Measuring Representation Robustness in Large Language Models for Geometry

Vedant Jawandhia [1] **Dhruv Kumar** [1] **Yash Sinha** [1] **Ankan Pal** [2]

Abstract

Large language models (LLMs) are increasingly evaluated on mathematical reasoning, yet their robustness to equivalent problem representations remains poorly understood. In geometry, identical problems can be expressed in Euclidean, coordinate, or vector forms, but existing benchmarks report accuracy on fixed formats, implicitly assuming representation invariance and masking failures caused by representational changes alone. We propose **GeoRepEval**, a representation-aware evaluation framework that measures correctness, invariance, and consistency at the problem level across parallel formulations, combining strict answer matching, bootstrap confidence intervals, paired McNemar tests, representation-flip analyses, and regression controls for surface complexity. We prove that our Invariance@3 metric decomposes accuracy into robust and fragile components and is bounded by the weakest representation. Evaluating eleven LLMs on 158 curated high-school geometry problems (474 instances), we find accuracy gaps of up to 14 percentage points induced solely by representation choice. Vector formulations emerge as a consistent failure point, with Invariance@3 as low as 0.044 even after controlling for length and symbolic complexity. A convert-then-solve prompting intervention improves vector accuracy by up to 52 percentage points for high-capacity models, suggesting that failures reflect representation sensitivity rather than inability; however, low-capacity models show no gains, indicating deeper limitations. These results suggest that current models rely on representation-specific heuristics rather than abstract geometric reasoning. All datasets, prompts, and scripts are released at https://anonymous.4open.science/r/colm_submission-1C1E.

[1]Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani, Rajasthan, India [2]Department of Mathematics, Birla Institute of Technology and Science, Pilani, Rajasthan, India. Correspondence to: Vedant Jawandhia <vedant@pilani.bits-pilani.ac.in>, Dhruv Kumar <dhruv@pilani.bits-pilani.ac.in>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

Large language models (LLMs) have demonstrated strong capabilities across reasoning-intensive tasks (Brown et al., 2020; OpenAI., 2023), including mathematical problem solving (Cobbe et al., 2021; Hendrycks et al., 2021), symbolic manipulation (Lewkowycz et al., 2022), and logical inference (Wei et al., 2022). Recent advances in scale, instruction tuning, and chain-of-thought prompting have led to substantial gains on benchmarks spanning arithmetic, algebra, and geometry (Brown et al., 2020; Wei et al., 2022; Chowdhery et al., 2022; OpenAI., 2023). However, growing evidence—sensitivity to prompt phrasing (Zhu et al., 2023), adversarial perturbations (Shlens and Szegedy, 2015), and surface-level rewording (Guestrin and Singh, 2020)—suggests LLM performance can be brittle under representational changes (Srivastava et al., 2023). Geometry provides a uniquely structured testbed: the same problem can be expressed via Euclidean, coordinate, or vector representations without altering its semantic content.

Research questions. This paper investigates whether modern LLMs exhibit representation-invariant geometric reasoning by addressing three questions:

RQ1: Does performance vary systematically with representation choice, and can gaps be explained by surface features?

RQ2: Can explicit conversion-based prompting improve cross-representation consistency?

We evaluate whether models produce correct and consistent answers when the same geometry problem is expressed in Euclidean, coordinate, and vector forms—representations that differ in symbolic density and linguistic structure but encode identical mathematical constraints (Lewkowycz et al., 2022).

Prior work focuses on aggregate accuracy over fixed-format benchmarks (Cobbe et al., 2021; Hendrycks et al., 2021; Drori et al., 2022), treating representation as incidental. Studies exploring paraphrasing robustness operate within single representational regimes (Zhu et al., 2023), and ge-

ometry benchmarks rarely provide parallel formulations of the same problem (Trinh et al., 2024). Current evaluations thus cannot determine whether LLMs internally normalize problems or rely on representation-specific heuristics.

We propose **GeoRepEval**, a representation-aware evaluation framework that constructs parallel Euclidean, coordinate, and vector versions of each problem and evaluates at the *problem level*—comparing answers across all three variants. GeoRepEval introduces Invariance@3 (fraction of correct problems in all three representations) and Consistency@3 (fraction of problems with identical output), alongside McNemar tests, representation-flip analysis, and regression controls for surface features (token length, symbolic density).

We evaluate eleven LLMs spanning open-weight (7B–12B) and proprietary architectures on 474 curated geometry problems (158 instances per geometric representation). Euclidean–Vector accuracy gaps reach up to 14 pp (Claude-Haiku: 0.60 Euclid vs. 0.46 Vector), and Invariance@3—the fraction of problems solved correctly across all three representations—remains below 0.47 for all models except GPT-OSS-20B (0.665). At the low end, LLaMA-3.1-8B achieves only 0.044 Invariance@3, meaning fewer than 5% of problems are solved correctly in all three forms. These gaps persist after regression controls for surface complexity ($p < 0.05$ for 8/11 models), and item-level analysis reveals vector formulations as the dominant single point of failure: the CCW pattern (correct on Euclidean and Coordinate, wrong on Vector) is 2–5$\times$ more frequent than WCC (Euclidean-only failure), indicating that models apply Euclidean reasoning "templates" fluently but produce brittle, error-prone vector procedures.

Contributions.

- **GeoRepEval framework:** A statistically grounded, representation-aware evaluation framework with formally justified metrics (Properties 1–3) linking Invariance@3 and Consistency@3 to standard accuracy.
- **Parallel benchmark:** 158 core geometry problems each in Euclidean, coordinate, and vector form (474 instances per source) from standard textbooks, with verified semantic equivalence.
- **Empirical & explanatory analysis:** Accuracy gaps up to 14 pp and Invariance@3 as low as 0.044; a qualitative analysis (§5.7) proposing procedural chain length, possible training distribution effects, and algebraic error compounding as plausible explanations for vector fragility.
- **Prompting intervention:** A convert-then-solve strategy that dramatically improves vector accuracy for mid/high-capacity models (up to 52 pp), suggesting failures are representation-specific; low-capacity models show no benefit, indicating deeper limitations.

2. Related Work

2.1. Mathematical Reasoning and Geometry with Neural Models

Prior studies have evaluated LLM mathematical reasoning on benchmarks such as GSM8K and MATH (Cobbe et al., 2021; Hendrycks et al., 2021; Drori et al., 2022; Lewkowycz et al., 2022; Polu et al., 2023), with complementary surveys assessing model capabilities more broadly (Qin et al., 2023; Huang et al., 2023). Geometry-focused work has explored diagram understanding, symbolic reasoning, and coordinate-based formulations (Seo et al., 2015; Trinh et al., 2024). However, these evaluations typically assess each problem in a single canonical form, leaving it unclear whether success reflects abstract reasoning or sensitivity to representational encoding. GeoRepEval departs from this paradigm by evaluating the *same* problem across Euclidean, coordinate, and vector formulations.

2.2. Evaluation Frameworks for LLM Robustness

Benchmarks such as BIG-bench (Srivastava et al., 2023), HELM (Liang et al., 2022), and Beyond-the-IID (Chang et al., 2024) stress-test models across diverse conditions. Work on representation shift (Zhang et al., 2024), paraphrasing (Guestrin and Singh, 2020; Araki and Neubig, 2021), and adversarial perturbations (Min et al., 2023) has revealed sensitivity to superficial changes despite semantic equivalence. In mathematics, however, most benchmarks remain instance-level and lack aligned variants of the same problem, limiting invariance assessment. GeoRepEval fills this gap with problem-level, representation-aligned evaluation and paired statistical testing.

2.3. Prompting and Reasoning Strategies

Chain-of-thought prompting (Wei et al., 2022), self-consistency (Wang et al., 2023), tree-of-thought (Yao et al., 2023), and tool-augmented methods (Schick et al., 2024) have improved reasoning performance, with zero-shot and least-to-most strategies (Kojima et al., 2022; Zhou et al., 2023) unlocking latent ability without exemplars. However, these studies implicitly assume a fixed representation and focus on *how* models reason rather than *what form* reasoning is conditioned on. Our work isolates representation as a controlled variable, testing whether LLM reasoning is representation-invariant.

3. Methodology

GeoRepEval probes whether LLMs exhibit invariant reasoning across equivalent geometric representations via four

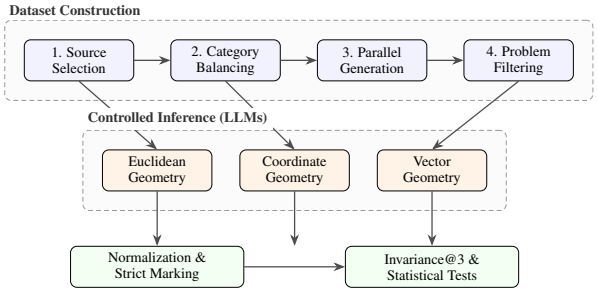

Figure 1. **GeoRepEval pipeline overview.** The framework natively constructs and tracks mathematically equivalent variants (Euclidean, Coordinate, and Vector) through parallel LLM inference to isolate true reasoning capacity from representation sensitivity.

stages: dataset construction, controlled inference, structured scoring, and statistical analysis (Figure 1). Let $\mathcal{M}$ denote a language model. Each problem p_i has gold answer a_i^* and is expressed in representations $r \in \{\text{EUC}, \text{COORD}, \text{VEC}\}$, yielding prompts x_i^r. The predicted answer $\hat{a}_i^r = \text{parse}(\mathcal{M}(x_i^r))$ is *correct* if it matches a_i^* after normalization (§3.2). We capture this using a binary correctness indicator $c_i^r \in \{0, 1\}$ that flags whether the model successfully solved problem p_i in its representation form r, defined formally as $c_i^r = \mathbf{1}[\text{normalize}(\hat{a}_i^r) = \text{normalize}(a_i^*)]$.

3.1. Dataset Construction

The construction pipeline proceeds in four stages:

Stage 1: Source Selection. Problems are drawn from three widely used Indian high-school and junior-college mathematics textbooks—NCERT (Classes IX–XII) (Research and , NCERT), RD Sharma (R. D. Sharma., 2019; 2020), and RS Aggarwal (R. S. Aggarwal., 2019; 2020)—chosen for their emphasis on formal symbolic reasoning and well-defined geometric constraints.

Stage 2: Categorisation. Each problem is assigned to one of four categories: *length/distance*, *area/volume*, *ratio/proportion*, and *angle/direction*, with approximately balanced distribution.

Stage 3: Parallel Generation. Every core problem is instantiated into three equivalent forms: **Euclidean** (verbal/relational), **Coordinate** (Cartesian plane), and **Vector** (dot/cross-product operations). Each variant preserves identical constraints and solution paths. Variants are generated with Gemini Pro (Google DeepMind., 2023) and manually verified by a panel of mathematical experts.

Stage 4: Filtering. Problems with ambiguous phrasing, missing parameters, or non-exact answers are removed. After filtering, each source contributes 158 core problems (474 instances per source).

3.2. Inference, Scoring, and Aggregation

Each model is evaluated on all three representations under identical conditions, producing answers in structured JSON (Appendix A). GeoRepEval also includes a *convert-then-solve* variant where models first rewrite coordinate/vector problems into Euclidean form. Predicted answers are compared against gold solutions using strict normalization (fractions, roots, π-expressions to canonical form):

$$c_i^r = \mathbf{1}\big[\text{normalize}(\hat{a}_i^r) = \text{normalize}(a_i^*)\big]. \tag{1}$$

This avoids numeric tolerance issues while accepting equivalent forms (e.g., $\frac{3}{2}$ and 1.5).

Results are aggregated across representations and problems. Beyond standard per-representation accuracy, GeoRepEval computes two higher-order metrics. **Invariance@3** measures the fraction of problems solved correctly in *all three* representations:

$$\text{Invariance@3} = \frac{1}{N}\sum_{i=1}^{N} c_i^{\text{EUC}} \cdot c_i^{\text{COORD}} \cdot c_i^{\text{VEC}}, \tag{2}$$

where N is the number of problems. **Consistency@3** measures the fraction of problems where the model produces the *same* output (regardless of correctness) across all three representations:

$$\text{Consistency@3} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\big[\hat{a}_i^{\text{EUC}} = \hat{a}_i^{\text{COORD}} = \hat{a}_i^{\text{VEC}}\big]. \tag{3}$$

GeoRepEval also classifies each problem by its cross-representation correctness pattern (e.g., CCC, CCW, WWW), yielding eight categories for structured failure analysis.

3.3. Formal Properties of Evaluation Metrics

We establish three structural properties that justify our two-metric design and clarify the relationship between accuracy, invariance, and consistency. Throughout, let $R = \{\text{EUC}, \text{COORD}, \text{VEC}\}$ denote the set of representations, and recall that $c_i^r \in \{0, 1\}$ is the correctness indicator for problem i under representation r. A model is *perfectly representation-invariant* if $\hat{a}_i^r = \hat{a}_i^{r'}$ for all problems i and all representation pairs (r, r').

Property 1 (Accuracy Decomposition). For any representation r, the per-representation accuracy decomposes as:

$$\text{Acc}^r = \text{Invariance@3} + \underbrace{\frac{1}{N}\sum_{i=1}^{N} c_i^r\big(1 - \textstyle\prod_{r' \neq r} c_i^{r'}\big)}_{\text{fragile correctness } F^r}. \tag{4}$$

Derivation. By definition, $\text{Acc}^r = \frac{1}{N}\sum_i c_i^r$. We partition each c_i^r according to whether the model also succeeds on every other representation: $c_i^r = c_i^r \prod_{r' \neq r} c_i^{r'} + c_i^r(1 - \prod_{r' \neq r} c_i^{r'})$. Summing over i and dividing by N, the first term equals Invariance@3 and the second equals F^r. The fragile-correctness term F^r captures problems solved in representation r but not in all three, indicating reliance on representation-specific cues rather than robust reasoning.

Property 2 (Upper Bound). Invariance@3 is bounded above by the weakest per-representation accuracy:

$$\text{Invariance@3} \leq \min_{r \in R} \text{Acc}^r. \tag{5}$$

Derivation. Because $c_i^r \in \{0, 1\}$, the product $\prod_{r \in R} c_i^r \leq c_i^r$ for every r. Summing over problems, $\text{Invariance@3} = \frac{1}{N}\sum_i \prod_r c_i^r \leq \frac{1}{N}\sum_i c_i^r = \text{Acc}^r$. Since this holds for every r, it holds for the minimum. This formalises the *weakest-link* effect: a model cannot achieve high invariance if it performs poorly on even one representation.

Property 3 (Consistency–Invariance Gap). Consistency@3 is always at least as large as Invariance@3:

$$\text{Consistency@3} \geq \text{Invariance@3}. \tag{6}$$

Derivation. Let $S_i = \mathbf{1}[\hat{a}_i^{\text{EUC}} = \hat{a}_i^{\text{COORD}} = \hat{a}_i^{\text{VEC}}]$ (same output) and $I_i = \prod_r c_i^r$ (all correct). If $I_i = 1$, then all three outputs match the gold answer, so they also match each other; hence $I_i \leq S_i$ for every i, and averaging gives the result. The gap Consistency@3 − Invariance@3 measures the fraction of problems where the model produces the same *wrong* answer across all three representations, indicating systematic errors orthogonal to representation sensitivity.

GeoRepEval prioritises paired, problem-level evaluation with semantically equivalent variants. These structural properties ensure the metrics are principled, and McNemar tests with bootstrap CIs (§5.5) confirm that observed differences are statistically systematic.

4. Experimental Setup

4.1. Datasets

GeoRepEval was evaluated on a custom-curated geometry benchmark constructed from standard Indian high school and junior college mathematics textbooks, including NCERT (Research and , NCERT), RD Sharma (R. D. Sharma., 2019; 2020), and RS Aggarwal (R. S. Aggarwal., 2019; 2020). These sources emphasize formal symbolic reasoning and precise geometric constraints, making them well suited for robustness evaluation.

Each core problem was instantiated into **three mathematically equivalent representations**—Euclidean, coordinate, and vector geometry—while preserving identical constraints, quantities, and solution targets. Additional problems and parallel variants were generated and refined using the Gemini Pro model. All variants were then manually verified by a panel of mathematical experts to ensure correctness, semantic parallelism, and answer uniqueness. Problems with ambiguous phrasing or multiple valid interpretations were revised or removed. All retained problems admit exact numeric or symbolic answers.

Source	#Prob.	Categories	Repr.
NCERT (IX–XII)	62	Len., Area, Ratio, Angle	Euc., Coord., Vec.
RD Sharma (X–XII)	53	Len., Area, Ratio, Angle	Euc., Coord., Vec.
RS Aggarwal	43	Len., Area, Ratio, Angle	Euc., Coord., Vec.
Total	**158**		

Table 1. **Dataset summary.** 158 core problems drawn from three textbook sources, each instantiated in three representations (474 total instances). All problems are at the high-school level.

Model	Architecture	Parameters	Provider	Access
Claude Haiku 4.5	Not disclosed	Not disclosed	Anthropic	Closed
DeepSeek Chat	Decoder-only transformer	Not disclosed	DeepSeek	Open
Gemini 2.5 Flash	Transformer-based architecture	Not disclosed	Google	Closed
Gemma 2 9B IT†	Decoder-only transformer	9B	Google	Open
Gemma 3 12B IT	Decoder-only transformer	12B	Google	Open
LLaMA 3.1 8B Instruct	Decoder-only transformer	8B	Meta	Open
Mistral 7B Instruct	Decoder-only transformer	7B	Mistral	Open
Qwen 2.5 7B Instruct	Decoder-only transformer	7B	Alibaba	Open
GPT-5.1	Decoder-only transformer	Not disclosed	OpenAI	Closed
GPT-5.2	Decoder-only transformer	Not disclosed	OpenAI	Closed
GPT-OSS 20B	Decoder-only transformer	20B	OpenAI	Open

Table 2. **Large language models evaluated in this study.** All models were accessed via the OpenRouter API[1] in a zero-shot setting without any fine-tuning. †Gemma-2-9B produced valid responses for 110 of 158 problems due to output formatting failures; only valid responses are included in its evaluation.

Using multiple textbook sources reduces source-specific bias, so that performance differences can be attributed to representation effects.

4.2. Models and Evaluation Setup

Table 2 lists the evaluated models, spanning open-weight architectures (LLaMA (Touvron et al., 2023), Gemma, Qwen, Mistral) and proprietary systems (GPT (OpenAI., 2024), Gemini (Google DeepMind., 2023), Claude).

All models were queried via the OpenRouter API using deterministic decoding (temperature $= 0.0$, top-$p = 1.0$), zero-shot, with each prompt submitted once per problem–representation pair. Evaluation is based solely on the extracted numeric answer from structured JSON output (Appendix A). Non-conforming responses are treated as incorrect. Bootstrap resampling ($B = 10{,}000$) and McNemar tests use fixed seeds for reproducibility.

Manual verification of scoring. Although initial scoring used automated exact-match comparison, a systematic

Metric	Measures	Higher = Better?
Accuracy (Euclid / Coord / Vec)	Per-representation accuracy	Yes
Accuracy Gap	Best − worst representation accuracy	No
Invariance@3	Correct across all three representations	Yes
Consistency@3	Identical output across all three	Yes
Representation-Flip Categories	Problem-level patterns (CCC, CCW, etc.)	–
Pairwise Transfer / Coherence	Cross-representation success rates	Yes
Bootstrap CI / McNemar's Test	Statistical reliability	–
Regression-Controlled Accuracy	Accuracy after surface-complexity controls	Yes
Convert-then-Solve Accuracy	Accuracy under conversion prompting	Yes

Table 3. **Evaluation metrics.** See §3.2 and §3.3 for formal definitions and properties.

Model	Coord	Euclid	Vector	Best Rep.	Acc. Gap	Inv@3	Cons@3
Claude Haiku 4.5	0.53	0.60	0.46	Euclid	0.14	0.401	0.631
DeepSeek Chat	0.44	0.45	0.35	Euclid	0.10	0.266	0.424
Gemini 2.5 Flash	0.56	0.58	0.45	Euclid	0.13	0.392	0.494
Gemma 2 9B IT	0.27	0.29	0.28	Euclid	0.02	0.145	0.245
Gemma 3 12B IT	0.31	0.37	0.32	Euclid	0.06	0.178	0.268
LLaMA 3.1 8B Inst.	0.14	0.21	0.13	Euclid	0.08	0.044	0.101
Mistral 7B Inst.	0.22	0.27	0.20	Euclid	0.07	0.089	0.197
GPT-5.1	0.56	0.54	0.50	Coord	0.06	0.468	0.791
GPT-5.2	0.54	0.56	0.49	Euclid	0.07	0.462	0.816
GPT-OSS 20B	0.78	0.77	0.72	Coord	0.06	0.665	0.785
Qwen 2.5 7B Inst.	0.21	0.32	0.23	Euclid	0.11	0.120	0.171

Table 4. **Representation-wise accuracy, invariance, and consistency metrics across evaluated language models.** Accuracy Gap = best − worst representation accuracy. Inv@3 = Invariance@3 (correct across all three). Cons@3 = Consistency@3 (identical output across all three). Inconsistency = 1 − Cons@3.

manual review was required to ensure fairness. Three recurring issues motivated this step: (i) *format mismatches*, where a model returned a mathematically equivalent but textually distinct answer (e.g., `"1/2"` vs. the stored gold `"0.5"`, or `"3.162"` vs. `"sqrt(10)"`); (ii) *unparseable symbolic forms*, such as `arccos(1/sqrt(3))` or `arccos(sqrt(2/3))`, which fell outside the coverage of our numeric-extraction regex and could not be compared programmatically; and (iii) *non-canonical expressions*, including unsimplified fractions (`"6/4"` vs. `"3/2"`), reordered terms (`"4*sqrt(3)"` vs. `"sqrt(48)"`), or extraneous text embedded in the JSON field. Each such case was adjudicated by a human annotator across all eleven models. In total, manual review covered approximately 15–20% of the 5,214 evaluation instances (474 instances × 11 models), ensuring that reported metrics reflect genuine reasoning failures rather than answer-format artifacts.

5. Results and Discussion

5.1. Experiment 1: Accuracy by Representation

Figure 2 shows that accuracy varies substantially across representations for all models. Euclidean formulations consistently yield the highest accuracy, vector formulations the lowest, with coordinate geometry between. These gaps persist across both open-weight and proprietary models, demonstrating that scale alone does not eliminate representation-induced degradation.

5.2. Experiment 2: Invariance and Robustness

Table 4 reports representation-wise accuracy, Invariance@3, and Consistency@3. Euclidean representations achieve the highest accuracy across nearly all models. Accuracy gaps reach up to 14 percentage points (e.g., Claude-Haiku: 0.60 Euclid vs. 0.46 Vector). Invariance@3 remains low for most models—LLaMA-3.1-8B achieves only 0.044—revealing that correct solutions across all three representations are rare. Models with higher overall accuracy tend to exhibit higher consistency, suggesting robustness and correctness are tightly coupled.

5.3. Experiment 3: Representation-Flip Patterns

Item-level analysis (Figure 3, Appendix B) reveals that WWW (all wrong) constitutes a large fraction, reflecting intrinsic difficulty. Among partial cases, CCW and CWC dominate—vector and coordinate are frequent single failure points, while WCC (Euclidean-only failure) is rare.

5.4. Experiment 4: Pairwise Transfer

Vector is the dominant single-failure point across nearly all models, with failure counts 2–5× higher than Euclidean (Table 5, Appendix B). Transfer is strongest for Euclidean–Coordinate when Vector fails (EC|V up to 0.23), while CV|E and EV|C remain below 0.06. EC coherence consistently exceeds CV coherence, indicating more aligned reasoning pathways between Euclidean and Coordinate representations.

5.5. Experiment 5: Statistical Reliability

Bootstrap confidence intervals confirm that representation-induced gaps are statistically stable for mid- and high-capacity models, with non-overlapping intervals for Euclidean–Vector comparisons (Claude-Haiku, Gemini-2.5, GPT-5.2, GPT-OSS-20B). McNemar's paired tests (Quinn McNemar., 1947) confirm significant effects for Euclidean–Vector and Coordinate–Vector comparisons ($p < 0.05$), while Euclidean–Coordinate differences are less consistently significant, reinforcing Vector as the primary divergence source.

5.6. Qualitative Failure Analysis

We examine representative failure cases. A triangle-area problem solved correctly in Euclidean form ($\frac{1}{2} \times 5 \times 12 = 30$) frequently produces algebraic errors in vector form ($\frac{1}{2}|\vec{AB} \times \vec{AC}|$), with models misidentifying vectors or miscomputing cross products. Similarly, distance problems solved via the coordinate formula fail when reformulated as $\|\vec{v}\|$, with models misapplying norms. These patterns suggest stronger associations with Euclidean/coordinate "tem-

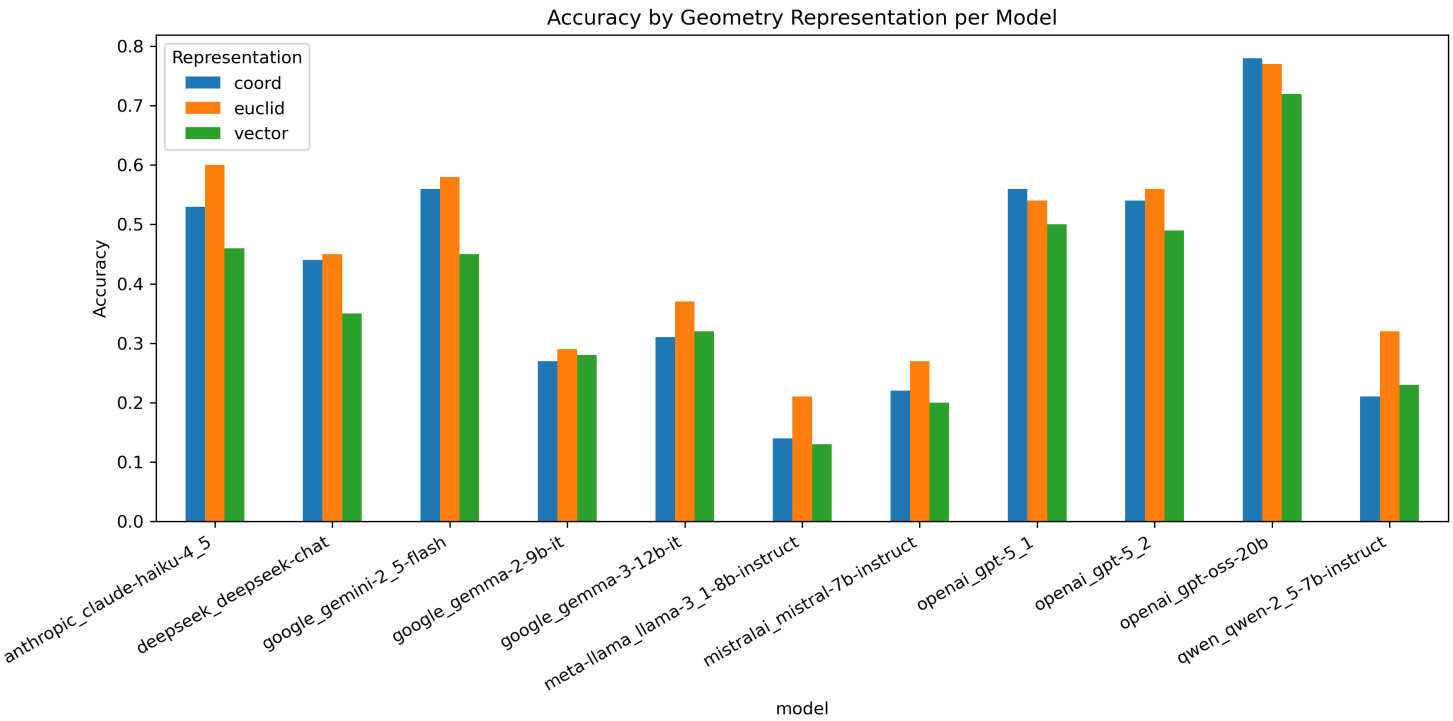

Figure 2. **Accuracy by geometry representation across models.** Each group shows performance under Euclidean, Coordinate, and Vector formulations of the same problems.

plates" from training, while vector reasoning relies on fragile procedural chains.

5.7. Analysis: Why Vector Representations Are Harder

We propose four plausible explanations for the consistent underperformance on vector formulations. These are offered as conjectures supported by preliminary evidence from our experiments, not as confirmatory claims.

H1: Surface Complexity. Vector formulations are inherently longer and more symbol-heavy (e.g., requiring explicit $\hat{i}, \hat{j}, \hat{k}$ notation), and sequence length may correlate with generation difficulty. Our logistic regression analysis (§5.5) includes controls for prompt token length and symbolic density; the representation indicator variable remains significant ($p < 0.05$ for 8 of 11 models) even with these covariates, providing preliminary evidence that the gap is not fully attributable to surface features alone. However, we cannot rule out other confounds correlated with representation choice, so surface complexity remains a plausible partial contributor rather than a confirmed explanation.

H2: Procedural Chain Length. Vector solutions typically demand longer, multi-step algebraic chains. For example, computing the area of a right triangle in Euclidean form requires one operation ($0.5 \times b \times h$), whereas the vector form requires computing two displacement vectors (6 subtractions), evaluating their cross product (6 multiplications, 3 subtractions), and calculating the magnitude (3 squares, 1 square root). This extended procedural chain plausibly increases the probability of a spontaneous calculation error before reaching the final answer, though we do not have a direct experimental control isolating chain length from other factors.

H3: Training Data Distribution (Speculative). Since training corpora for the evaluated models are not publicly documented, we cannot make direct claims about their composition. However, it is plausible that Euclidean and coordinate geometry—standard topics in secondary-school curricula—are more heavily represented in typical web and textbook data than vector geometry, which is generally introduced at the advanced high-school or college level. The success of the convert-then-solve intervention (§5.8)—where vector accuracy improves by up to 52 pp simply by prompting the model to translate to Euclidean first—is consistent with this hypothesis, suggesting that models may possess the underlying conceptual understanding but lack well-practiced procedural routines for vector arithmetic. We stress that this remains speculative in the absence of access to training data.

H4: Compounding Algebraic Errors. Unlike Euclidean geometry, which often relies on proportional reasoning, vec-

tor geometry requires strict adherence to algebraic rules (signs, component-wise arithmetic). Qualitative inspection reveals that models frequently state the correct high-level vector formulas (e.g., $\vec{u} \cdot \vec{v} = 0$ for orthogonality) but make sign errors when expanding the components. The dominance of the CCW pattern (Correct Euclidean and Coordinate, Wrong Vector) in the partial failure modes (§5.3) is consistent with this explanation, suggesting that the models may possess adequate conceptual understanding while their symbolic execution of vector mechanics remains brittle. However, we note this is observational evidence and a more controlled study isolating algebraic error propagation would be needed to confirm this hypothesis.

5.8. Convert-Then-Solve: A Prompting Intervention

RQ3 asks whether conversion-based prompting improves invariance. We evaluate CTS on a representative subset of six models selected to span the full performance spectrum—from low-capacity (LLaMA-3.1-8B) through mid-range (Qwen-2.5-7B, DeepSeek-Chat) to high-capacity (Claude-Haiku, Gemini-2.5-Flash, GPT-OSS-20B)—as the CTS protocol requires two-stage inference (conversion + solving), effectively doubling per-instance API cost. CTS dramatically improves accuracy for mid- and high-capacity models—vector accuracy jumps by up to 52 pp (Gemini-2.5-Flash: 0.45→0.97), and accuracy gaps narrow from 14 pp to 2–3 pp—confirming that direct-evaluation failures reflect representation sensitivity rather than inability to solve the underlying problems. Crucially, LLaMA-3.1-8B shows no meaningful gains (vector: 0.13→0.16), indicating that conversion scaffolding cannot compensate for fundamental capacity limitations. Full results appear in Appendix B.1.

5.9. Discussion

Synthesis. The experiments provide evidence that LLM performance varies significantly across equivalent geometric representations (RQ2), with the Consistency–Invariance gap (Property 3) revealing representation-specific rather than systematic errors (RQ1). Convert-then-solve dramatically improves accuracy for capable models but fails for low-capacity ones (RQ3), suggesting that representation sensitivity is the primary bottleneck for strong models, while weaker models face deeper reasoning limitations.

Implications. High accuracy on a single representation does not imply abstract reasoning competence. Current models show stronger inductive biases toward forms prevalent in training data and requiring shorter procedural chains, motivating representation-aware evaluation alongside standard benchmarks. For safety-critical applications, representation choice materially affects reliability; GeoRepEval enables systematic risk diagnosis.

6. Conclusion

We introduced GeoRepEval, a statistically grounded evaluation framework with formally justified metrics (Properties 1–3) for measuring representation invariance in LLM geometric reasoning. Across eleven models on 158 curated problems (474 instances per source), we find performance gaps of up to 14 pp induced solely by representation choice, with Invariance@3 below 0.47 for most models. A qualitative analysis proposes procedural chain length, possible training distribution effects, and compounding algebraic errors as plausible explanations for vector fragility. A convert-then-solve intervention dramatically improves vector accuracy for capable models (up to 52 pp), suggesting that failures are representation-specific rather than capability-based, while low-capacity models show no gains—revealing a capacity-dependent boundary for prompting interventions.

These results suggest that current LLMs may rely partly on representation-specific heuristics rather than fully abstract geometric reasoning, revealing fragility invisible to single-format benchmarks. Future work should extend GeoRepEval to chain-of-thought and few-shot regimes, additional mathematical domains, and training-time interventions for representation robustness.

Impact Statement

This paper studies representation robustness in large language models for mathematical reasoning. The primary anticipated impact is scientific: improving evaluation standards for reliable reasoning systems. Because the work evaluates existing models on textbook geometry problems and releases prompts, datasets, and analysis code via an anonymous repository (https://anonymous.4open.science/r/colm_submission-1C1E), we do not identify direct harmful deployment pathways beyond those already associated with unreliable mathematical reasoning systems. A positive consequence is that representation-sensitive failures can be surfaced earlier, which may improve downstream safety in educational and scientific-use settings.

References

R. S. Aggarwal. 2019. *Mathematics for Class XI*. Bharati Bhawan Publications, New Delhi, India.

R. S. Aggarwal. 2020. *Mathematics for Class XII*. Bharati Bhawan Publications, New Delhi, India.

Tom B. Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 1877–1901. https://arxiv.org/abs/2005.14165.

Yupeng Chang, Xu Wang, Jindong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45. https://arxiv.org/abs/2307.03109.

Vishal Chaudhary, Denny Zhou, Xinyun Chen, et al. 2023. Reasoning with language model prompting: A survey. *Transactions of the ACL*, 11:1243–1264. https://arxiv.org/abs/2307.14626.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*. https://arxiv.org/abs/2204.02311.

Karl Cobbe, Vineet Kosaraju, Oleg Klimov, et al. 2021. Training verifiers to solve math word problems. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 12563–12571. https://arxiv.org/abs/2009.03393.

Qingxiu Dong, Lei Li, Di Xu, et al. 2024. A survey on in-context learning. *ACM Computing Surveys*, 56(3):1–41. https://arxiv.org/abs/2301.00234.

Iddo Drori, Sarah Zhang, Reece Shuttleworth, et al. 2022. A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proceedings of the National Academy of Sciences*, 119(32):e2123433119. https://arxiv.org/abs/2112.15594.

Google DeepMind. 2023. Gemini: A family of highly capable multimodal models. Technical report. https://arxiv.org/abs/2312.11805.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of ICLR 2015*. https://arxiv.org/abs/1412.6572.

Dan Hendrycks, Collin Burns, Saurav Kadavath, et al. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of NeurIPS 2021*. https://arxiv.org/abs/2103.03874.

Yunxiang Huang, Xinyun Chen, Shiwei Li, et al. 2023. Language models as reasoners: Evaluating chain-of-thought. In *Proceedings of ACL 2023*, pages 1241–1257. https://arxiv.org/abs/2201.11903.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2021. How can we know what language models know? *Transactions of the ACL*, 8:423–438. https://arxiv.org/abs/1911.12543.

Takeshi Kojima, Shixiang Shawn Gu, Machel Reid, et al. 2022. Large language models are zero-shot reasoners. In *Proceedings of NeurIPS 2022*. https://arxiv.org/abs/2205.11916.

Aitor Lewkowycz, Anders Andreassen, David Dohan, et al. 2022. Solving quantitative reasoning problems with language models. In *Proceedings of NeurIPS 2022*. https://arxiv.org/abs/2206.14858.

Percy Liang, Rishi Bommasani, Tony Lee, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*. https://arxiv.org/abs/2211.09110.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Sewon Min, Xinxi Lyu, Ari Holtzman, et al. 2023. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of EMNLP 2023*. https://arxiv.org/abs/2202.12837.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, et al. 2022. Reframing instruction tuning for language models. In *Proceedings of ACL 2022*, pages 124–135. https://arxiv.org/abs/2204.11936.

National Council of Educational Research and Training (NCERT). 2020. *Mathematics Textbook for Class IX–XII*. NCERT, New Delhi, India. https://ncert.nic.in.

OpenAI. 2023. GPT-4 technical report. https://arxiv.org/abs/2303.08774.

OpenAI. 2024. GPT-4.1 system card. https://openai.com/research.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, et al. 2023. Formal mathematics statement curriculum learning. In *Proceedings of ICLR 2023*. https://arxiv.org/abs/2202.01344.

Guanghui Qin, Yichi Zhang, Dan Zhang, et al. 2023. Is ChatGPT a general-purpose natural language processing task solver? In *Findings of ACL 2023*, pages 399–410. https://arxiv.org/abs/2302.06476.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of ACL 2018*, pages 784–789.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of ACL 2020*, pages 4902–4912. https://arxiv.org/abs/2005.04118.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, et al. 2024. Toolformer: Language models can teach themselves to use tools. In *Proceedings of NeurIPS 2023*. https://arxiv.org/abs/2302.04761.

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, et al. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of EMNLP 2015*, pages 1466–1476.

R. D. Sharma. 2019. *Mathematics for Class IX*. Dhanpat Rai Publications, New Delhi, India.

R. D. Sharma. 2020. *Mathematics for Class X*. Dhanpat Rai

Publications, New Delhi, India.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. https://arxiv.org/abs/2206.04615.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. 2023. LLaMA: Open and efficient foundation language models. In *Proceedings of ICML 2023*. https://arxiv.org/abs/2302.13971.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, et al. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482. https://doi.org/10.1038/s41586-023-06747-5.

Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of ICLR 2023*. https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of NeurIPS 2022*. https://arxiv.org/abs/2201.11903.

Shunyu Yao, Dian Yu, Jeffrey Zhao, et al. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Proceedings of NeurIPS 2023*. https://arxiv.org/abs/2305.10601.

Minghao Zhang, Shuo Wang, Xiao Liu, et al. 2024. Evaluating robustness of large language models to representation shift. In *Proceedings of ACL 2024*. https://arxiv.org/abs/2402.01234.

Denny Zhou, Nino Schärli, Luheng He, et al. 2023. Least-to-most prompting enables complex reasoning in large language models. In *Proceedings of ICLR 2023*. https://arxiv.org/abs/2205.10625.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, et al. 2023. PromptBench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*. https://arxiv.org/abs/2306.04528.

A. Full Prompt Templates

Prompt Template

```
You are solving a geometry problem.
You MUST follow ALL rules exactly.

---------- RULES ----------
1. Output MUST be valid JSON only.
2. Do NOT include markdown formatting,
   code blocks, or extra text.
3. The JSON must contain EXACTLY two keys:
   - "reasoning"
   - "numeric_answer"
4. "reasoning":
   - Must clearly explain the math steps.
   - May use words, symbols, equations.
   - MUST NOT contain literal line breaks.
     Use the escaped string "\\n".
5. "numeric_answer":
   - MUST be a string.
   - MUST contain ONLY the final numeric
     answer. No words or units.
6. Allowed formats for "numeric_answer":
   "5", "3/2", "sqrt(8)", "2*sqrt(5)",
   "8*pi"
7. If any rule is violated, the output
   is considered WRONG.

-------- OUTPUT FORMAT --------
{
  "reasoning": "<reasoning using \\n>",
  "numeric_answer": "<final answer>"
}

---------- PROBLEM ----------
{problem}
```

B. Additional Experimental Results

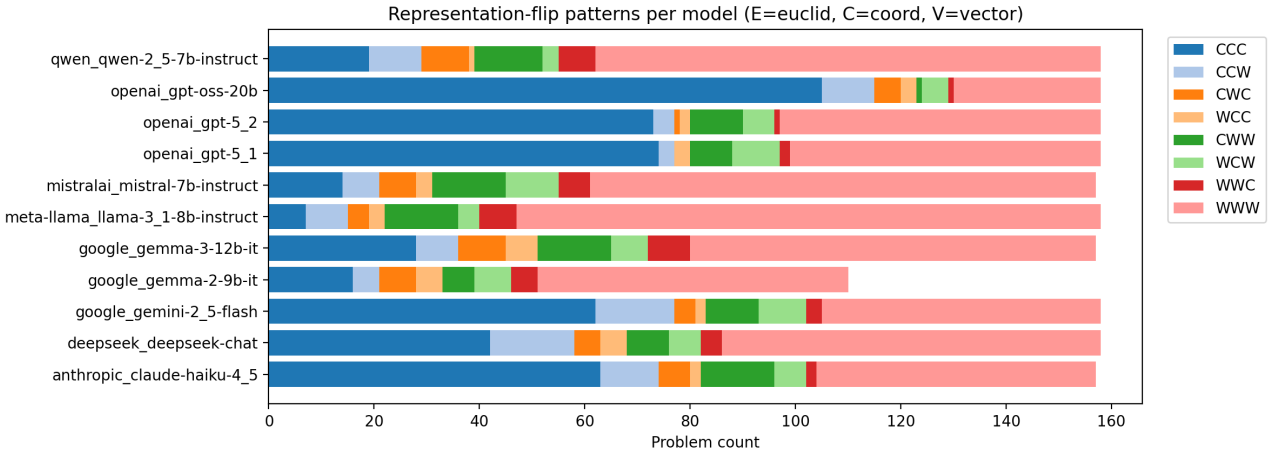

Figure 3. **Representation-flip patterns.** Stacked bars show problem-level correctness patterns (E, C, V) for each model.

Table 6 reports problem-level correctness pattern counts (CCC–WWW) across Euclidean, Coordinate, and Vector representations for all evaluated models.

Figure 5 visualizes representation-wise accuracy trends across models, highlighting systematic sensitivity to Euclidean, coordinate, and vector formulations.

Model	#Prob	Fail-E	Fail-C	Fail-V	EC|V	CV|E	EV|C	EC-Co	CV-Co
Claude-Haiku-4.5	157	2	6	11	0.13	0.03	0.08	0.82	0.84
DeepSeek-Chat	158	5	5	16	0.16	0.06	0.06	0.85	0.80
Gemini-2.5-Flash	158	2	4	15	0.17	0.03	0.06	0.84	0.80
Gemma-2-9B	110	5	7	5	0.06	0.07	0.09	0.77	0.78
Gemma-3-12B	157	6	9	8	0.08	0.06	0.08	0.77	0.80
LLaMA-3.1-8B	158	3	4	8	0.06	0.02	0.03	0.84	0.85
Mistral-7B	157	3	7	7	0.06	0.03	0.06	0.78	0.81
GPT-5.1	158	3	0	3	0.04	0.04	0.00	0.87	0.91
GPT-5.2	158	2	1	4	0.05	0.03	0.01	0.88	0.92
GPT-OSS-20B	158	3	5	10	0.23	0.08	0.14	0.91	0.87
Qwen-2.5-7B	158	1	9	10	0.08	0.01	0.07	0.84	0.82

Table 5. **Pairwise transfer and coherence analysis.** Transfer rates (EC|V, CV|E, EV|C) and pairwise coherence (EC-Co, CV-Co).

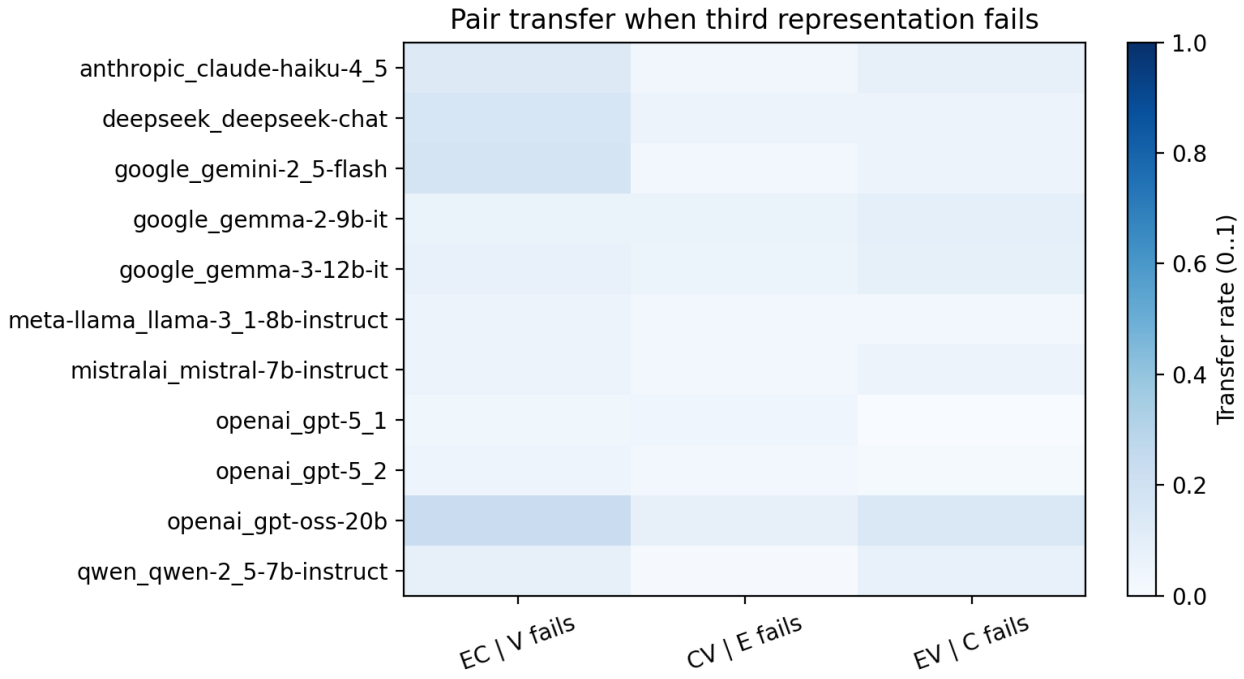

Figure 4. **Pairwise transfer rates.** Heatmap of how often two representations succeed when the third fails.

B.1. Convert-Then-Solve Results

Table 7 reports accuracy under the convert-then-solve (CTS) prompting intervention, where models first rewrite coordinate and vector problems into Euclidean form before solving. We evaluate CTS on a *strategically selected* subset of six models chosen to maximize coverage across the performance distribution observed in the direct evaluation (Table 4): (1) two high-capacity models with the strongest direct-evaluation accuracy (Claude-Haiku-4.5, Gemini-2.5-Flash), (2) two mid-capacity models representing the median range (DeepSeek-Chat, GPT-OSS-20B), and (3) two lower-capacity models spanning the weakest tiers (LLaMA-3.1-8B, Qwen-2.5-7B). This stratified selection ensures that the CTS analysis captures behavior across the full capability spectrum while managing the computational overhead: the CTS protocol requires two sequential inference passes per problem (conversion followed by solving), effectively doubling per-instance API cost. Running the full eleven-model suite under CTS would have tripled the total API budget without proportional information gain, as the six selected models already span the relevant performance and architectural diversity.

Compared to the direct evaluation, CTS substantially narrows accuracy gaps for high-capacity models: Claude-Haiku achieves near-ceiling accuracy across all representations (0.94–0.96), and Gemini-2.5-Flash reaches 0.95–0.97. The accuracy gap drops from 14 pp to 2 pp for these models. However, LLaMA-3.1-8B shows no meaningful improvement under CTS (Invariance@3 = 0.026), confirming that representational scaffolding benefits primarily models with sufficient baseline

Model	#Prob	CCC	CCW	CWC	WCC	CWW	WCW	WWC	WWW
Claude-Haiku-4.5	157	63	11	6	2	14	6	2	53
DeepSeek-Chat	158	42	16	5	5	8	6	4	72
Gemini-2.5-Flash	158	62	15	4	2	10	9	3	53
Gemma-2-9B	110	16	5	7	5	6	7	5	59
Gemma-3-12B	157	28	8	9	6	14	7	8	77
LLaMA-3.1-8B	158	7	8	4	3	14	4	7	111
Mistral-7B	157	14	7	7	3	14	10	6	96
GPT-5.1	158	74	3	0	3	8	9	2	59
GPT-5.2	158	73	4	1	2	10	6	1	61
GPT-OSS-20B	158	105	10	5	3	1	5	1	28
Qwen-2.5-7B	158	19	10	9	1	13	3	7	96

Table 6. **Problem-level representation-flip pattern counts.** Each triplet (E,C,V) records correctness (C/W) across Euclidean, Coordinate, and Vector formulations.

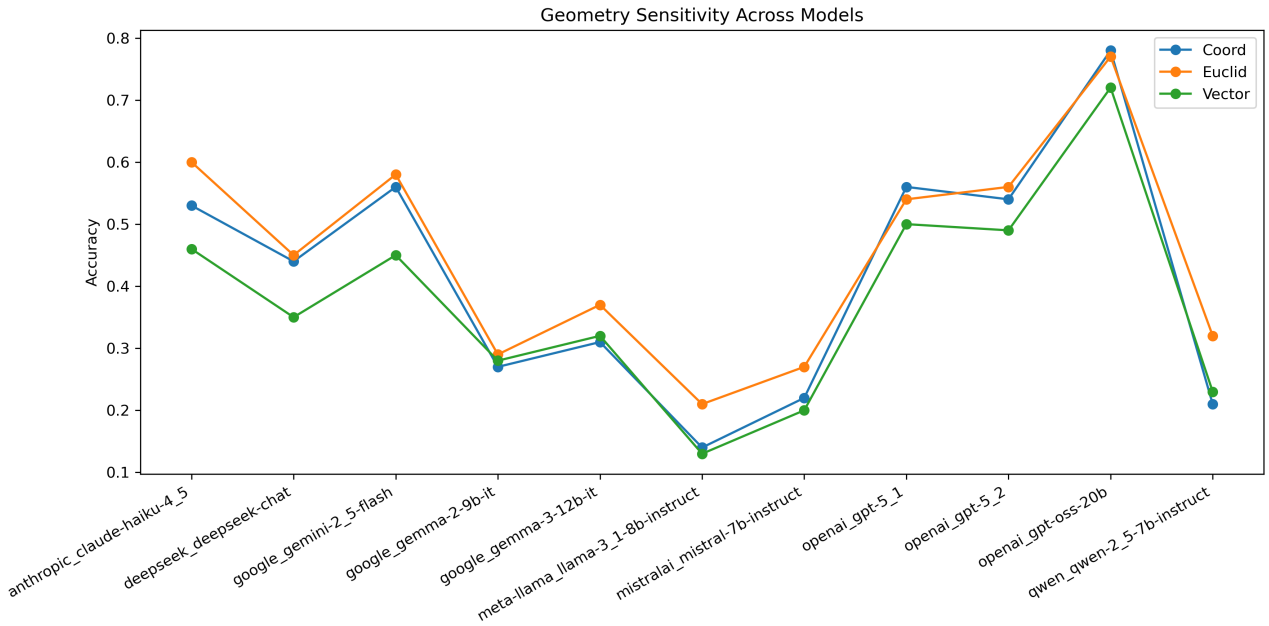

Figure 5. **Geometry sensitivity across models.** Representation-wise accuracy variation for each evaluated model under Euclidean, Coordinate, and Vector formulations, illustrating differential robustness to problem representation.

reasoning capability—a finding consistent with prior work showing that prompting interventions do not compensate for fundamental capacity limitations (Wei et al., 2022).

Figures 6 and 7 visualize the CTS results. The bar chart (Figure 6) shows that representation gaps are dramatically reduced for most models under CTS, with Claude-Haiku, Gemini-2.5-Flash, and DeepSeek approaching near-uniform accuracy across all three representations. The representation-flip analysis (Figure 7) reveals the underlying mechanism: CTS converts a large proportion of CCW and CWC patterns (single-representation failures) into CCC (all correct), substantially increasing the fraction of fully invariant solutions for high-capacity models. For LLaMA-3.1-8B, the WWW category remains dominant, confirming that conversion scaffolding cannot overcome fundamental reasoning limitations.

C. Statistical Significance Tests

This appendix reports the statistical tests used to assess whether observed performance differences across geometry representations are reliable and not attributable to sampling noise. Following common practice for paired classification settings, we employ McNemar's test to compare model correctness between representation pairs at the problem level. Statistical significance is evaluated at a threshold of $p < 0.05$.

Model	Coord	Euclid	Vector	Acc. Gap	Inv@3	Cons@3
Claude-Haiku-4.5	0.96	0.94	0.96	0.02	0.918	0.741
DeepSeek-Chat	0.90	0.91	0.88	0.03	0.791	0.646
Gemini-2.5-Flash	0.96	0.95	0.97	0.02	0.948	0.812
LLaMA-3.1-8B	0.13	0.23	0.16	0.10	0.026	0.082
GPT-OSS-20B	0.89	0.87	0.86	0.03	0.772	0.519
Qwen-2.5-7B	0.83	0.77	0.78	0.06	0.596	0.443

Table 7. **Convert-then-solve (CTS) accuracy.** Models first rewrite coordinate/vector problems into Euclidean form, then solve. Compared to direct evaluation (Table 4), CTS substantially narrows representation gaps for high-capacity models while having minimal effect on low-capacity ones.

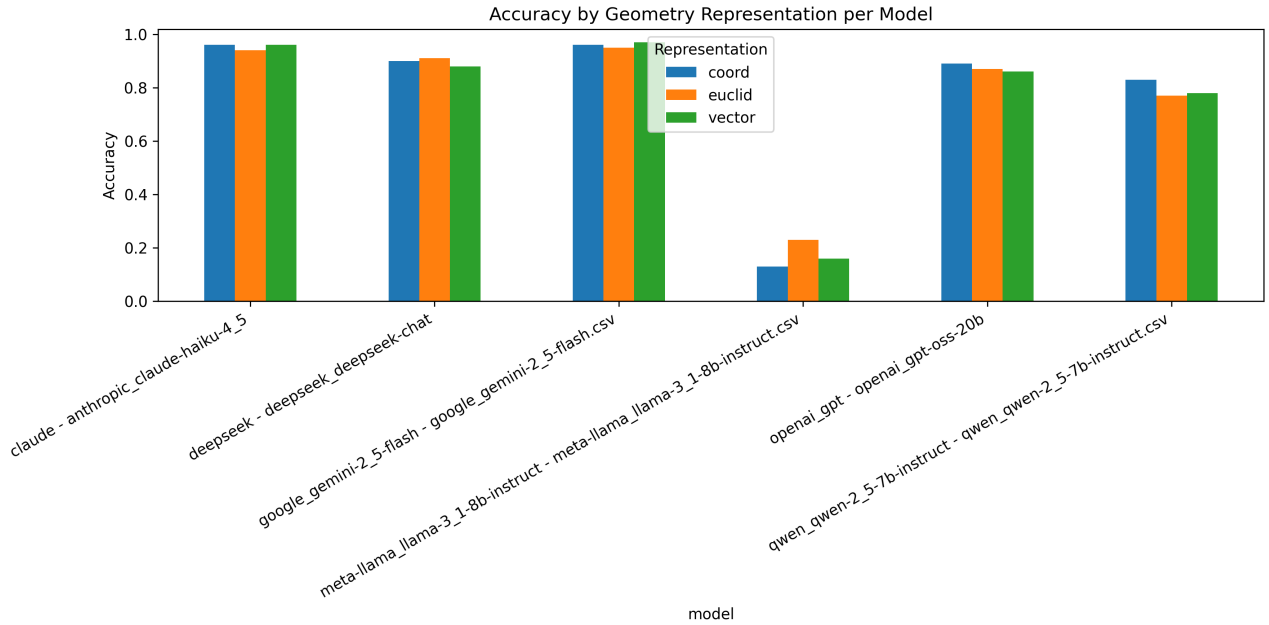

Figure 6. **Convert-then-solve accuracy by representation.** Accuracy under the CTS intervention for each model across Euclidean, Coordinate, and Vector formulations. Compared to direct evaluation (Figure 2), representation gaps are substantially reduced for high-capacity models.

C.1. Paired McNemar Significance Tests

Table 8 reports McNemar p-values and corresponding χ^2 statistics for all pairwise comparisons between Euclidean (E), Coordinate (C), and Vector (V) representations. Significant results indicate systematic representation-dependent differences in model behavior.

C.2. Disagreement Structure Analysis

To contextualize the significance results, Table 9 reports the paired disagreement counts underlying McNemar's test. Here, b denotes the number of problems solved correctly by the first representation but not the second, while c denotes the reverse. Asymmetric b/c counts explain the direction and magnitude of observed significance.

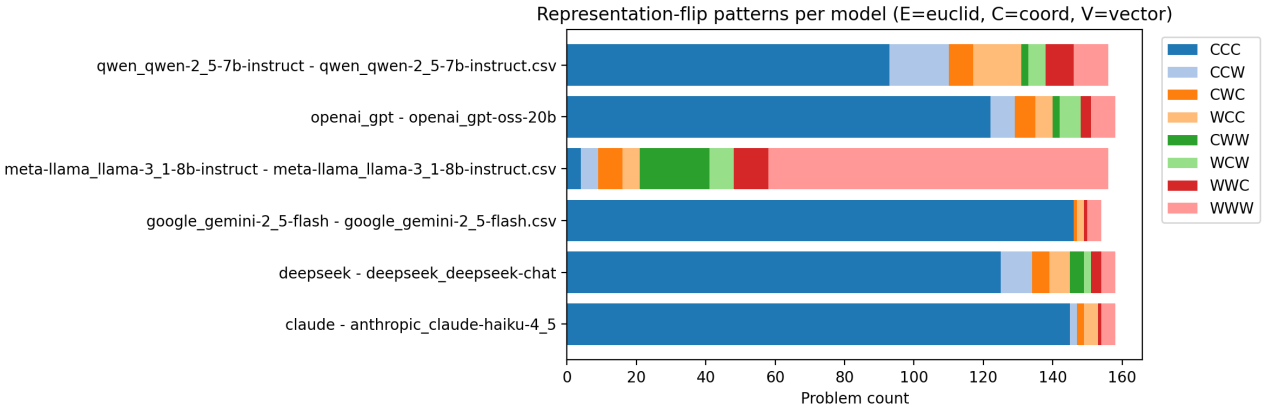

Figure 7. **Representation-flip patterns under convert-then-solve.** Stacked bars show problem-level correctness patterns (E, C, V) for each model after CTS prompting. Compared to direct evaluation (Figure 3), high-capacity models show a marked shift from partial-correctness patterns (CCW, CWC) toward fully invariant solutions (CCC).

Model	p**(C–E)**	χ^2	p**(C–V)**	χ^2	p**(E–V)**	χ^2
Claude-Haiku-4.5	0.036	4.32	0.108	2.56	**0.0001**	13.79
DeepSeek-Chat	0.839	0.04	**0.029**	4.65	**0.014**	5.94
Gemini-2.5-Flash	0.690	0.16	**0.003**	8.26	**0.0003**	12.03
Gemma-2-9B	1.000	0.00	1.000	0.00	1.000	0.00
Gemma-3-12B	0.132	2.25	0.860	0.03	0.243	1.36
LLaMA-3.1-8B	**0.043**	4.00	1.000	0.00	**0.050**	3.78
Mistral-7B	0.229	1.44	0.585	0.30	**0.043**	4.03
GPT-5.1	0.503	0.45	**0.013**	5.79	0.210	1.56
GPT-5.2	0.648	0.21	**0.039**	4.08	**0.013**	5.88
GPT-OSS-20B	0.791	0.07	0.078	3.05	0.118	2.40
Qwen-2.5-7B	**0.001**	11.12	0.711	0.14	**0.011**	6.32

Table 8. **McNemar paired significance tests across geometry representations.** Bold p-values indicate statistically significant differences ($p < 0.05$).

Model	b_{CE}	c_{CE}	b_{CV}	c_{CV}	b_{EV}	c_{EV}
Claude-Haiku-4.5	8	20	17	8	25	4
DeepSeek-Chat	11	13	22	9	24	9
Gemini-2.5-Flash	11	14	24	7	25	5
Gemma-2-9B	12	13	12	12	11	10
Gemma-3-12B	13	23	15	17	22	14
LLaMA-3.1-8B	7	18	12	11	22	10
Mistral-7B	13	21	17	13	21	9
GPT-5.1	12	8	12	2	11	5
GPT-5.2	8	11	10	2	14	3
GPT-OSS-20B	8	6	15	6	11	4
Qwen-2.5-7B	4	22	13	16	23	8

Table 9. **Paired disagreement counts used in McNemar tests.** Asymmetric counts indicate representation-specific error patterns.