the single-trace setting that this paper targets, since it requires an ensemble, and in the reranking task of Section 6 the verifier beats majority voting. Still, any deployment that already has multiple model responses should consider answer-agreement, and a paper on this topic must say so.

## 9 Limitations

**The findings are correlational.** The intervention test of Section 7 shows that the classifier is sensitive to trace structure, but it does not establish a causal mechanism linking trajectory geometry to errors.

**Scale.** The study covers three benchmarks, five models, and temperature-0 decoding only. Generalization to other task families, to larger model sets, and to sampled (temperature greater than zero) decoding is untested.

**Single embedding backbone.** All features derive from one sentence-embedding model and one sentence splitter. We have not ablated either choice.

## 10 Conclusion

Under leakage-controlled evaluation, interpretable trajectory-geometry features predict chain-of-thought correctness significantly above length, lexical, discourse-graph, and generic sentence-embedding baselines, and a purely geometric subset of the features carries that signal on its own, well above the lexical and discourse-marker groups. The result survives label permutation, length ablation, final-answer removal, and prefix truncation. Used as a verifier the classifier is genuinely useful in distribution: it reranks candidate traces to a correct answer 88.3% of the time, 8.5 points above majority voting, and it supports selective prediction. That usefulness does not transfer: trained on two benchmarks and applied to a third, the verifier is beaten by plain majority voting on two of three held-out benchmarks. Cheap, interpretable trajectory features predict reasoning correctness from trace geometry alone, and rerank usefully when a verifier can be trained on the target distribution; deploying one across a distribution shift, however, calls for retraining rather than transfer.

### Broader Impact Statement

The method is a lightweight, post-hoc filter on reasoning traces. Used well, it can flag low-confidence answers for review and reduce the cost of ensemble inference. The main risk is overtrust: a verifier that is right roughly

88% of the time in the in-distribution reranking setting is still wrong often, and Section 6.3 shows that out of distribution it is beaten by plain majority voting. It should not be used as a sole gate on high-stakes decisions, and it should not be applied to a distribution it was not trained on without first retraining and re-measuring its accuracy on that distribution. The work uses only model-generated reasoning traces on public reasoning benchmarks and involves no human-subjects data.

### Reproducibility Statement

All numbers in this paper are produced by a single pipeline. The trace corpus, the feature-extraction code,[1] the evaluation scripts, and the verifier and interpretability experiments will be released. Every reported metric corresponds to a named artifact (classifier metrics, ablation tables, robustness reports, cross-dataset metrics, verifier metrics, and interpretability reports) emitted by that pipeline. The geometric-subset results of Section 5.2 and the out-of-distribution reranking results of Section 6.3 are produced by an additional released script, `scripts/10_review_experiments.py`, run on the same host and environment as the main pipeline; it reproduces the committed pipeline figures (for example overall in-distribution reranking 0.883 and `all_features` 0.748), so Tables 4 and 8 are consistent with the rest of the paper.

## References

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, et al. FOLIO: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*, 2022.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4):1–21, 2012.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[1] In the released code, the `trajectory_features` group is named `thermo_features` for historical reasons.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *International Conference on Learning Representations*, 2023.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, 2023.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, 2020.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

## A Trajectory feature definitions

Let $\mathbf{e}_1, \ldots, \mathbf{e}_n$ be the sentence embeddings of a trace and let $d_t = 1 - \cos(\mathbf{e}_t, \mathbf{e}_{t+1})$ be the transition energy of step $t$. Table 9 defines the 15 features of the `trajectory_features` group.

## B Full robustness table

Table 10 reports the trace-variant controls of Section 5.4. All rows use problem-grouped cross-validation on the length-filtered subset of 2,714 traces for which every variant is defined, so the `full_trace` row (0.741) is the matched reference for the prefix rows rather than the headline 0.748 of Table 2.

## C Risk-coverage curve

Table 11 reports the selective-prediction curve of Section 6: accuracy on the accepted set as a function of coverage, when traces are accepted in decreasing order of classifier confidence.

Table 9: Definitions of the `trajectory_features` group.

| Feature | Definition |
|---|---|
| `transition_energy_mean/std/max/min` | Mean, standard deviation, maximum, and minimum of the step energies $\{d_t\}$. |
| `total_path_energy` | $\sum_t d_t$, the total cosine distance traversed. |
| `direct_start_end_distance` | $1 - \cos(\mathbf{e}_1, \mathbf{e}_n)$, the straight-line semantic distance from first to last sentence. |
| `path_efficiency` | `direct_start_end_distance` divided by `total_path_energy`; near 1 for a direct path, near 0 for a meandering one. |
| `loopiness` | Mean over sentences $t \geq 3$ of the maximum cosine similarity to any non-adjacent earlier sentence; high when the trace returns near earlier states. |
| `semantic_drift_slope` | Slope of a linear fit of $1 - \cos(\mathbf{e}_1, \mathbf{e}_t)$ against $t$; the rate of outward drift from the opening sentence. |
| `conclusion_compression` | Cosine similarity between the final sentence embedding and the centroid of all preceding sentence embeddings. |
| `path_entropy` | Shannon entropy (base 2) of the normalized step-energy distribution $p_t = d_t / \sum_{t'} d_{t'}$. |
| `drift_monotonicity` | Fraction of consecutive steps for which the distance from the opening sentence increases. |
| `energy_front_back_ratio` | Share of the total step displacement (sum of $\|\mathbf{e}_{t+1} - \mathbf{e}_t\|$) that occurs in the first half of the trace. |
| `path_irreversibility` | Log-ratio of reverse to forward residual variance of an affine transition map fit in a per-trace principal-component space (Section 4.3). Carries no measurable signal; see Section 5.2. |
| `marker_directionality_gap` | Forward minus reverse marker progression: 1 if conclusion markers (*therefore*, *hence*, *finally*, *so*) tend to appear later than premise markers (*because*, *since*, *first*, *second*), and the reverse for the reversed trace. |

Table 10: Balanced accuracy under trace-variant controls, on the length-filtered subset ($n = 2{,}714$) on which every variant is defined. Numbers therefore differ from Table 2.

| Variant | `trajectory_features` | `all_features` |
|---|---|---|
| full trace | 0.694 | 0.741 |
| final answer removed | 0.611 | 0.690 |
| first 50% of trace | 0.576 | 0.636 |
| first 75% of trace | 0.607 | 0.656 |

Table 11: Selective accuracy at selected coverage levels (`all_features`, random forest).

| Coverage | 0.10 | 0.25 | 0.50 | 0.65 | 0.80 | 0.90 | 1.00 |
|---|---|---|---|---|---|---|---|
| Selective accuracy | 0.986 | 0.961 | 0.895 | 0.851 | 0.818 | 0.790 | 0.764 |