INVARIANTBENCH: CAN LARGE LANGUAGE MODELS EXHIBIT INHERENT REASONING CONSISTENTLY ACROSS EQUIVALENT TRANSFORMATIONS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning is often attributed to large language models (LLMs), yet it remains unclear whether they operate over underlying semantics or rely on surface-form patterns. Existing benchmarks evaluate correctness on fixed problem instances, but overlook a fundamental property of reasoning: invariance under semantics-preserving transformations. If a model truly understands a problem, its predictions should remain consistent across equivalent representations. We introduce INVARIANTBENCH, a benchmark of 1159 seed problems $\times$ 4 invariant forms, spanning 16 tasks across 3 reasoning families and 12 fine-grained invariance axes. Each problem is paired with multiple semantically equivalent variants under a strict invariance contract, enabling evaluation beyond accuracy to measure consistency across representations. Experiments on 15 frontier and open-weight LLMs reveal a persistent invariance gap: invariant reformulations reduce accuracy by up to 30% for strong models, while the gap between solving at least one versus all four variants reaches 60%. Full consistency remains below 5% for most open-weight models, compared with $\sim 48\%$ for the two strongest models and 98.6% for human experts. These results show that high base accuracy substantially overestimates reasoning ability, and establish invariance as a necessary axis for evaluating and improving robust language understanding.

1 INTRODUCTION

Large language models (LLMs) are now routinely described as performing *reasoning*, multi-step deductive, mathematical, algorithmic, and semantic inference across a wide range of benchmarks (Wei et al., 2022b; Bubeck et al., 2023; OpenAI et al., 2024; Anthropic, 2024). Headline scores on GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), BIG-BENCH HARD (Suzgun et al., 2022), and MMLU (Hendrycks et al., 2021a) have risen so quickly that several recent reports interpret them as evidence for emergent general-purpose reasoning (Bubeck et al., 2023; Wei et al., 2022a). A natural question follows: *do these scores measure reasoning, or do they measure pattern-matching against a particular surface form of the problem?* If the former, a model that solves a problem under one phrasing should solve it under any semantically-equivalent phrasing. If the latter, we should expect surface perturbations, even those preserving every mathematical and logical entailment, to systematically degrade performance.

The empirical evidence is increasingly that the latter possibility is non-trivial. Mirzadeh *et al.* (Mirzadeh et al., 2025) show that simply replacing the proper nouns and numerical values in GSM8K items reduces accuracy across every model they evaluate, with degradations as large as 65% when a single, semantically-irrelevant clause is added. Turpin *et al.* (Turpin et al., 2023) establish that chain-of-thought rationales can be systematically biased by features the rationale itself does not mention, undermining the interpretation of the chain as a window into the model's reasoning. Ribeiro *et al.* (Ribeiro et al., 2020) and Naik *et al.* (Naik et al., 2018) have long documented brittleness to surface paraphrase in NLP systems. Concerns about benchmark contamination (Xu et al., 2024; Sainz et al., 2023; Golchin & Surdeanu, 2024) sharpen the worry: if a model can recall an answer because it has seen the exact item, then accuracy on the original instance is uninformative about its competence on *any* instance. The community has responded with dynamic benchmarks (Kiela et al., 2021; Srivastava et al., 2023), contamination-resistant evaluations (Zhu et al., 2024; White et al.,

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

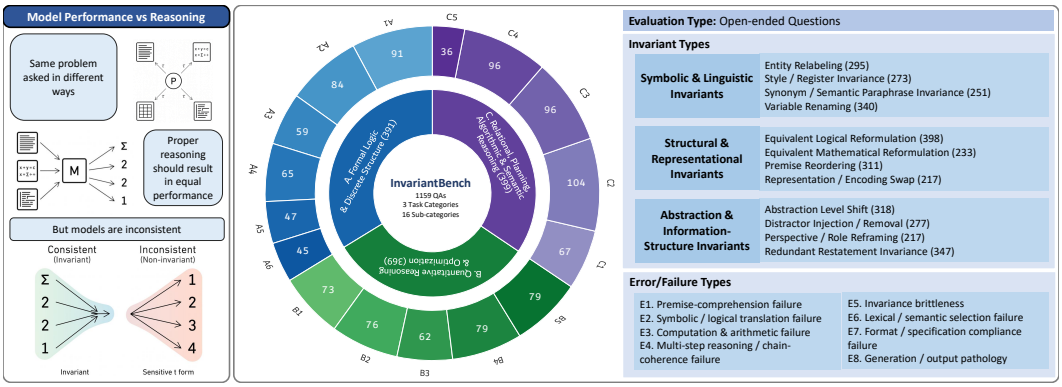

Figure 1: **Overview of INVARIANTBENCH.** The benchmark evaluates invariance in reasoning by comparing model performance across semantically equivalent transformations of the same problem (left), and consists of 1159 seed questions spanning 3 reasoning families and 16 subcategories (center). It applies 12 controlled invariance transformations grouped into symbolic, structural, and abstraction-level shifts, and analyzes failures through 8 distinct error modes capturing where reasoning breaks under representation change (right).

2024), and proposals for testing reasoning under controlled perturbations (Mirzadeh et al., 2025; Li et al., 2024; Gulati et al., 2024). None of these efforts has produced a unified, multi-axis framework that makes *representational invariance* a first-class evaluation primitive.

Reasoning, in the classical sense, is closed under semantically-preserving transformations: an inference is valid in virtue of the relations between its propositions, not in virtue of how those propositions are spelled (Russell & Norvig, 2010; Marcus, 2020). Renaming the variables of a logical formula, permuting independent premises, switching from a fraction to its decimal expansion, or rephrasing a deduction as an equivalent contrapositive, none of these should change the answer for an agent that operates over the underlying semantics. Conversely, any perturbation that does change the answer reveals a feature of the model's behaviour that is *not* a feature of the problem. We do not claim invariance under semantics-preserving transformation is sufficient evidence of reasoning; rather, we treat systematic inconsistency as evidence that a model's correct answers on the base formulation are not reliably grounded in the underlying reasoning steps, whether the failure originates in language understanding, symbol grounding, or reasoning proper.

We introduce INVARIANTBENCH, a benchmark that pairs seed reasoning problems with three semantics-preserving transformations drawn from twelve invariance axes (Table 7). The transformations span three families: *symbolic and linguistic invariance*, *structural and representational invariance*, and *abstraction and information-structure invariance*. All transformations preserve the underlying semantics, including the correct answer and entailment structure, as defined in §2. Our core contributions are:

1. We introduce a **formal notion of representational invariance** for LLM evaluation, together with a semantic-preservation contract that all transformations in our work satisfy (§ 2).
2. We present **INVARIANTBENCH**, a large-scale reasoning benchmark of 1159 seed problems $\times$ 4 invariant forms, spanning 16 tasks across 3 reasoning families and 12 fine-grained invariance axes, with verifiable answers and an automated evaluation protocol (§3).
3. We conduct a **systematic empirical study** of 15 frontier and open-weight LLMs, revealing a persistent invariance gap: accuracy drops of $1-5\%$ for strong models and $> 10\%$ for smaller ones, with very low full-consistency rates ($< 5\%$ for most models), showing that high base accuracy overestimates true reasoning ability (§4).
4. We provide **comprehensive failure analysis and diagnostic interventions**, introducing eight main failure modes from model outputs that link errors to invariant axes and localize breakdowns across perception, transformation, and composition, and further evaluating **SFT** and **GRPO**, showing SFT induces memorization while GRPO improves reasoning and cross-form generalization without closing the invariance gap.

Across all evaluated models, we observe substantial invariance degradation and low consistency, with full consistency below 5% for most models and reaching only $\sim$ 48% for the best models,

compared to $> 98\%$ for humans. Notably, invariant reformulations preserve problem semantics yet can change model performance by up to 5% across variants in average, while the at-least-one vs. all-four gap reaches $30 - 60\%$ in propitiatory and large open source models. These results show that high base accuracy overestimates true reasoning robustness, and that invariance provides a stricter evaluation axis. By operationalizing invariance at scale, INVARIANTBENCH establishes a foundation for evaluating and improving representation-robust reasoning in language models.

2 PRELIMINARIES

Reasoning under Representational Invariance. A reasoning problem is a pair $(P, a^\star)$ where P is a problem statement and $a^\star \in \mathcal{Y}$ is the verifiable ground-truth answer. A surface form $s \in \Sigma^\star$ denotes P under a fixed semantics $[\![\cdot]\!]$, i.e., $[\![s]\!] = (P, a^\star)$. Two forms s, s' are semantically equivalent, $s \equiv s'$, if $[\![s]\!] = [\![s']\!]$. A transformation $\tau : \Sigma^\star \to \Sigma^\star$ is semantics-preserving if

$$[\![\tau(s)]\!] = [\![s]\!] = (P, a^\star), \tag{1}$$

i.e., $\tau(s) \equiv s$. INVARIANTBENCH defines a fixed set of $K = 12$ transformations $\mathcal{T}$ (Table 7). Each seed s_0 yields variants $s_k = \tau_k(s_0)$. A model $M : \Sigma^\star \to \mathcal{Y}$ predicts answers from surface forms. For a seed and its variants, we define $\hat{a}_M(P) = \big(M(s_0), M(s_1), \ldots, M(s_K)\big)$. Representation invariance requires

$$M(s) = M(s') \quad \forall\, s \equiv s'. \tag{2}$$

This property is independent of correctness, isolating sensitivity to representation from accuracy. We identify two failure modes: (i) *accuracy fragility*, where $M(s_0) = a^\star$ but $M(s_k) \neq a^\star$ for some k; and (ii) *inconsistency*, where predictions vary across equivalent forms even when some outputs match $a^\star$. INVARIANTBENCH measures both, separating stability from seed-specific performance.

The Twelve Invariance Axes. The twelve transformations used in INVARIANTBENCH are grouped into three families: symbolic & linguistic, structural & representational, and abstraction & information-structure, as outlined in Figure 1 and Table 7. The symbolic & linguistic family includes entity relabeling, style/register variation, synonym/paraphrase substitution, and variable renaming. The structural & representational family covers equivalent logical reformulation, equivalent mathematical reformulation, premise reordering, and representation/encoding swaps. The abstraction & information-structure family includes abstraction level shifts, distractor injection/removal, perspective/role reframing, and redundant restatement invariance. Each transformation $\tau \in \mathcal{T}$ is designed to preserve semantics (Eq. equation 1) while altering surface form, structure, or information organisation, and all variants are human-verified to ensure answer equivalence.

Task Design and Distribution. INVARIANTBENCH spans three principal reasoning families: Formal Logic & Discrete Structure (391), Quantitative Reasoning & Optimization (369), and Relational, Planning, Algorithmic & Semantic Reasoning (399), ensuring balanced coverage across symbolic, numerical, and structural regimes. Each family is further decomposed into fine-grained sub-types capturing key reasoning primitives, including logic-based constraints (e.g., multi-quantifier and set algebra), algebraic and probabilistic systems, and graph- and language-based reasoning tasks. This hierarchical design isolates distinct reasoning capabilities while maintaining controlled overlap across related modes, enabling systematic analysis of performance degradation under invariant transformations. Full task design details, category definitions, and explanations are provided in § B.

Metrics: Accuracy, Consistency, and Stability. To evaluate whether models exhibit genuine representation-invariant reasoning rather than surface-level pattern matching, we use a compact set of metrics operating at both aggregate and conditional levels (full details in § F). Each sample is evaluated across four invariant variants, forming a correctness vector $\mathbf{x}_i = (x_{i,v})_{v \in \mathcal{V}} \in \{0,1\}^4$ with $c_i = \sum_{v \in \mathcal{V}} x_{i,v}$. At the *aggregate level*, we characterize robustness via the invariant mass index $\mathrm{IMI} = P(c = 4)$, capturing the fraction of samples solved correctly under all transformations. To capture *conditional stability*, we define the conditional invariance retention $\mathrm{CIR} = P\Big(\bigwedge_{j \in \mathcal{I}} x_{i,j} = 1 \;\Big|\; x_{i,\text{base}} = 1\Big)$, and the invariance collapse rate $\mathrm{ICR} = P\Big(x_{i,\text{base}} = 1 \wedge \sum_{j \in \mathcal{I}} x_{i,j} \leq 1\Big)$, which together distinguish transferable correctness from brittle, non-generalizing solutions. Pairwise dependence is measured via $\mathrm{PIDI}(j,k) = P(x_{i,j} = 0 \mid x_{i,k} = 0)$, indicating whether failures are independent or structurally coupled. We also summarize *structural behavior* using consistency entropy $\mathrm{CE} = \frac{1}{N} \sum_{i=1}^{N} H\big(\frac{c_i}{4}\big)$, and bilateral symmetry

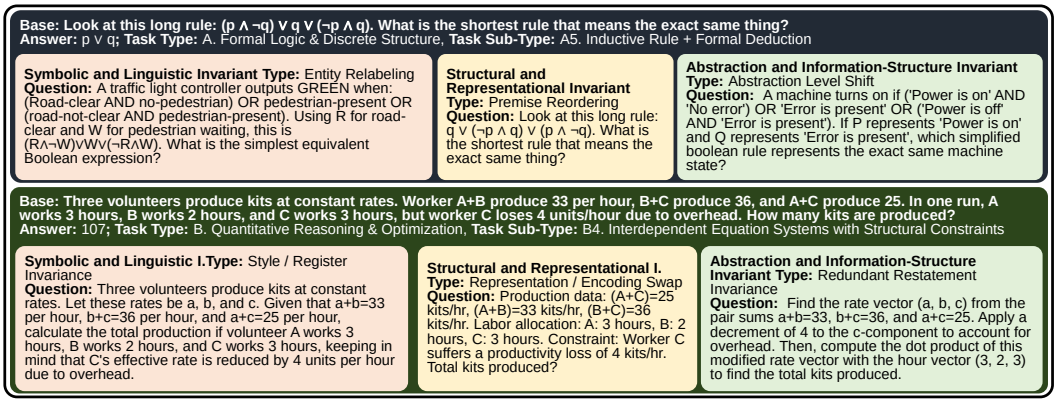

Figure 2: Two representative examples from INVARIANTBENCH.

$\mathrm{BSS} = 1 - \frac{|F^+ - F^-|}{F^+ + F^-}$, which together distinguish deterministic, symmetric reasoning from unstable or directionally biased behavior. Finally, we define invariance sensitivity through degradation gradients $g_{12} = \mathrm{Acc}(\mathcal{I}_1) - \mathrm{Acc}(\mathcal{I}_2), \quad g_{23} = \mathrm{Acc}(\mathcal{I}_2) - \mathrm{Acc}(\mathcal{I}_3)$, where violations of monotonicity indicate non-semantic artifacts in representation shifts. Together, these metrics define a unified evaluation framework that captures not only correctness, but also stability, transferability, and structural coherence under semantically equivalent transformations.

Scope and limitations of the contract. We focus on *open-ended* evaluation, where the answer space $\mathcal{Y}$ admits multiple mathematically equivalent forms (e.g., $\frac{12}{25} \equiv 0.48$). To ensure that Acc reflects semantic correctness rather than notation, we employ a verifier (Cobbe et al., 2021; Lightman et al., 2023) (§E.2) that normalizes numerical and symbolic expressions prior to comparison; residual notation-sensitive mismatches are tracked separately (§ I). Importantly, invariance is *not* a proxy for correctness: $\mathrm{Cons}(M) = 1$ is necessary but not sufficient, as a model may be consistently wrong across all variants. We therefore interpret correctness under consistency (e.g., IMI/CIR) as the strongest verifiable signal. Conversely, any $\mathrm{Cons}(M) < 1$ provides a constructive counterexample to representation-invariant reasoning under Eq. 2. Finally, invariance is *axis-dependent*: different transformations preserve semantics at different granularities (from strict logical equivalence to shifts in pragmatics or information structure). Accordingly, we report per-axis behaviour to distinguish superficial robustness from deeper representation-invariant competence.

3 INVARIANTBENCH BENCHMARK

We construct INVARIANTBENCH as a controlled evaluation suite for representational invariance in large language models. The benchmark is designed to systematically vary surface form while preserving semantic content, enabling precise measurement of whether models maintain consistent reasoning under equivalent reformulations. The dataset comprises 1159 seed problems spanning 16 core tasks (Table 7), organized into 3 principal reasoning categories: symbolic and linguistic reasoning, structural and representational reasoning, and abstraction and information-structure reasoning. Each seed problem is instantiated into four semantically equivalent surface forms, generated via a structured transformation pipeline that ensures invariance preservation under a formal semantic contract.

Invariance design. Each problem is transformed under a strict semantic invariance contract, ensuring that all variants preserve the underlying solution despite surface-level or representational perturbations (details in § D.2), as shown in Figure 2. All transformations are expert-verified.

Expert annotation protocol. Annotation is performed by domain-expert annotators trained in formal reasoning, logic, and mathematical problem design. Each seed problem is first independently validated for unambiguous solvability and then manually expanded into invariant variants under strict equivalence constraints. Annotators ensure that transformations do not introduce new solution paths or alter difficulty through extraneous reasoning shortcuts. A secondary expert review stage verifies semantic equivalence across all variants, while a final adjudication pass resolves edge cases involving ambiguity in logical or mathematical reformulations.

Quality control and verification. To ensure correctness, each problem–variant set is validated using a two-stage verification pipeline. First, automated consistency checks verify structural equivalence and detect invalid transformations (e.g., accidental constraint changes). Second, expert reviewers manually inspect sampled instances for semantic drift and unintended ambiguity. Lastly, we performed de-

duplication to remove any redundant questions. Only items passing both stages are included in the final benchmark. This results in a high-confidence dataset of invariantly paired reasoning tasks suitable for measuring representation-sensitive degradation in modern LLMs.

4 EXPERIMENTS AND EVALUATION

Models. We categorize the evaluated models into three groups. **Small open-source models** (below 30B) comprise Llama-3.2-1B-Instruct (Grattafiori et al., 2024), Ministral-3B-Instruct-2512 (Liu et al., 2026), Gemma-3-4B-Instruct (Team et al., 2025), Qwen3-8B (Yang et al., 2025a), and Gemma-4-26B-A4B-Instruct (Google, 2025). **Large open-source models** (more than 30B) comprise Olmo-3.1-32B-Instruct (Olmo et al., 2026), Aya-Expanse-32B (Dang et al., 2024), Llama-3.3-70B-Instruct (Grattafiori et al., 2024), Qwen3.6-Plus (Yang et al., 2025a), and DeepSeek-V3.1 (DeepSeek-AI et al., 2025). Lastly, **proprietary models** (closed source) include o4-mini (OpenAI, 2025b), GPT-5.2 (OpenAI, 2025a), Gemini-3-Flash (Team, 2025), and Claude 4.5 Haiku (Anthropic, 2025).

Human Baselines. Human baselines are derived from two groups: *base* (high-school to early undergraduate level) and *expert* (STEM graduates, Master's, and PhD students) (detailed in § E.3), with three annotators per group and majority voting used to get final responses for each question.

Evaluation Process and Metrics. For evaluation, models are prompted to generate free-form answers, which are subsequently assessed using a large language model judge, with full prompt specifications provided in § E.1. To validate the reliability of the automated judge, we perform a human–LLM agreement analysis, comparing judgments against expert human annotations as well as majority vote labels. As reported in Table **??**, the LLM judge (Gemma-4-26B-A4B-Instruct) achieves near-perfect alignment with human decisions across all task families. Overall agreement with majority vote is 0.99, with a corresponding Cohen's $\kappa_{\text{LLM-Human}} = 0.99$, while human inter-annotator agreement is $\kappa_{\text{Human}} = 0.97$, indicating that the judge operates at or above human consistency levels. This pattern holds across task categories, including Formal Logic & Discrete Structure ($\kappa = 1.0$), Quantitative Reasoning & Optimization ($\kappa = 1.0$), and Relational, Planning, Algorithmic & Semantic Reasoning ($\kappa = 0.99$). These results suggest that the LLM-based evaluator provides a highly reliable surrogate for human judgment in open-ended evaluation settings, with negligible deviation from expert consensus.

Improving Invariant Reasoning Capabilities. We evaluate supervised fine-tuning (SFT) and GRPO-based RL on GEMMA-3-1B-IT. Under SFT, a 40:60 stratified split leads to strong memorization, with gains driven by leakage across invariant variants. In the constrained `base` + `inv1` setting, unseen variants remain low (17.42, 14.58) while seen variants reach 41.42 and 41.75, indicating poor invariance. GRPO improves unseen variants to 19.17 and 20.69 within 5 epochs, a gain of $\sim$ 3–6 points over SFT, while seen variants reach 29.72 and 31.53. This reflects modest but limited generalization.

5 FINDINGS AND ANALYSIS

5.1 OVERALL RESULTS

Overall Accuracy. Table 1 shows substantial sensitivity to invariant reformulations across model families. Strong models achieve high Base accuracy but still lose performance on $\mathcal{I}_1$–$\mathcal{I}_3$: QWEN3.6-PLUS and GPT-5.2, for example, drop from 69.0% and 69.4% on Base to approximately 65–68% across invariant variants. Model scale alone does not explain performance: QWEN3-8B reaches 54.0% Base accuracy, substantially exceeding several larger open-source models, including LLAMA-3.3-70B-I (25.7%) and DEEPSEEK-V3.1 (20.0%).

Cross-Variant Consistency. The $\geq k$ statistics (Table 1) reveal a larger problem than the per-variant accuracy drops. Models often answer at least one formulation correctly but fail to maintain the same capability across all four variants. For GPT-5.2, 81.4% of samples have at least one correct answer, but only 47.6% are correct on all four, producing a 33.8-point Gap. The discrepancy is even larger for CLAUDE HAIKU 4.5 (77.4% vs. 17.4%, Gap = 60.0) and GEMINI-3-FLASH (61.1% vs. 17.3%, Gap = 43.8). Thus, many model successes are variant-specific rather than invariantly preserved.

Human–Model Gap. The *All Wrong* column in Table 1 shows that weaker models frequently fail on every formulation, with rates above 60% for several small and large open-source models. Stronger models reduce this absolute failure rate but still exhibit substantial cross-variant inconsistency. Humans show a qualitatively different pattern: HUMAN (EXPERT) achieves 99.9% Base accuracy, 98.6% all-four accuracy, and only a 1.4-point Gap, with 0.0% All Wrong. The large contrast suggests

Table 1: **Core accuracy and cross-variant consistency across evaluated models.** Base = original question; $\mathcal{I}_1$/$\mathcal{I}_2$/$\mathcal{I}_3$ denote Symbolic & Linguistic, Structural & Representational, and Abstraction & Information-Structure invariants, respectively. Accuracy is reported for the four variants; *All Wrong* is the proportion of samples where no variant is answered correctly. $\geq k$ denotes the proportion with at least k of 4 variants correct, with ≥ 4 equivalent to IMI. Gap = $\geq 1 - \geq 4$. **Bold** = best per column.

Model	*Accuracy (%) (↑)*				*All Wrong (%) (↓)*	*Variant Consistency (%) (↑)*				*Gap (%) (↓)*
	Base	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$		≥ 1	≥ 2	≥ 3	≥ 4	
Open-Source Models (Small) ≤30B										
Llama-3.2-1B-I	7.2	6.2	7.9	7.8	83.9	16.1	7.2	4.1	1.7	14.4
Ministral-3B-2512	11.1	10.1	12.1	11.0	74.6	25.4	12.4	5.2	1.2	24.2
Gemma-3-4B-I	10.3	10.6	11.3	9.9	75.1	24.9	10.0	5.2	2.0	22.9
Qwen3-8B	54.0	56.9	57.0	56.3	24.0	76.0	65.3	51.7	31.1	44.9
Gemma-4-26B-A4B-I	30.0	27.3	27.7	26.7	50.8	49.2	33.7	21.3	7.5	41.7
Open-Source Models (Large) >30B										
Olmo-3.1-32B-I	18.1	14.1	17.6	14.5	66.3	33.7	18.5	9.1	3.1	30.6
Aya-Expanse-32B	16.2	12.4	17.2	14.8	67.6	32.4	16.9	8.6	2.8	29.6
Llama-3.3-70B-I	25.7	24.5	28.1	24.2	49.7	50.3	29.7	16.8	5.7	44.6
Command-A (111B)	21.6	21.3	21.7	21.7	56.3	43.7	24.2	14.0	4.5	39.2
DeepSeek-V3.1 (671B)	20.0	16.7	20.8	16.4	60.5	39.5	22.1	10.1	2.2	37.3
Closed-Source Models										
Qwen3.6-Plus	69.0	64.7	64.6	65.6	20.5	79.5	72.3	64.4	47.7	31.8
o4-mini	22.3	15.5	19.1	15.4	63.4	36.6	21.0	11.0	3.8	32.8
GPT-5.2	69.4	66.6	67.9	68.5	18.6	81.4	75.3	68.1	47.6	33.8
Gemini-3-Flash	43.0	38.3	38.4	39.8	38.9	61.1	46.9	34.2	17.3	43.8
Claude Haiku 4.5	48.3	49.4	48.8	47.5	22.6	77.4	58.8	40.5	17.4	60.0
Human Baseline										
Human (Base)	90.0	90.4	90.1	89.6	**0.0**	**100.0**	99.5	95.2	65.5	34.5
Human (Expert)	**99.9**	**100.0**	**99.2**	**99.4**	**0.0**	**100.0**	**100.0**	**99.9**	**98.6**	**1.4**

Table 2: **Invariance stability metrics (Overall)**. **IMI** = Invariant Mass Index; **CIR** = Conditional Invariance Reliability; **ICR** = Invariance Collapse Rate; **BSS** = Bilateral Symmetry Score (1 = symmetric failures, 0 = all failures in one direction); **CE** = Consistency Entropy (lower = more structured reasoning); **PIDI** = pairwise failure co-occurrence probability.

	Global Stability (%)			*Novel (%)*		*Pairwise Failure Dep. (%)*		
Model	IMI↑	CIR↑	ICR↓	BSS↑	CE↓	$P(\mathcal{I}_1{=}0 \mid \mathcal{I}_2{=}0)$	$P(\mathcal{I}_1{=}0 \mid \mathcal{I}_3{=}0)$	$P(\mathcal{I}_2{=}0 \mid \mathcal{I}_3{=}0)$
Open-Source Models (Small) ≤30B								
Llama-3.2-1B-I	1.73	24.10	3.19	75.90	12.22	96.63	96.54	94.95
Ministral-3B-2512	1.21	10.85	6.56	81.85	21.04	92.84	93.12	90.99
Gemma-3-4B-I	1.98	19.33	5.35	72.18	19.53	92.80	92.62	91.48
Qwen3-8B	31.15	57.67	7.16	**98.08**	38.97	72.34	72.73	66.60
Gemma-4-26B-A4B-I	7.51	25.00	10.61	91.93	36.14	83.89	83.76	81.88
Open-Source Models (Large) >30B								
Olmo-3.1-32B-I	3.11	17.14	9.75	98.03	26.62	91.62	90.31	87.79
Aya-Expanse-32B	2.76	17.02	8.37	90.96	25.57	92.29	92.40	88.04
Llama-3.3-70B-I	5.69	22.15	10.79	89.75	38.62	83.19	84.30	80.09
Command-A	4.49	20.80	8.80	87.03	33.77	85.01	85.45	85.78
DeepSeek-V3.1	2.24	11.21	11.48	95.37	32.50	89.32	87.41	85.24
Closed-Source Models								
Qwen3.6-Plus	47.71	69.13	5.87	66.12	27.33	74.63	72.43	71.18
o4-mini	3.80	17.05	12.08	87.37	28.47	90.72	90.41	86.73
GPT-5.2	47.63	68.66	4.83	71.10	28.74	70.97	73.70	69.86
Gemini-3-Flash	17.34	40.36	11.22	82.84	37.88	80.11	80.09	79.94
Claude Haiku 4.5	17.43	36.07	12.77	96.98	52.12	66.61	67.27	63.65
Human Baseline								
Human (Base)	65.49	72.77	2.50	58.00	28.81	11.30	9.17	11.67
Human (Expert)	**98.62**	**98.70**	**0.00**	12.50	**1.14**	0.00	0.00	0.00

that invariant reformulation is not inherently difficult; rather, current models often fail to preserve a solved capability when the same underlying problem is expressed differently.

Invariance Stability. As shown in Table 2, across models, invariance stability metrics reveal a clear separation between scaling gains and true robustness, with most systems showing low IMI and CIR alongside non-trivial ICR, indicating that full consistency across variants remains rare. Smaller

Table 3: **Performance on invariance analysis (symmetry, entropy, recovery) metrics. BSS** = Bilateral Symmetry Score; F^+ = forward failure rate (base ✓ $\rightarrow$ inv ✗); F^- = backward failure rate (base ✗ $\rightarrow$ inv ✓); **CE** = Consistency Entropy (0 = fully structured, 1 = random); IRR_{full} = full invariant recovery given base failure; IRR_{part} = partial recovery ($\geq$1 invariant correct); $\text{IRR}_{\Delta} = \text{IRR}_{\text{full}} - (1 - \text{CIR})$ (positive = base harder than invariants); **FG** = Fragility Gradient monotonicity score (0–2, 2 = expected ordering); $\bar{g}$ = mean gradient magnitude in pp.

	Bilateral Symmetry			*Entropy*	*Invariant Recovery (%)*			*Fragility Gradient*	
Model	BSS↑	$F^+(\%)$	$F^-(\%)$	CE↓	IRR_{full} ↑	IRR_{part} ↑	IRR_{Δ}	FG	$\bar{g}$
Open-Source Models (Small) ≤30B									
lama-3.2-1B-I	75.90	5.44	8.89	12.22	0.09	9.57	-75.81	1	-0.78
Ministral-3B-2512	81.85	9.92	14.32	21.04	0.68	16.12	-88.47	1	-0.44
Gemma-3-4B-I	72.18	8.28	14.67	19.53	0.29	16.35	-80.38	1	+0.34
Qwen3-8B	**98.08**	22.86	22.00	38.97	10.51	47.84	-31.82	1	+0.26
Gemma-4-26B-A4B-I	91.93	22.52	19.15	36.14	2.71	27.37	-72.29	1	+0.30
Open-Source Models (Large) >30B									
Olmo-3.1-32B-I	98.03	15.01	15.62	26.62	0.84	19.07	-82.02	1	-0.18
Aya-Expanse-32B	90.96	13.46	16.13	25.57	0.93	19.26	-82.05	1	-1.21
Llama-3.3-70B-I	89.75	20.02	24.59	38.62	2.56	33.10	-75.29	1	+0.17
Command-A	87.03	17.08	22.17	33.77	1.54	28.27	-77.66	1	-0.22
DeepSeek-V3.1	95.37	17.77	19.50	32.50	1.94	24.38	-86.85	1	+0.17
Closed-Source Models									
Qwen3.6-Plus	66.12	21.31	10.53	27.33	3.90	33.98	-26.97	1	-0.43
o4-mini	87.37	18.46	14.32	28.47	1.11	18.42	-81.84	1	+0.04
GPT-5.2	71.10	21.74	11.99	28.74	11.55	39.15	-19.79	0	-0.95
Gemini-3-Flash	82.84	25.63	18.12	37.88	4.24	31.77	-55.40	0	-0.73
Claude Haiku 4.5	96.98	30.89	29.08	52.12	9.52	56.26	-54.41	**2**	+0.95
Human Baseline									
Human (Base)	58.00	24.50	10.01	28.81	**76.72**	**100.00**	+49.49	2	+0.39
Human (Expert)	12.50	1.29	0.09	**1.14**	0.00	100.00	-1.30	1	+0.30

models exhibit uniformly weak stability, with low IMI and CIR values and high CE, reflecting fragmented and largely unstructured reasoning under reformulation. Larger open-source and closed-source models improve substantially in CIR and IMI (e.g., QWEN3.6-PLUS and GPT-5.2), but still retain non-zero ICR and elevated CE, showing that correct reasoning often fails to persist across all invariant axes. Pairwise dependency scores (PIDI) remain high for most models, indicating strongly correlated failure modes rather than independent errors, though they are notably lower in stronger systems, suggesting partial decoupling of invariant weaknesses. In contrast, human baselines achieve near-perfect CIR and IMI with negligible ICR and near-zero PIDI, establishing that robust invariance is achievable in principle but not yet matched by current models.

Symmetry, Entropy and Recovery. Table 3 shows that most models achieve high bilateral symmetry (BSS), but this does not translate into robust reasoning, as failure rates remain substantial and invariant recovery is weak. Open-source models have low full recovery ($\text{IRR} * \text{full} \leq 10.51\%$) and uniformly negative $\text{IRR} * \Delta$, while closed-source models show modestly stronger recovery, with GPT-5.2 reaching 11.55% full recovery and Claude Haiku 4.5 achieving the highest partial recovery (56.26%). Consistency entropy varies substantially across models, indicating that symmetry alone does not capture behavioural consistency. Most models obtain FG$= 1$, with GPT-5.2 and Gemini-3-Flash at FG$= 0$ and Claude Haiku 4.5 at FG$= 2$. In contrast, Human (Base) achieves 76.72% full and 100% partial recovery, positive IRR_{Δ} (+49.49), and FG$= 2$, revealing a substantial gap between model symmetry and human-like invariant reasoning.

Invariance Sensitivity Profile. Table 4 reveals substantial variation in invariance retention across models. Among open-source models, QWEN3-8B is a notable outlier, achieving $\sim$75% TCIR across all three invariants, despite its smaller size. Closed-source models QWEN3.6-PLUS and GPT-5.2 perform even better, with TCIR above 81% across all invariants. In contrast, most other models remain around 40–65%, indicating substantial sensitivity to invariant transformations. Collapse contribution is generally high (typically 55–75%), suggesting that failures are strongly associated with invariance changes rather than being uniformly distributed. Human performance remains markedly stronger: Human (Base) retains $\sim$90% across invariants, while Human (Expert) approaches 100% with near-zero collapse for $\mathcal{I}_1$ and $\mathcal{I}_3$. Overall, model robustness varies substantially across invariant types, with strong models achieving high retention but still falling short of the near-perfect consistency observed in human reasoning.

Table 4: **Invariance sensitivity profile (Overall)**. **TCIR** = conditional invariance retention; **TCC** = collapse contribution rate; **RIS** = relative sensitivity (pp deviation from model mean; positive = easier, negative = harder). Subscripts 1/2/3 denote $\mathcal{I}_1/\mathcal{I}_2/\mathcal{I}_3$.

	Cond. Inv. Retention (%)			*Collapse Contribution (%)*			*Rel. Sensitivity (pp)*		
Model	$TCIR_1$	$TCIR_2$	$TCIR_3$	$TCC_1 \downarrow$	$TCC_2 \downarrow$	$TCC_3 \downarrow$	RIS_1	RIS_2	RIS_3
Open-Source Models (Small) ≤30B									
Llama-3.2-1B-I	38.71	49.46	48.39	78.42	59.71	61.87	-1.06	+0.67	+0.50
Ministral-3B-2512	39.66	47.46	43.05	68.94	60.00	67.66	-0.98	+1.01	-0.11
Gemma-3-4B-I	42.56	45.33	39.79	69.00	62.01	71.62	+0.08	+0.77	-0.61
Qwen3-8B	74.80	74.91	74.12	60.28	59.22	53.90	+0.82	+0.91	+0.30
Gemma-4-26B-A4B-I	55.44	56.32	54.21	66.87	62.54	64.40	-0.65	-0.21	-1.25
Open-Source Models (Large) >30B									
Olmo-3.1-32B-I	41.94	52.17	42.97	71.33	60.14	69.93	-1.94	+1.51	-1.59
Aya-Expanse-32B	38.40	53.07	45.87	75.27	57.82	67.27	-2.74	+2.01	-0.32
Llama-3.3-70B-I	48.71	55.92	48.03	68.56	57.22	68.04	-1.13	+2.50	-1.47
Command-A	48.72	49.70	49.70	64.93	64.93	65.51	-0.28	+0.15	+0.15
DeepSeek-V3.1	42.36	52.62	41.48	68.04	60.12	70.09	-1.75	+2.30	-2.10
Closed-Source Models									
Qwen3.6-Plus	81.34	81.24	82.43	69.89	62.50	53.98	-1.27	-1.36	-0.41
o4-mini	42.45	52.36	42.22	73.31	62.50	72.64	-2.57	+1.05	-2.66
GPT-5.2	81.87	83.46	84.20	70.13	**55.84**	55.84	-1.49	-0.20	+0.41
Gemini-3-Flash	62.71	62.85	65.11	66.03	66.03	62.50	-1.55	-1.46	-0.08
Claude Haiku 4.5	63.88	63.10	61.43	62.15	58.41	64.25	+0.91	+0.31	-0.99
Human Baseline									
Human (Base)	90.42	90.08	89.65	48.21	58.93	55.36	+0.39	+0.05	-0.38
Human (Expert)	**100.00**	**99.22**	**99.40**	**0.00**	100.00	**0.00**	+0.37	-0.41	-0.23

5.2 Error and Reasoning Failure Mode Analysis

We conducted a systematic two-pass qualitative coding of all failures produced during evaluation across all scored question–response pairs, yielding an eight-category failure mode taxonomy aligned to the reasoning pipeline (§ I): premise comprehension (E1), problem formalisation (E2), local computation (E3), chain coherence (E4), invariance robustness (E5), lexical-semantic selection (E6), output specification (E7), and generation pathology (E8). Aggregate failure rates span from 31.3% (GPT-5) to 84.4% (OLMO-3.1-32B), with a clear tier structure: top models (GPT-5, QWEN3.6-PLUS) are primarily limited by E5 and E4 failures, while lower-tier models are dominated by E1 and E3 errors, indicating that constraint integration and basic computation break down before higher-order reasoning effects are observable. The most prevalent single failure mode is *dropped or ignored constraints* (E1.1), which accounts for a disproportionate share of errors in algebraic and multi-constraint. Across all systems, *Abstraction Level Shift* is the most uniformly damaging invariant transformation, with breakage rates in the 11–21% range even for top-performing models, consistently exceeding all other invariance axes and confirming that failures reflect genuine representational fragility rather than memorization artifacts. Mitigation analysis identifies five systemic root causes: fragile input representation binding, weak cross-form consistency, limited multi-step state persistence, over-reliance on surface priors, and decoding instability under structured constraints.
Effective interventions therefore require joint enforcement across invariance families rather than axis-specific augmentation, explicit constraint-aware attention regularization, and structured intermediate state tracking. Full error taxonomy, per-model breakage rates by sub-type and invariant axis, concrete failure examples, and detailed mitigation strategies are provided in § I.

5.3 Performance Improvement Approaches

We investigate how different optimization strategies influence invariance learning and cross-form generalization under identical evaluation conditions.
Supervised Fine-Tuning (SFT). As shown in Figure 3, SFT exhibits a sharp and uniform rise under the *All Invariants* regime, with all four variants converging tightly to $\sim 54\%$ by epoch 5 (Base: 54.53, Inv1: 54.36, Inv2: 54.18, Inv3: 53.84), indicating strong cross-variant leakage and memorization of overlapping surface forms rather than invariant reasoning. In contrast, under *Base and Inv1* training, performance improves primarily on seen variants (Base: 42.36, Inv1: 42.45), while unseen Inv2/Inv3 remain low (17.17 and 14.24), creating a persistent $\sim$ 25–28 point gap relative to the corresponding seen-variant performance. This divergence demonstrates that SFT learns form-specific mappings that

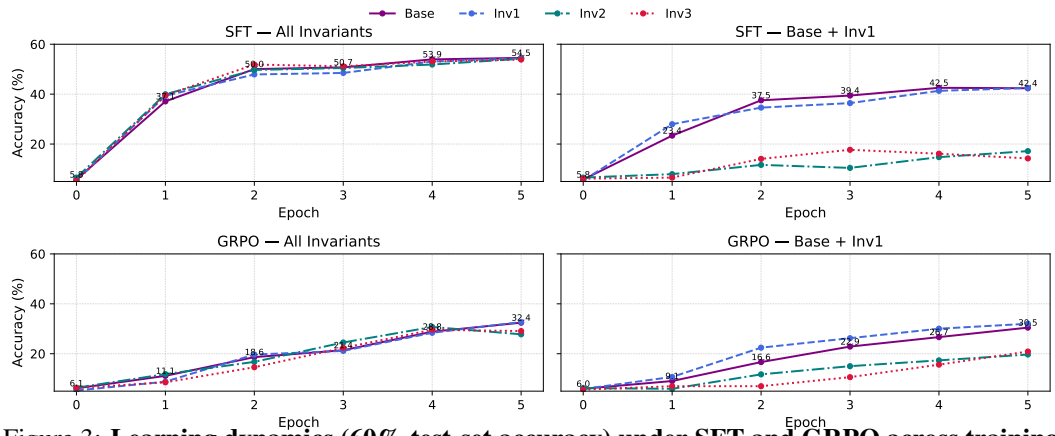

Figure 3: **Learning dynamics (60% test-set accuracy) under SFT and GRPO across training regimes.** SFT shows rapid gains but strong regime sensitivity and poor generalization under partial invariants, while GRPO exhibits tighter base–invariant coupling, indicating improved robustness.

fail to transfer across semantically equivalent reformulations. Full training details and epoch-wise results are provided in § J.1.

Reinforcement Learning (RL) with GRPO (Group Relative Policy Optimization). In Figure 3, GRPO shows a slower but more differentiated learning trajectory, with early divergence across variants that gradually narrows under *All Invariants*, reaching $\sim 32\%$ on Base/Inv1 and ~ 28–29% on Inv2/Inv3 by epoch 5 (Base: 32.44, Inv1: 32.70, Inv2: 27.78, Inv3: 29.08). Under *Base and Inv1*, GRPO improves generalization to unseen variants more than SFT, reaching 19.67 (Inv2) and 20.88 (Inv3), while seen variants achieve 30.46 (Base) and 32.01 (Inv1). This corresponds to gains of 2.50 and 6.64 percentage points over SFT on the unseen variants, respectively. These results indicate that reward-driven optimization partially reduces dependence on surface form, enabling modest invariance transfer, but does not achieve strong generalization. Additional implementation details and full results are reported in § J.2.

Overall, both SFT and GRPO improve performance but are insufficient to close the invariance gap. This highlights the need for stronger mechanisms to induce true invariant reasoning.

6 Discussion

Our results show that current LLMs can often solve a reasoning problem in one representation but fail to preserve that ability under semantics-preserving reformulations. The large gap between at-least-one and all-four consistency, together with the prevalence of invariance-specific failures in stronger models, suggests that scaling improves problem solving more readily than invariant reasoning. Our SFT/GRPO study further shows that standard optimization can improve accuracy without removing sensitivity to surface form. These findings have implications for both evaluation and training: aggregate accuracy can overestimate reasoning robustness when equivalent formulations are not evaluated jointly, while training on observed reformulations can improve performance without establishing transfer to unseen invariants. For applications where inputs may vary in wording, structure, or presentation, such sensitivity can also make model behavior less predictable even when individual formulations are solved correctly.

These results highlight a gap in current reasoning training: models need mechanisms that connect equivalent representations to the same underlying reasoning state and preserve that state across transformations. Future work should explore transformation-aware objectives, representation-level consistency constraints, and structured state tracking, together with curricula that explicitly separate seen from unseen invariants.

7 Conclusion

In this work, we introduce InvariantBench, a benchmark and set of metrics for evaluating reasoning stability under semantics-preserving transformations. Our results show that fixed-benchmark accuracy can overestimate reasoning ability: models often degrade substantially across equivalent representations. The failure taxonomy further shows systematic sensitivity to how problems are expressed, while GRPO provides only partial gains. We hope InvariantBench establishes invariance as a core evaluation axis and motivates reasoning systems that generalize reliably across representations.

AI USE STATEMENT

In this work, we used generative AI tools for several research and writing tasks. We used an LLM-as-a-judge as part of the evaluation pipeline; its reliability was assessed through human–AI agreement analysis before using it for verification. In addition, we used generative AI tools for proofreading and improving the readability of the manuscript, debugging code, and formatting and styling tables and figures. After preparing first draft, we also used Stanford Agentic Reviewer, to obtain an initial review of the work, identify potential issues in the experiments and suggest related works for further consideration. We also used ICLR-supplied PAT review. All AI-assisted outputs were reviewed and verified by the authors. We take full responsibility for the final content of this work, including all claims, analyses, code, and artifacts.

REPRODUCIBILITY STATEMENT

All relevant implementation details, including hyperparameters, prompts, temperature settings, token limits, and evaluation protocols, are documented in the appropriate sections of the paper. These details are sufficient to reproduce our reported results on the benchmark. Upon acceptance, we will further release the complete codebase with full data and evaluation scripts.

REFERENCES

Anthropic. The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pd.

Anthropic. Introducing claude haiku 4.5, 2025. URL https://www.anthropic.com/news/claude-haiku-4-5.

Abhilekh Borah, Shubhra Ghosh, Kedar Joshi, Aditya Kumar Guru, and Kripabandhu Ghosh. Don't judge a book by its cover: Testing LLMs' robustness under logical obfuscation. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2162–2180, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.95. URL https://aclanthology.org/2026.eacl-long.95/.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. URL https://arxiv.org/abs/2303.12712.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

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, 2021. URL https://arxiv.org/abs/2110.14168.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker. Aya expanse: Combining research breakthroughs for a new multilingual frontier, 2024. URL https://arxiv.org/abs/2412.04261.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. ¡i¿did aristotle use a laptop?¡/i¿a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00370. URL http://dx.doi.org/10.1162/tacl_a_00370.

Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2Rwq6c3tvr.

Google. Gemma 4 model card, 2025. URL http://ai.google.dev/gemma/docs/core/model_card_4.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova,

Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle

Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. Putnam-AXIOM: A functional and static benchmark for measuring higher level mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=YXnwlZe0yf.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. FOLIO: Natural language reasoning with first-order logic. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1229. URL https://aclanthology.org/2024.emnlp-main.1229/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Rushang Karia, Daniel Bramblett, Daksh Dobhal, Pulkit Verma, and Siddharth Srivastava. ∀uto∃val: Autonomous assessment of llms in formal synthesis and interpretation tasks, 2024. URL https://arxiv.org/abs/2403.18327.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive

method on realistic data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SygcCnNKwr.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4110–4124, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.324. URL https://aclanthology.org/2021.naacl-main.324/.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9087–9105, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.731. URL https://aclanthology.org/2020.emnlp-main.731/.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, 2017. URL https://api.semanticscholar.org/CorpusID:46761158.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. GSM-plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2961–2984, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.163. URL https://aclanthology.org/2024.acl-long.163/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas

Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026. URL https://arxiv.org/abs/2601.08584.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning, 2020. URL https://arxiv.org/abs/2007.08124.

Gary Marcus. The next decade in ai: Four steps towards robust artificial intelligence, 2020. URL https://arxiv.org/abs/2002.06177.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, 2025. URL https://arxiv.org/abs/2410.05229.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress test evaluation for natural language inference. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle (eds.), *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2340–2353, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1198/.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in wonderland: Simple tasks showing complete reasoning breakdown in state-of-the-art large language models, 2025. URL https://arxiv.org/abs/2406.02061.

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Detecting generalization deficits in large language and reasoning models by using natural variations in simple problems. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=frA7uYn2um.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

OpenAI. Update to gpt-5 system card: Gpt-5.2, December 2025a. URL https://deploymentsafety.openai.com/gpt-5-2.

OpenAI. Openai o3 and o4-mini system card, April 2025b. URL https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty

Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning, 2024. URL https://arxiv.org/abs/2402.13950.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 3 edition, 2010. URL http://repo.darmajaya.ac.id/5272/1/Artificial%20Intelligence-A%20Modern%20Approach%20%283rd%20Edition%29%20%28%20PDFDrive%20%29.pdf. Accessed: 2026-04-30.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.722. URL https://aclanthology.org/2023.findings-emnlp.722/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Neha Srikanth, Marine Carpuat, and Rachel Rudinger. How often are errors in natural language reasoning due to paraphrastic variability? *Transactions of the Association for Computational Linguistics*, 12:1143–1162, 2024. doi: 10.1162/tacl_a_00692. URL https://aclanthology.org/2024.tacl-1.63/.

Aarohi Srivastava et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=uyTL5Bvosj. Featured Certification.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schölkopf, and Mrinmaya Sachan. A causal framework to quantify the robustness of mathematical reasoning with language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 545–561, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.32. URL https://aclanthology.org/2023.acl-long.32/.

Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:252917648.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Gemini Team. Gemini: A family of highly capable multimodal models, 2025. URL https://arxiv.org/abs/2312.11805.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne,

Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

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 *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models: a critical investigation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. In *Conference on Empirical Methods in Natural Language Processing*, 2019. URL https://api.semanticscholar.org/CorpusID:201698258.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022a. URL https://arxiv.org/abs/2206.07682.

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 *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022b. Curran Associates Inc. ISBN 9781713871088.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. Livebench: A challenging, contamination-limited llm benchmark, 2024. URL https://arxiv.org/abs/2406.19314.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. Benchmark data contamination of large language models: A survey, 2024. URL https://arxiv.org/abs/2406.04244.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger

Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Yang Wang, and Liangming Pan. How is LLM reasoning distracted by irrelevant context? an analysis using a controlled benchmark. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13329–13347, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.674. URL https://aclanthology.org/2025.emnlp-main.674/.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples, 2023. URL https://arxiv.org/abs/2311.04850.

Yue Zhou, Yada Zhu, Diego Antognini, Yoon Kim, and Yang Zhang. Paraphrase and solve: Exploring and exploiting the impact of surface form on mathematical reasoning in large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2793–2804, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.153. URL https://aclanthology.org/2024.naacl-long.153/.

Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F. Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=nDvgHIBRxQ.

Kaijie Zhu, Jiaao Chen, Jindong Wang, Neil Zhenqiang Gong, Diyi Yang, and Xing Xie. Dyval: Dynamic evaluation of large language models for reasoning tasks, 2024. URL https://arxiv.org/abs/2309.17167.

Terry Yue Zhuo, Zhuang Li, Yujin Huang, Fatemeh Shiri, Weiqing Wang, Gholamreza Haffari, and Yuan-Fang Li. On robustness of prompt-based semantic parsing with large pre-trained language model: An empirical study on codex. In Andreas Vlachos and Isabelle Augenstein (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1090–1102, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.77. URL https://aclanthology.org/2023.eacl-main.77/.

Appendix

Table of Content

A Related Work **22**

B Task Design and Descriptions **23**

B.1 Formal Logic & Discrete Structure (Category A) 25

B.2 Quantitative Reasoning & Optimization (Category B) 25

B.3 Relational, Planning, Algorithmic and Semantic Reasoning (Category C) 26

C Invariant Type Descriptions **26**

C.1 Symbolic and Linguistic Invariants 26

C.2 Structural and Representational Invariants 28

C.3 Abstraction and Information-Structure Invariants 28

D More Details on Benchmark Construction of INVARIANTBENCH **29**

D.1 Annotator Recruitment 29

D.2 Annotator Training and Protocol 30

D.3 Quality Control and Verification 30

E More details on Experiments and Evaluation **31**

E.1 Evaluation Protocol 31

E.2 Evaluation of LLM-as-a-Judge for Open-ended Evaluation 32

E.2.1 Cohen's Kappa 32

E.2.2 Accuracy against Majority Vote 33

E.2.3 Fleiss' Kappa (Human Reliability) 33

E.2.4 Stratified Evaluation across Task Structure 33

E.2.5 Human-Level Alignment Criterion 33

E.2.6 Interpretation 33

E.3 Human Baselines Development 34

F Metric Definitions and Interpretations **35**

F.1 Baseline Accuracy Metrics 35

F.2 ICP: Invariance Consistency Profile & IMI 35

F.3 CIR: Conditional Invariance Reliability 36

F.4 ICR: Invariance Collapse Rate 36

F.5 PIDI: Pairwise Invariance Dependency Index 36

F.6 TCIR: Type-Wise Conditional Invariance Retention 36

F.7 TCC: Type Collapse Contribution 37

F.8 RIS: Relative Invariance Sensitivity 37

F.9 BSS: Bilateral Symmetry Score . . . 37
F.10 CE: Consistency Entropy . . . 38
F.11 IRR: Invariant Recovery Rate . . . 38
F.12 FG: Fragility Gradient . . . 38
F.13 At-Least-k Correct Analysis . . . 39
F.14 Why these metrics are contamination-aware? . . . 39

G Additional Experiments and Results **40**
G.1 On Negative Invariance Gaps . . . 40
G.2 On Semantics-Preservation and Perturbation Axes . . . 41
G.3 Fragility Gradient and Non-Monotonic Difficulty . . . 41

H Full Per-Task and Per-Subtype Results **41**

I Error and Reasoning Failure Mode Analysis **42**
I.1 Setup and aggregate behavior . . . 42
I.2 E1. Premise-comprehension failure . . . 42
I.3 E2. Symbolic / logical translation failure . . . 43
I.4 E3. Computation & arithmetic failure . . . 44
I.5 E4. Multi-step reasoning / chain-coherence failure . . . 45
I.6 E5. Invariance brittleness . . . 45
I.7 E6. Lexical / semantic selection failure . . . 47
I.8 E7. Format / specification compliance failure . . . 47
I.9 E8. Generation / output pathology . . . 48
I.10 Overall Observations . . . 49
I.11 Root Cause Analysis and Mitigation Strategies . . . 49

J Details on Improving Invariant Reasoning Capabilities **51**
J.1 Supervised Fine-Tuning (SFT) . . . 51
J.2 Reinforcement Learning via GRPO . . . 51
J.3 On SFT/GRPO Diagnostic Scope . . . 52

K Discussion **53**
K.1 Overall Discussion . . . 53
K.2 Future Work Directions . . . 54
K.3 Broader Impacts . . . 55

L Representative Benchmark Samples with Model Answers and Evaluations **56**

A Related Work

LLM reasoning benchmarks. The dominant tradition treats reasoning as accuracy on a held-out test set of fixed instances. GSM8K (Cobbe et al., 2021) introduced 8.5K grade-school math word problems; MATH (Hendrycks et al., 2021b) extended this to 12.5K competition-level items; MMLU (Hendrycks et al., 2021a) aggregates 57 multiple-choice domains. BIG-Bench (Srivastava et al., 2023) contributed 204 tasks with broad topical coverage, distilled by Suzgun et al. (2022) into the 23-task BIG-Bench Hard subset that requires non-trivial multi-step reasoning. Domain-specific benchmarks include HumanEval (Chen et al., 2021) for code, StrategyQA (Geva et al., 2021) and ARC (Clark et al., 2018) for commonsense and science, LogiQA (Liu et al., 2020) and FOLIO (Han et al., 2024) for formal logic, and PlanBench (Valmeekam et al., 2023) for planning. All of these score a single fixed phrasing per instance; none probe whether the model would have answered the same way under a paraphrased, reordered, or otherwise semantics-preserving rewrite.

Robustness and surface-form sensitivity. A separate literature has documented that NLP systems are sensitive to surface form: Ribeiro et al. (2020) use minimum-pair perturbations to expose behavioural failures; Naik et al. (2018) stress-test NLI systems with antonym, length, and word-overlap perturbations; Wallace et al. (2019) show that universal adversarial triggers can flip predictions; and Zhuo et al. (2023) report that small prompt perturbations damage chain-of-thought accuracy. For reasoning specifically, Mirzadeh et al. (2025) produce GSM-Symbolic by templating 100 GSM8K items and varying the names and numerical values, finding that all models tested exhibit non-trivial performance variance and that adding a clause that "looks relevant" drops accuracy by up to 65%. Li et al. (2024) introduce GSM-Plus, an adversarial variant of GSM8K with eight perturbation types. Stolfo et al. (2023) present a causal framework for diagnosing arithmetic-reasoning robustness. InvariantBench differs from these in three respects: (i) it is not restricted to grade-school arithmetic; (ii) it covers twelve invariance axes spanning symbolic, structural, equivalence-based, and stylistic families; and (iii) every transformation is required to be *semantics-preserving* in a formal sense, so a degradation on a variant is unambiguously attributable to representation-dependence in the model rather than to a change in problem difficulty.

Faithfulness of chain-of-thought reasoning. Turpin et al. (2023) demonstrate that CoT explanations systematically misrepresent the actual reason for a prediction: biasing a multiple-choice prompt without altering the question can flip the answer while leaving the rationale apparently coherent. Lanham et al. (2023) measure CoT faithfulness by truncation, mistake-injection, and paraphrase, and find substantial unfaithfulness across model scales. Paul et al. (2024) survey the broader landscape. These results imply that even when a model produces a fluent chain that arrives at the correct answer, the chain may not be the mechanism by which the answer was derived, an interpretation that *predicts* brittleness under semantics-preserving rewrites and that our results are consistent with.

Contamination and dynamic evaluation. The risk that strong benchmark scores reflect memorisation of seen instances rather than transferable competence is now well documented (Xu et al., 2024; Sainz et al., 2023; Yang et al., 2023; Golchin & Surdeanu, 2024). Dynamic benchmarks respond by regenerating items at evaluation time (Kiela et al., 2021; Zhu et al., 2024; White et al., 2024; Gulati et al., 2024). InvariantBench contributes to this line by treating each invariant variant as a contamination-resistant probe of the same underlying problem. Performance on the invariant variants provides a lower bound on reasoning beyond instance-specific memorization, while the gap between base-item and variant performance quantifies the extent to which base performance may be inflated by surface-form memorization. CheckList (Ribeiro et al., 2020) introduced Invariance Tests as one of three behavioral test types (Minimum Functionality, Invariance, Directional Expectation) for verifying that model predictions remain unchanged under label-preserving input perturbations, organized around a taxonomy of linguistic capabilities. Our invariance measure differs in scope (general multi-step reasoning problems vs. linguistic capability unit tests) and in using four distinct representation types (symbolic, structural, and two abstraction axes) rather than a fixed perturbation template library.

Robustness to Irrelevant Context. Prior work shows that LLM reasoning is sensitive to context that is irrelevant to the underlying problem. Shi et al. (2023) showed that appending irrelevant sentences to grade-school math problems can substantially reduce accuracy. Extending this direction

beyond arithmetic, Yang et al. (2025b) introduced GSM-DC, a controlled benchmark that injects irrelevant context into reasoning problems and found that such distractors can alter both reasoning paths and arithmetic accuracy. They further showed that targeted training with strong distractors improves robustness in both in- and out-of-distribution settings. Our 277-sample *Distractor Injection* task type builds on this line but evaluates distraction across a broader range of reasoning domains and jointly compares it with three other forms of representation change.

Reasoning Under Structure-Preserving Transformations. A complementary line of work studies whether LLM reasoning is stable when the underlying problem structure is preserved. Nezhurina et al. (2025) found dramatic performance swings under natural, structure-preserving variations of a single problem template, with accuracy ranging from near 0% to near 100%. A subsequent TMLR 2026 extension (Nezhurina et al., 2026) examined this phenomenon across a broader set of simple problems. Zhou et al. (2024) measure how paraphrases change answer distributions and solve rates in mathematical reasoning, and propose self-consistency over paraphrases as a mitigation. Srikanth et al. (2024) quantify paraphrastic consistency in natural-language reasoning using human-written, validated reformulations. Zhou et al. (2025) introduce a broad checklist for mathematical reasoning that includes answer-preserving rephrasing and irrelevant information conditions. These studies establish that accuracy alone is an incomplete measure of reasoning robustness. More recently, Borah et al. (2026) introduced Logifus, a structure-preserving logical-obfuscation framework, together with the 1,108-question LogiQAte benchmark. Across six models, they observed substantial degradation under logical obfuscation, but did not study training-time mitigation. In contrast, our benchmark spans multiple reasoning domains and multiple invariance types, allowing these failure modes to be compared within a common evaluation framework. We further test whether the resulting invariance gap can be reduced through supervised fine-tuning (SFT) and GRPO (Section J).

Compositional and systematic generalization. A long line of work, beginning with Lake and Baroni (Lake & Baroni, 2017) and continued by Keysers *et al.*, and Kim and Linzen (Keysers et al., 2020; Kim & Linzen, 2020), asks whether neural models can generalise compositionally, for example, by handling novel combinations of seen primitives. INVARIANTBENCH sits adjacent to this literature but probes a different property: not generalisation across compositional recombinations of primitives, but *invariance under recoding of the same problem*. The two properties are related but distinct: a model can in principle be compositionally generalising yet representation-fragile, or representation-invariant yet non-compositional. Our experiments target the first axis directly.

Logical and symbolic evaluation. For first-order and modal logic, FOLIO (Han et al., 2024) and PROOFWRITER (Tafjord et al., 2021) provide formal deduction items; LOGIQA (Liu et al., 2020) provides multiple-choice logical items in natural language. The ∀uto∃val framework (Karia et al., 2024) synthesises items from formal grammars. None of these explicitly enforces invariance across surface forms of the same logical content. INVARIANTBENCH's *equivalent logical reformulation* axis (e.g. contrapositive, De Morgan's, prenex normal form) and *variable renaming* axis together enable the precise version of the question *"does the model solve the proposition or recognise the string?"*

Position of INVARIANTBENCH. INVARIANTBENCH unifies four threads of prior work, robustness testing (Ribeiro et al., 2020; Mirzadeh et al., 2025), contamination-resistant evaluation (Kiela et al., 2021; Zhu et al., 2024), faithfulness analysis (Turpin et al., 2023; Lanham et al., 2023), and reasoning-domain coverage (Srivastava et al., 2023; Suzgun et al., 2022; Han et al., 2024), into a single framework whose primitive is the (seed, variant) pair. The result is a benchmark that scores a model not on whether it answered a particular instance, but on whether *the answer is the same across every semantics-preserving variant of that instance*.

B TASK DESIGN AND DESCRIPTIONS

INVARIANTBENCH partitions its 1159 evaluation instances across three high-level task families, Formal Logic & Discrete Structure (A), Quantitative Reasoning & Optimization (B), and Relational, Planning, Algorithmic and Semantic Reasoning (C), as shown in Table 5, each decomposed into fine-grained sub-types that target distinct competency dimensions of language model reasoning, as per Table 6.

Table 5: **Distribution of task types and sub-types in the benchmark.** Counts indicate the number of samples in each category.

Category	Task Type / Sub-Type	Count
Task Type		
A	Formal Logic & Discrete Structure	391
B	Quantitative Reasoning & Optimization	369
C	Relational, Planning, Algorithmic & Semantic Reasoning	399
Task Sub-Type		
A1	Multi-Quantifier First-Order Logic	91
A2	Set Algebra + Cardinality with Constraints	84
A3	Conditional Override Transformation Systems	59
A4	Constraint Satisfaction in Propositional Logic	65
A5	Inductive Rule + Formal Deduction	47
A6	Unit/Representation Invariance in Physics-Style Word Problems	45
B1	Sequential Algebraic Systems with Layered Constraints	73
B2	Optimization over Coupled Algebraic Equations	76
B3	Probabilistic Reasoning with Conditional Dependence	62
B4	Interdependent Equation Systems with Structural Constraints	79
B5	Constraint-Driven Algebraic Reasoning Problems	79
C1	Algebra, Pseudocode, and Algorithmic Reasoning	67
C2	Graph Search and Traversal Reasoning	104
C3	Path-Based Search and Multi-Agent Interaction Reasoning	96
C4	Graph Isomorphism and Structural Mapping	96
C5	Language, Semantics & Information Structure-based Reasoning	36
Total		1159

Table 6: **Distribution of invariant types across the benchmark.** Counts indicate the number of samples per transformation category.

Invariant Family	Transformation Type	Count
Symbolic & Linguistic	Entity Relabeling	295
	Style / Register Invariance	273
	Synonym / Semantic Paraphrase Invariance	251
	Variable Renaming	340
Structural & Representational	Equivalent Logical Reformulation	398
	Equivalent Mathematical Reformulation	233
	Premise Reordering	311
	Representation / Encoding Swap	217
Abstraction & Information-Structure	Abstraction Level Shift	318
	Distractor Injection / Removal	277
	Perspective / Role Reframing	217
	Redundant Restatement Invariance	347

B.1 Formal Logic & Discrete Structure (Category A)

Category A contains 393 instances that test symbolic reasoning over logic, sets, constraints, and structured representations. The six subtypes examine whether models preserve correct deductions under surface-level reformulations.

♦ **A1. Multi-Quantifier First-Order Logic (91 instances).** A1 requires reasoning over formulas with two or more quantifier alternations (∀/∃), with careful handling of variable scope, substitution, and universal versus existential claims. The tasks distinguish logically different forms such as $\forall x, \exists y$ and $\exists y, \forall x$, making shallow pattern matching insufficient. We test whether conclusions remain invariant under premise reordering and quantifier-preserving reformulations, probing whether models track quantifier semantics rather than surface form.

♦ **A2. Set Algebra + Cardinality with Constraints (84 instances).** A2 combines set operations such as intersections, unions, complements, and power sets with cardinality constraints across multiple regions. Solving the tasks requires reasoning about implicit set partitions, exclusive cases, and numerical bounds. We test whether this reasoning remains stable under notation changes, such as $|A \cap B|$ versus "the number of elements common to A and B," rather than relying on familiar surface patterns.

♦ **A3. Conditional Override Transformation Systems (59 instances).** A3 uses small rule systems in which priority rules can override a base transformation when their conditions are met. Models must identify applicable rules, resolve conflicts using the given priority order, and apply all relevant overrides. The tasks test whether models maintain a consistent state while reasoning through layered, exception-based rules.

♦ **A4. Constraint Satisfaction in Propositional Logic (65 instances).** A4 presents Boolean constraint problems using different representations, including CNF, natural-language conditionals, truth tables, and scenarios. Solving them requires case analysis, unit propagation, or backtracking over small but non-trivial solution spaces. We test whether models reach the same satisfying assignment, or correctly identify unsatisfiability, across equivalent representations.

♦ **A5. Inductive Rule + Formal Deduction (47 instances).** A5 provides base facts and inductive rules, then asks models to determine which facts follow after a bounded number of rule applications. The tasks require forward chaining and recognition of fixed-point behavior, while distinguishing derivable facts from those that remain unknown. This tests whether models can reliably simulate a small recursive inference system.

♦ **A6. Unit/Representation Invariance in Physics-Style Word Problems (45 instances).** A6 uses elementary physics and engineering problems involving motion, energy, pressure, and rates, expressed under different unit systems or measurement conventions. Models must apply the appropriate conversions before solving the problem. The same physical scenario should yield the same answer after conversion, allowing us to test whether reasoning is invariant to equivalent numerical representations.

B.2 Quantitative Reasoning & Optimization (Category B)

Category B contains 399 instances covering algebraic, probabilistic, and optimization reasoning with multiple interacting quantities. The five subtypes test whether quantitative reasoning remains stable when mathematically equivalent formulations are presented differently.

♦ **B1. Sequential Algebraic Systems with Layered Constraints (73 instances).** B1 presents multi-step algebraic problems where each constraint depends on previous results. Models must follow the dependency chain and maintain exact intermediate values, since early errors propagate to later steps. We test whether outputs remain consistent when independent constraints are reordered.

♦ **B2. Optimization over Coupled Algebraic Equations (76 instances).** B2 asks models to optimize an objective over several variables linked by algebraic constraints. Solutions may require substitution, optimality conditions, or boundary analysis rather than brute-force search. We test whether models correctly identify active constraints and optimal solutions when the same constraints are algebraically rearranged.

♦ **B3. Probabilistic Reasoning with Conditional Dependence (62 instances).** B3 tests probability reasoning in settings with conditional dependencies such as chains, forks, and colliders. Models must determine when independence assumptions are valid, apply the chain rule or Bayes' rule correctly, and update probabilities across multiple steps. We test whether results remain unchanged under equivalent descriptions of the same joint distribution.

♦ **B4. Interdependent Equation Systems with Structural Constraints (79 instances).** B4 combines nonlinear or piecewise equations with structural constraints such as integrality, non-negativity, or ordering. Models must identify valid algebraic solutions and filter out those that violate the constraints. We test whether the feasible set is preserved when these constraints are expressed in logically equivalent forms.

♦ **B5. Constraint-Driven Algebraic Reasoning Problems (79 instances).** B5 makes domain constraints—such as ranges, modular conditions, and divisibility—central to the solution. Models must reason from these constraints to identify feasible assignments or determine which expressions are fixed or bounded. This tests constraint-first reasoning rather than standard equation solving.

B.3 Relational, Planning, Algorithmic and Semantic Reasoning (Category C)

Category C contains 408 instances covering graph reasoning, multi-agent planning, algorithmic execution, and language semantics. The five subtypes test whether models preserve structural reasoning under equivalent surface reformulations.

♦ **C1. Algebra, Pseudocode, and Algorithmic Reasoning (67 instances).** C1 combines algebra with pseudocode execution, requiring models to trace loops, conditionals, updates, and intermediate states to obtain the final output. We test whether results remain stable under semantics-preserving changes such as variable renaming, limited loop unrolling, or equivalent recursive-to-iterative rewrites.

♦ **C2. Graph Search and Traversal Reasoning (104 instances).** C2 asks models to solve graph-search tasks involving reachability, shortest paths, connected components, topological ordering, and cycle detection. Graphs are represented as adjacency lists, edge sets, or natural-language descriptions. We test whether the same structural answer is obtained under node/edge relabeling and equivalent graph representations.

♦ **C3. Path-Based Search and Multi-Agent Interaction Reasoning (96 instances).** C3 extends graph reasoning to multiple agents, resource constraints, and competitive or cooperative interactions. Tasks include constrained routing, pursuit-evasion, synchronized navigation, and turn-based scenarios. Solving them requires reasoning over joint states and trajectories. We test whether solutions remain stable under equivalent agent relabeling and graph restructuring.

♦ **C4. Graph Isomorphism and Structural Mapping (96 instances).** C4 asks models to determine whether two graphs are isomorphic, construct a valid vertex mapping, or identify a largest common substructure. Solving these tasks requires matching structural properties and eliminating inconsistent candidate mappings. We test whether isomorphism judgments remain unchanged when vertex labels are permuted.

♦ **C5. Language, Semantics & Information Structure-based Reasoning (36 instances).** C5 covers coreference, presupposition, implicature, information structure, and discourse-level entailment. The tasks require tracking meaning-relevant distinctions that may not be visible from surface syntax. We test whether semantic conclusions remain stable under meaning-preserving paraphrases and register changes while changing appropriately when syntax alters meaning.

C Invariant Type Descriptions

InvariantBench evaluates model robustness via twelve invariant transformation types organized into three families: Symbolic and Linguistic, Structural and Representational, and Abstraction and Information-Structure, as shown in Table 7. Each transformation constructs a semantically equivalent problem variant; a robust model should produce the same answer on both the original and transformed instances, whereas a fragile model reveals its reliance on surface features that change under the transformation.

C.1 Symbolic and Linguistic Invariants

Symbolic and linguistic invariants modify the *surface vocabulary* of a problem, the names of entities, the phrasing of conditions, the syntactic register of the question, while leaving the underlying logical, mathematical, or relational content entirely unchanged. These four invariant types collectively contribute 1159 instances and probe whether model performance is driven by semantic understanding or by lexical and symbolic surface cues.

♦ **Entity Relabeling (295 instances).** Entity Relabeling replaces the named participants in a problem

Table 7: The twelve invariance axes in INVARIANTBENCH.

Family	Transformation τ	Operation
Symbolic & Linguistic	Variable renaming	Injective renaming of free and bound variables.
	Entity relabeling	Consistent relabeling of named entities and predicates.
	Synonym / paraphrase	Lexical substitution and sentence-level paraphrasing.
	Style / register	Register shifts without altering semantic content.
Structural & Representational	Premise reordering	Permutation of independent clauses under dependency constraints.
	Redundant restatement	Addition of paraphrased clauses without new information.
	Distractor injection / removal	Addition or removal of semantically irrelevant clauses.
	Representation swap	Conversion across equivalent encodings (text, symbolic, tabular).
Abstraction & Information Structure	Equivalent logical reformulation	Sound logical equivalence transformations.
	Equivalent mathematical reformulation	Algebraic and arithmetic equivalences.
	Abstraction level shift	Change in granularity while preserving task semantics.
	Perspective / role reframing	Reassignment of roles or viewpoints without altering relations.

(persons, cities, objects, organizations) with semantically neutral alternatives, typically unfamiliar proper nouns or placeholder identifiers, so that any answer derived from world-knowledge associations about the original entities would be disrupted. The transformation preserves the complete logical and relational structure of the problem while eliminating the possibility of bypassing reasoning by recalling facts about the specific entities named. This invariant is especially diagnostic in relational and planning tasks (Category C) where familiar entity names (canonical graph nodes, well-known route networks) may trigger memorized traversal patterns. Correct invariance here provides evidence that a model is processing the relational structure of the problem rather than pattern-completing from training-distribution entity co-occurrences.

♦ **Style / Register Invariance (273 instances).** Style and Register Invariance rewrites the problem statement in a different communicative register, formal academic prose to informal colloquial phrasing, passive to active voice, second-person to third-person narration, or telegraphic notation to full prose, without altering any factual content, logical connectives, or quantitative values. The transformation targets the well-documented sensitivity of language models to stylistic surface variation, which can cause identical questions to receive different answers when phrased in unfamiliar registers. This invariant is particularly relevant to deployment robustness: models used in consumer-facing settings must produce consistent answers across the broad stylistic variation of user inputs, and failures on this invariant reveal that internal representations conflate style with substance. Correct invariance demonstrates that the model's reasoning is anchored to propositional content rather than surface fluency patterns.

♦ **Synonym / Semantic Paraphrase Invariance (251 instances).** Synonym and Semantic Paraphrase Invariance substitutes key content words and phrases with semantically equivalent alternatives, synonyms, definitional expansions, or well-known paraphrases, that preserve truth conditions while changing lexical form. Unlike register variation, which affects global stylistic register, this transformation targets local lexical substitutions that individually preserve meaning but collectively produce a surface form distant from the original, testing whether the model's internal semantic representation is robust to vocabulary diversity. The transformation is constructed to avoid pragmatic shifts: substitutions are selected so that the paraphrase is truth-conditionally equivalent in context, ruling out cases where synonymy fails under specific presuppositions. This invariant most directly probes lexical-semantic grounding, whether concepts are represented in terms of their meanings or their most frequent surface lexicalizations.

♦ **Variable Renaming (340 instances).** Variable Renaming substitutes all symbolic identifiers in a formal problem, mathematical variable names, logical proposition symbols, set labels, graph node names, with fresh symbols drawn from a different alphabet or naming convention, while maintaining all structural relationships among identifiers identically. This transformation directly tests whether model reasoning over symbolic expressions is anchored to the algebraic/logical structure or to the specific symbols used, a distinction that is definitional for genuine formal reasoning: a sound algebraic solver should be entirely indifferent to whether the unknown is called x, z, α, or `myVar`. Variable Renaming is the most frequently applied invariant type (351 instances), reflecting its fundamental importance across all three task categories and its tight connection to the notion of syntactic invariance in formal systems. Failures, where renaming a variable causes a model to

solve a different equation or draw a different logical conclusion, constitute unambiguous evidence of surface-level symbol-matching rather than formal symbolic reasoning.

C.2 Structural and Representational Invariants

Structural and representational invariants modify the *organization and encoding* of problem information, the logical form, mathematical notation, premise order, or representational modality, while preserving semantic content. These four invariant types probe the depth of structural comprehension: a model that has internalized the mathematical or logical structure of a problem should be unaffected by surface restructuring, while a model exploiting positional or notational regularities will exhibit systematic failures.

♦ **Equivalent Logical Reformulation (398 instances).** Equivalent Logical Reformulation rewrites the logical structure of a problem using a known equivalence transformation, replacing $P \Rightarrow Q$ with $\neg Q \Rightarrow \neg P$ (contrapositive), distributing quantifiers, applying De Morgan's laws, converting between disjunctive and conjunctive normal forms, or replacing biconditionals with conjunctions of conditionals, producing a problem whose propositional content is provably identical. This is the most frequently applied invariant (408 instances, matching Category C count), reflecting the ubiquity of equivalent logical reformulation as a natural source of surface variation in reasoning tasks. Correct invariance requires that the model recognize the semantic equivalence of the reformulated structure, a competency that is foundational to logical reasoning and formal argumentation.

♦ **Equivalent Mathematical Reformulation (233 instances).** Equivalent Mathematical Reformulation transforms the algebraic or arithmetic structure of quantitative problem conditions using known mathematical identities, distributing products, factoring sums, applying commutativity and associativity, rewriting fractions as decimals, or converting between parametric and explicit representations, so that all quantities and relationships are preserved but their syntactic form changes. This invariant isolates the algebraic layer of mathematical understanding from the arithmetic execution layer: a model that computes correctly on one form but fails on an equivalent rearrangement has demonstrated shallow algebraic competence grounded in lexical form recognition rather than structural mathematical understanding.

♦ **Premise Reordering (311 instances).** Premise Reordering permutes the order in which problem conditions are stated, placing the goal before the constraints, reversing the order of case definitions, interleaving conditions that were originally grouped by type, without adding or removing any information. This invariant directly tests positional sensitivity: models that construct reasoning chains by attending locally to adjacent sentences or that leverage positional priors ("earlier premises are more important") will exhibit degraded performance under arbitrary reorderings that a logically competent reasoner would be indifferent to. Premise Reordering is the second most common structural invariant (321 instances), reflecting the natural variation in premise presentation order across problem formulations in the wild.

♦ **Representation / Encoding Swap (217 instances).** Representation and Encoding Swap translates the entire problem from one representational modality to an equivalent one, graph adjacency list to adjacency matrix, set notation to characteristic-function description, truth table to Boolean formula, or decimal numbers to binary/hexadecimal, while preserving all structural information. This invariant targets the representational layer of understanding: a model that has genuinely internalized the abstract structure being described should be able to process it regardless of the particular encoding, while a model that has learned to process specific surface encodings will fail when the same information is encoded differently. Encoding Swap tests are especially powerful diagnostics for graph reasoning (C2, C3, C4) and propositional logic (A4) tasks, where multiple standard encodings co-exist and competent human solvers routinely translate between them.

C.3 Abstraction and Information-Structure Invariants

Abstraction and information-structure invariants modify the *level of detail, perspective, or information density* at which a problem is presented, testing whether models can retrieve the correct answer when surface information is reorganized, augmented with irrelevant material, or reframed from a different viewpoint. These invariants probe the robustness of reasoning to the kind of contextual and framing variation that characterizes real-world language use.

♦ **Abstraction Level Shift (318 instances).** Abstraction Level Shift rewrites the problem at a higher or lower level of conceptual granularity, replacing concrete numerical instantiations with parameterized variables (upward abstraction), or replacing abstract structural descriptions with fully

concrete worked examples (downward instantiation), while preserving the specific answer required. This transformation tests whether a model's reasoning capability is calibrated to a specific level of abstraction in the training distribution or is genuinely level-agnostic; models overfit to, say, concrete arithmetic problems will fail on their algebraically generalized counterparts. The upward variant is particularly relevant to evaluating algebraic generalization: can a model that correctly solves "if $x = 3$, compute $2x + 1$" also correctly evaluate the general expression $2x + 1$ for a symbolically stated x? Abstraction Level Shift is the third most common invariant (329 instances), reflecting its broad applicability across all task categories.

♦ **Distractor Injection / Removal (277 instances).** Distractor Injection and Removal adds semantically irrelevant but topically plausible information to a problem (injection) or strips away such distractors that were present in the original (removal), testing whether model performance degrades in the presence of irrelevant context or improves spuriously when distractors are removed. Injected distractors are constructed to be relevant in surface form, they mention entities and quantities that appear in the problem, but to be logically irrelevant to the correct solution, so that a model attending to all mentioned quantities will be misled while a focused reasoner will correctly ignore them. This invariant is closely related to the classic comprehension construct of "reading to find relevant information" and is diagnostic of selective attention and relevance filtering, two capabilities critical for robust performance in long-context reasoning. Removal of distractors from the original problem tests the complementary direction: whether the original problem's answer is recovered when the confounding material is excised.

♦ **Perspective / Role Reframing (217 instances).** Perspective and Role Reframing restates the problem from the vantage point of a different participant, switching from a first-person agent to an observer, from a questioner to a responder, from a buyer to a seller in a transaction scenario, while leaving the underlying facts and relational structure unchanged. This invariant tests whether a model's reasoning is anchored to the objective structure of the problem or to the particular role assignment used to introduce it, a failure mode that would manifest as role-specific answer biases without corresponding changes in the correct answer. Perspective Reframing is particularly diagnostic for multi-agent planning tasks (C3) and relational reasoning tasks (C2, C4) where role assignments are explicit and the same graph-structural or logical content is accessed from different positions in the interaction. Correct invariance on this transformation provides evidence for genuinely objective structural reasoning rather than role-anchored heuristic responses.

♦ **Redundant Restatement Invariance (347 instances).** Redundant Restatement Invariance augments the problem with additional sentences that restate already-given information in different words, paraphrasing a key constraint, re-expressing a given quantity using equivalent arithmetic, or confirming an already-stated relationship, without adding any new information or contradicting existing content. This invariant tests robustness to verbosity: a logically competent model should recognize the restatements as redundant and neither weight them as new evidence nor become confused by the apparent repetition, whereas a model that tracks information by occurrence frequency may spuriously amplify the importance of redundantly stated facts. Redundant Restatement is the most frequent abstraction-layer invariant (358 instances), reflecting both the ease of generating such variants and their diagnostic value for probing information-redundancy sensitivity. Correct invariance on this transformation is a necessary condition for reliable reasoning over real-world texts, which routinely contain paraphrastic repetition for rhetorical and communicative purposes.

D More Details on Benchmark Construction of InvariantBench

D.1 Annotator Recruitment

We recruited annotators aged 24–30, all with engineering backgrounds and prior exposure to logic- or reasoning-oriented tasks. Participation was voluntary, and annotators worked independently without external incentives to reduce bias induced by reward-driven labeling. Before annotation began, we conducted structured onboarding sessions covering task objectives, invariance principles, and quality expectations. Ethical considerations and data handling protocols were strictly enforced throughout the process. This ensured that all annotations were produced under consistent and controlled conditions aligned with the benchmark design.

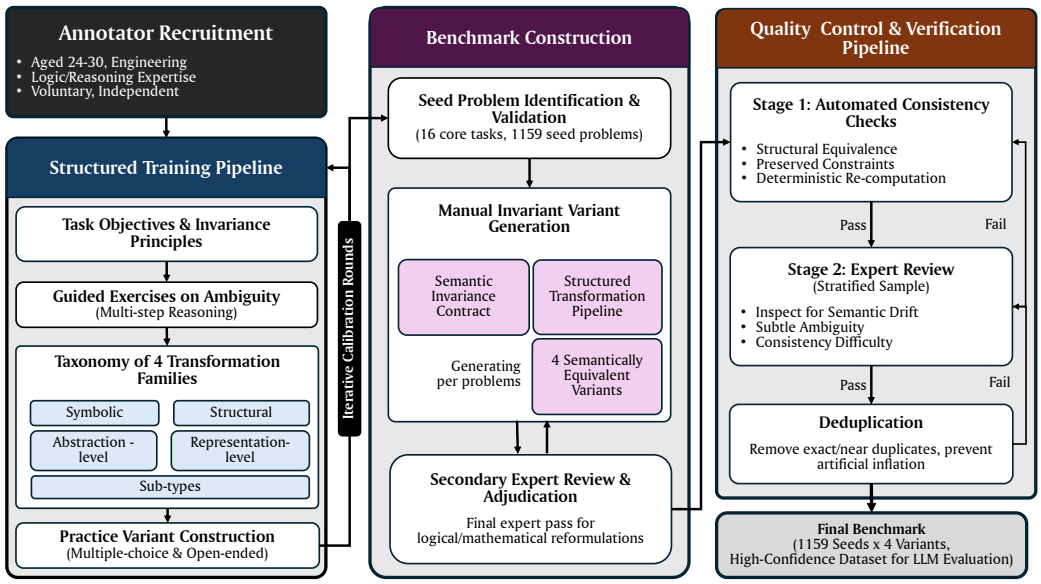

Figure 4: **Overview of the INVARIANTBENCH construction and verification pipeline.** Left: Domain-expert annotators undergo a structured training protocol covering the four transformation families, semantic invariance contracts, and edge-case resolution. Center: From 1,159 validated seed problems across 16 core reasoning tasks, annotators generate four semantically equivalent surface forms per problem under secondary expert review and iterative calibration. Right: A two-stage quality assurance pipeline applies automated consistency checks, expert manual inspection for semantic drift, and de-duplication to produce the final high-confidence evaluation benchmark.

D.2 ANNOTATOR TRAINING AND PROTOCOL

We developed a structured training pipeline to ensure consistency across the full invariance taxonomy. Annotators were first introduced to the three reasoning families and their associated task structures, with emphasis on how semantic equivalence must be preserved under surface transformations. For each invariance type, we provided curated examples and counter-examples illustrating valid and invalid transformations, including cases where subtle semantic drift could occur.

Training further emphasized the four transformation families (symbolic, structural, abstraction-level, and representation-based invariances) and their fine-grained subtypes, ensuring annotators could reliably distinguish permissible reformulations from those that alter underlying semantics. We also trained annotators to explicitly verify equivalence under the invariance contract, particularly for logical, mathematical, and compositional transformations.

To ensure robustness across visual and textual variability, we incorporated guided exercises on handling ambiguous cases, including multi-step reasoning chains, compositional transformations, and distractor-rich formulations. Annotators practiced constructing both multiple-choice and open-ended variants, with emphasis on preserving answer invariance while varying surface structure.

D.3 QUALITY CONTROL AND VERIFICATION

To ensure correctness and invariance integrity, each problem–variant set is validated through a two-stage verification pipeline. The first stage consists of consistency checks that enforce structural equivalence across variants and identify invalid transformations. These checks ensure that problem constraints are preserved, that no unintended assumptions are introduced or removed, and that symbolic or structural modifications do not alter the underlying solution space. Where applicable, deterministic recomputation is used to confirm answer consistency across transformed instances. Any instance that violates these constraints is rejected or returned for regeneration.

In the second stage, three independent expert reviewers with training in logic, mathematics, and formal reasoning evaluated each sampled item. For primary review, each reviewer assigned one

of three labels: *accept as is*, *revision needed*, or *reject*. Inter-annotator agreement was substantial ($\kappa = 0.78$), and we used majority voting to determine the initial decision. For items marked as requiring revision, reviewers provided specific feedback on semantic drift, ambiguity, transformation fidelity, and difficulty consistency. We revised these items based on the feedback and subjected them to a final acceptance check. An item was accepted only when all three reviewers agreed that the revised version fully satisfied the transformation requirements, yielding unanimous agreement on final acceptance.

Only instances that satisfy both automated validation and expert review criteria are retained in the final benchmark. We additionally perform deduplication to remove exact and near-duplicate instances across the benchmark, including semantically overlapping question pairs. This prevents repeated or highly similar items from artificially inflating model performance and ensures that evaluation results reflect distinct reasoning instances. This dual-layer verification and deduplication process produces a high-confidence dataset of invariantly paired reasoning tasks designed to enable controlled evaluation of representation sensitivity in modern large language models.

Finally, after these iterative calibration rounds, final dataset sample is generated. This iterative process ensured high-fidelity alignment between intended invariance transformations and final benchmark instances.

E More details on Experiments and Evaluation

E.1 Evaluation Protocol

We evaluate all models using a controlled prompting and automated judging pipeline to ensure consistency and comparability across variants.

Answer Generation. Each model is prompted with a strict output constraint to minimize verbosity and reduce format-induced variance.

Prompt for Answering Model

Answer the following question as concisely as possible, using short sentences. Give only the answer itself. Don't add any unnecessary information.
Question: `<question>`

System Prompt for Judge Model

You are a strict, impartial answer evaluator.
You will be given:
- A question
- The correct answer
- A model's response

Your job: determine whether the model's response is CORRECT or INCORRECT.
Rules:
- Match to given correct answer, instead of reasoning.
- Be flexible about formatting.
- Be strict about semantic content — a vague or partially correct answer is INCORRECT.
- Ignore extra whitespace, punctuation, capitalisation differences.
- If the model's response contains the correct answer within a sentence, mark it CORRECT.
- If the model says "I don't know" or gives an irrelevant answer, mark it INCORRECT.
- Any ambiguous answer is INCORRECT.

Respond with ONLY a JSON object in this exact format (no markdown fences, no extra text):
"score": 1, "reason": "one sentence (7-10 words) explaining the decision"
or
"score": 0, "reason": "one sentence (7-10 words) explaining the decision"

Base Prompt for Judge Model

Question:
¡question¿
Correct Answer:
¡correct_answer¿
Model Response:
¡model_response¿

Judging Protocol. All responses are evaluated using a separate judge model under a strict binary scoring scheme. The judge compares the model output directly against the ground-truth answer, allowing minor formatting differences (e.g., whitespace, capitalization), but rejecting semantically incomplete or approximate answers. If the correct answer appears within a longer response, it is accepted; otherwise, any ambiguity results in a score of 0. The judge returns a structured JSON object containing a binary score and a brief justification.

Decoding and Inference Settings. We set `temperature` to 0.1 to reduce stochasticity and ensure stable outputs across variants. The maximum generation length is set to 600 tokens for standard models and 20,000 tokens for reasoning-augmented models, ensuring sufficient capacity for intermediate reasoning when required. The judge model is allocated a maximum of 1,200 tokens. All evaluations are conducted via the OPENROUTER API with default parameters otherwise unchanged, with API usage costs ranging from approximately $0.10 to $15 per 1M input/output tokens; the full set of experiments, including initial testing and ablations, incurred a total cost of approximately $550.

Rationale. This evaluation design isolates correctness from verbosity and stylistic variation, ensuring that measured differences reflect reasoning performance rather than output formatting. The strict judge further enforces consistency across models, enabling reliable comparison under invariant transformations.

E.2 EVALUATION OF LLM-AS-A-JUDGE FOR OPEN-ENDED EVALUATION

To evaluate *Gemma-4-26B-A4B-Instruct* as an automatic judge, we selected a balanced set of 240 samples spanning all task categories and sub-types. Each sample consists of a question, ground truth, and model-generated output. Three independent human annotators evaluated each response, assigning binary labels: $\{1 = \text{correct}, 0 = \text{incorrect}\}$.

To obtain a robust human reference signal, we construct a **majority-vote label** for each sample:

$$y_i^{\text{maj}} = \text{mode}\big(y_i^{(1)}, y_i^{(2)}, y_i^{(3)}\big), \tag{3}$$

which serves as the primary proxy for ground-truth judgment in all comparisons with the LLM judge.

We evaluate the LLM judge against human annotations using multiple complementary agreement metrics, computed both globally and stratified by *Task Type* and *Task Sub-Type*. This enables us to assess not only overall alignment with human judgments but also the stability of the judge across different evaluation regimes.

E.2.1 COHEN'S KAPPA

Cohen's kappa (κ) measures chance-corrected agreement between two raters. It is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{4}$$

where p_o denotes the observed agreement and p_e is the expected agreement under independence:

$$p_e = \sum_{c \in \{0,1\}} p_c^{(1)} p_c^{(2)}. \tag{5}$$

We compute κ between the LLM judge and each individual annotator, as well as between the LLM and the majority vote label:

$$\kappa_{\text{LLM,maj}}. \tag{6}$$

This formulation allows direct comparison of the LLM judge against both individual human variability and aggregated human consensus.

E.2.2 Accuracy against Majority Vote

We further report raw agreement with the majority label:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\big(y_i^{\text{LLM}} = y_i^{\text{maj}}\big), \tag{7}$$

which captures direct alignment with human consensus but does not correct for chance agreement effects.

E.2.3 Fleiss' Kappa (Human Reliability)

To quantify inter-annotator reliability, we compute Fleiss' kappa (κ_F), defined as:

$$P_i = \frac{1}{n(n-1)}\sum_j n_{ij}(n_{ij}-1), \quad \bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i, \tag{8}$$

$$p_j = \frac{1}{Nn}\sum_i n_{ij}, \quad \bar{P}_e = \sum_j p_j^2, \tag{9}$$

$$\kappa_F = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}. \tag{10}$$

This serves as an empirical *human agreement ceiling*, against which the LLM judge is compared.

E.2.4 Stratified Evaluation across Task Structure

Beyond aggregate performance, we perform a fine-grained analysis conditioned on *Task Type* and *Task Sub-Type*. For each subset $\mathcal{D}_t$, we compute:

$$\text{Accuracy}_t, \quad \kappa_t^{\text{LLM vs maj}}, \quad \kappa_t^{\text{human}} \tag{11}$$

where κ_t^{human} denotes the mean pairwise Cohen's kappa among human annotators within the subset. This allows us to measure whether the LLM judge achieves *human-level agreement locally within specific reasoning regimes*, rather than only in aggregate.

E.2.5 Human-Level Alignment Criterion

We define the LLM judge as achieving human-level performance if:

$$\kappa_{\text{LLM, maj}} \approx \kappa_{\text{human}}, \tag{12}$$

and if its accuracy lies within the empirical confidence interval of human inter-annotator agreement. We additionally report bootstrap confidence intervals over accuracy and conduct permutation tests to assess whether observed agreement significantly exceeds random alignment.

Across all evaluations, results are reported both globally and disaggregated by task structure, providing a comprehensive view of the reliability and calibration of LLM-based automatic evaluation.

E.2.6 Interpretation

Human Validation of the LLM Judge. Across all 720 validation responses, the LLM judge exactly matched the human validation labels, giving both accuracy and Cohen's κ of 1.000. This agreement holds across all reasoning families, including formal logic and discrete structure, quantitative reasoning and optimization, and relational, planning, algorithmic, and semantic reasoning. Independent human annotators also showed high agreement, with pairwise κ values of 0.955–0.973.

These statistics measure different forms of agreement and should not be treated as comparable ceilings. Judge–human κ measures how closely the judge reproduces the human validation labels,

Table 8: **Human validation of the LLM judge.** Judge accuracy and κ are measured against the human validation labels. Human pairwise κ measures agreement between independent annotators.

Task family	n	**Judge acc.**	$\kappa_{\text{judge,human}}$	κ_{human}
Overall	720	1.000	1.000	0.962
Formal Logic & Discrete Structure	152	1.000	1.000	0.973
Quantitative Reasoning & Optimization	300	1.000	1.000	0.956
Relational, Planning, Algorithmic & Semantic	268	1.000	1.000	0.955

whereas pairwise human κ measures agreement between individual annotators. The key result for judge credibility is therefore the direct audit: on all 720 audited responses, the judge reproduced the human decision, including across reasoning families. Because this agreement is saturated, the audit cannot reveal the direction or magnitude of any undetected systematic bias. We therefore retain human validation as the reference procedure for ambiguous or disputed cases.

On LLM Judge Circularity. GEMMA-4-26B-A4B-INSTRUCT is both an evaluated model and the LLM judge, so potential judge circularity must be considered. The judge does not see generation traces, logits, hidden states, or model identity. It receives only the question, reference answer, and final response, and follows a structured rubric that compares the response against the reference answer.

A shared checkpoint could still introduce a model-specific preference for familiar answer styles, so the evaluation setup alone cannot rule this out. The human audit provides an empirical check. On all 720 audited responses, including every reasoning family, the judge exactly reproduced the human labels, with no observed disagreement that would indicate a model-specific effect. Thus, the audit supports the credibility of the judge on the validated sample, while human adjudication remains the final authority for contested cases.

‘‘‘

E.3 HUMAN BASELINES DEVELOPMENT

To construct reliable human baselines, we evaluate two distinct proficiency levels: a *Human (Base)* group and a *Human (Expert)* group. The Human (Base) cohort consists of annotators with educational backgrounds ranging from high-school graduates to early undergraduate (first-year bachelor) students, representing non-specialist reasoning ability. The Human (Expert) cohort comprises individuals with advanced STEM training, including Master's and PhD-level students, representing domain-competent expert reasoning.

Each group consists of three annotators, and for every evaluation sample we collect independent responses from all three. Final answers for both Human (Base) and Human (Expert) are determined via majority voting, ensuring robustness against individual variation and annotation noise. This aggregated protocol is used to report human performance across all experiments, providing a stable reference for comparison against model behavior.

All annotators evaluated the full benchmark, including all invariant variants. To mitigate recognition bias, questions were randomly sampled and presented in a shuffled order, with the constraint that semantically equivalent variants of the same seed problem were separated by at least 50 intervening questions. Annotators were not provided with ground-truth answers or feedback at any stage.

We acknowledge that this within-subject design may introduce partial recognition effects, potentially inflating human invariance performance relative to a strict between-subjects protocol. However, enforced spacing and randomization substantially reduce short-term recall, and each instance still requires independent recomputation under varied formulations. Accordingly, the reported human results should be interpreted as an approximate upper bound on invariance under controlled, non-adversarial evaluation conditions.

F METRIC DEFINITIONS AND INTERPRETATIONS

We evaluate each model on a benchmark of N *complete samples*. Each sample $i \in \{1, \ldots, N\}$ consists of $K = 4$ question variants:

- $x_{i,\text{base}} \in \{0, 1\}$ — the original (base) question,
- $x_{i,1} \in \{0, 1\}$ — Invariant $\mathcal{I}_1$: Symbolic & Linguistic reformulation,
- $x_{i,2} \in \{0, 1\}$ — Invariant $\mathcal{I}_2$: Structural & Representational reformulation,
- $x_{i,3} \in \{0, 1\}$ — Invariant $\mathcal{I}_3$: Abstraction & Information-Structure reformulation,

where 1 denotes a correct model response and 0 denotes an incorrect one. All four variants are semantically equivalent and logically identical — only their surface representation differs. A robust model should answer all four variants consistently. We define the *correctness count* for sample i as

$$c_i = x_{i,\text{base}} + x_{i,1} + x_{i,2} + x_{i,3} \ \in \ \{0, 1, 2, 3, 4\}. \tag{13}$$

Only samples for which all $K = 4$ variant scores are available are included in any metric computation. The set of invariant variants is denoted $\mathcal{I} = \{1, 2, 3\}$ and the full variant set $\mathcal{V} = \{\text{base}\} \cup \mathcal{I}$.

F.1 BASELINE ACCURACY METRICS

Accuracy per Variant. The per-variant accuracy is simply the mean correctness across all N samples:

$$\text{Acc}(v) = \frac{1}{N} \sum_{i=1}^{N} x_{i,v}, \qquad v \in \mathcal{V}. \tag{14}$$

The *mean invariant accuracy* averages the three invariant variants:

$$\overline{\mathcal{I}} = \frac{1}{3} \sum_{j \in \mathcal{I}} \text{Acc}(j). \tag{15}$$

The *invariance degradation gap* Δ measures how much performance drops from the base question to invariant reformulations:

$$\Delta = \text{Acc}(\text{base}) - \overline{\mathcal{I}}. \tag{16}$$

$\Delta > 0$ indicates that the model performs worse on reformulations than on the original question — the primary signal of *invariance fragility*. $\Delta \approx 0$ indicates robust, representation-agnostic reasoning. $\Delta < 0$ (rare) indicates that invariant forms are answered more reliably, potentially due to surface cues introduced by reformulation.

F.2 ICP: INVARIANCE CONSISTENCY PROFILE & IMI

The **Invariance Consistency Profile (ICP)** characterises the full empirical distribution of per-sample correctness counts $c_i \in \{0, \ldots, 4\}$:

$$P(c = k) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[c_i = k], \qquad k \in \{0, 1, 2, 3, 4\}. \tag{17}$$

This five-point distribution provides a richer picture than any single scalar: $P(c = 4)$ means the model got everything right; $P(c = 0)$ means the model failed all variants.

The **Invariant Mass Index (IMI)** is the mode of the distribution at complete correctness:

$$\text{IMI} = P(c_i = 4) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[\sum_{v \in \mathcal{V}} x_{i,v} = 4\right]. \tag{18}$$

Interpretation: Higher IMI is better. IMI is the strictest accuracy measure: it counts only samples where the model is *fully invariant* — correct on the base question *and* on all three reformulations. A model with high mean accuracy but low IMI is inconsistent: it gets some variants right by chance rather than through genuine understanding. IMI is equivalent to the ≥ 4 column in the at-least-k table (Section F.13).

F.3 CIR: Conditional Invariance Reliability

Let $\mathcal{B}^+ = \{i : x_{i,\text{base}} = 1\}$ be the set of samples where the model answers the base question correctly. **CIR** measures the fraction of those samples on which the model *also* answers every invariant reformulation correctly:

$$\text{CIR} = \frac{\left|\{i \in \mathcal{B}^+ : x_{i,1} = 1 \wedge x_{i,2} = 1 \wedge x_{i,3} = 1\}\right|}{|\mathcal{B}^+|}. \tag{19}$$

Interpretation: Higher CIR is better. CIR conditions on the model having the base knowledge and asks: given that the model *knows* the answer, does it remain correct when the question is re-expressed? A high CIR (≈ 1) means the model's knowledge is representation-stable. A low CIR means the model's correct responses on the base question are *not* robust — rephrasing collapses its performance even when the underlying reasoning should be sufficient.

F.4 ICR: Invariance Collapse Rate

A sample i is said to experience an *invariance collapse* if the model solves the base question but fails the majority of invariant variants. With $K - 1 = 3$ invariant variants, "majority wrong" is defined as $\sum_{j \in \mathcal{I}} x_{i,j} \leq \lfloor (K-1)/2 \rfloor = 1$, i.e. at most one of the three invariants is answered correctly. ICR is:

$$\text{ICR} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\left[x_{i,\text{base}} = 1 \ \wedge \ \sum_{j \in \mathcal{I}} x_{i,j} \leq 1 \right]. \tag{20}$$

Interpretation: Lower ICR is better. ICR quantifies the most pathological failure mode: the model *knows* the answer (base correct) but catastrophically fails upon reformulation. This is strong evidence that the model has memorised surface-level patterns associated with the base phrasing rather than internalising the underlying concept.

F.5 PIDI: Pairwise Invariance Dependency Index

PIDI quantifies the conditional co-failure probability between pairs of invariant variants. For each ordered pair $(j, k) \in \mathcal{I} \times \mathcal{I}$ with $j \neq k$:

$$\text{PIDI}(j,k) = P(x_{i,j} = 0 \mid x_{i,k} = 0) = \frac{|\{i : x_{i,j} = 0 \wedge x_{i,k} = 0\}|}{|\{i : x_{i,k} = 0\}|}. \tag{21}$$

All six directed pairs are reported: $P(\mathcal{I}_1{=}0|\mathcal{I}_2{=}0)$, $P(\mathcal{I}_2{=}0|\mathcal{I}_1{=}0)$, and so on.

Interpretation: Lower PIDI is better. If a model fails variant k, PIDI measures how likely it is to also fail variant j. High PIDI across all pairs indicates that failures are *correlated* — the model tends to fail multiple invariant types simultaneously rather than independently. This suggests a systematic reasoning deficit rather than random noise. Values near 1 across all pairs indicate that the model *always* fails together (or succeeds together), pointing to a single shared failure mechanism.

F.6 TCIR: Type-Wise Conditional Invariance Retention

Let $\mathcal{S} = \{i : c_i > 0\}$ be the set of samples where the model answered *at least one* variant correctly — the "solvable" samples. **TCIR** measures the per-invariant mean accuracy conditioned on solvability:

$$\text{TCIR}_j = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} x_{i,j}, \qquad j \in \mathcal{I}. \tag{22}$$

Interpretation: Higher TCIR is better. By restricting to solvable samples, TCIR removes items the model cannot answer in any representation (perhaps genuinely hard items), and isolates whether the model *retains* its reasoning when only the surface form changes. A low TCIR_j relative to other invariants indicates that invariant type j is a specific vulnerability — the model can answer the question in some representations but consistently fails in the $\mathcal{I}_j$ representation.

F.7 TCC: Type Collapse Contribution

A *partial collapse* occurs at sample i when $c_i \in \{1, 2\}$ — the model gets exactly one or two of the four variants right. Let $\mathcal{C} = \{i : c_i \in \{1, 2\}\}$. **TCC** measures how much each invariant type *contributes* to these collapse events:

$$\mathrm{TCC}_j = P(x_{i,j} = 0 \mid i \in \mathcal{C}) = \frac{|\{i \in \mathcal{C} : x_{i,j} = 0\}|}{|\mathcal{C}|}, \qquad j \in \mathcal{I}. \tag{23}$$

Interpretation: Lower TCC is better. TCC identifies which invariant types are the *primary drivers* of partial collapses. A high TCC_j means that whenever the model is in a partial collapse (not fully correct and not fully wrong), it is variant j that tends to be wrong. This pinpoints the weakest representational form.

F.8 RIS: Relative Invariance Sensitivity

Let $\bar{x}$ be the global mean accuracy across all variants and all samples:

$$\bar{x} = \frac{1}{K \cdot N} \sum_{i=1}^{N} \sum_{v \in \mathcal{V}} x_{i,v}. \tag{24}$$

RIS measures each invariant's deviation from this global baseline:

$$\mathrm{RIS}_j = \mathrm{Acc}(j) - \bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_{i,j} - \bar{x}, \qquad j \in \mathcal{I}. \tag{25}$$

Interpretation: RIS is a *signed* measure in percentage points. **Positive** values ($\mathrm{RIS}_j > 0$) mean invariant j is *easier than average* for this model; **negative** values mean it is harder. Values near zero indicate that the model treats this invariant form as equally difficult as the average. RIS is useful for comparing across models: a model with strongly negative RIS_3 is particularly sensitive to the Abstraction & Info-Structure reformulation.

F.9 BSS: Bilateral Symmetry Score

Two types of *asymmetric* failure can occur:

$$F^+ = \left|\{i : x_{i,\text{base}} = 1 \ \wedge \ \exists j \in \mathcal{I} : \ x_{i,j} = 0\}\right|, \tag{26}$$

$$F^- = \left|\{i : x_{i,\text{base}} = 0 \ \wedge \ \exists j \in \mathcal{I} : \ x_{i,j} = 1\}\right|. \tag{27}$$

F^+ counts *forward failures*: the model answers the original correctly but fails at least one reformulation. F^- counts *backward failures*: the model fails the original but recovers on at least one reformulation. The **Bilateral Symmetry Score** is:

$$\mathrm{BSS} = 1 - \frac{|F^+ - F^-|}{F^+ + F^-} \ \in \ [0, 1], \tag{28}$$

with $\mathrm{BSS} = \texttt{null}$ when $F^+ = F^- = 0$ (no asymmetric failures).

Interpretation: Higher BSS is better. $\mathrm{BSS} = 1$ means failures are *perfectly symmetric*: the model fails the base exactly as often as it fails invariants, indicating consistent difficulty rather than directional bias. $\mathrm{BSS} \approx 0$ means nearly all failures are in one direction.

- $F^+ \gg F^-$ (low BSS, high forward rate): the model knows the answer in the base phrasing but cannot maintain correctness under reformulation — the canonical *invariance collapse*.
- $F^- \gg F^+$ (low BSS, high backward rate): the model fails the base but recovers on reformulations — a strong signal of *surface-pattern exploitation*, where the reformulated phrasing inadvertently provides exploitable cues.

$\mathrm{BSS} < 0.4$ is flagged as highly asymmetric; $\mathrm{BSS} \in [0.4, 0.7)$ is moderate; $\mathrm{BSS} \geq 0.7$ is considered symmetric.

F.10 CE: CONSISTENCY ENTROPY

For each sample i, define the per-sample correct fraction $\hat{p}_i = c_i/K$. The *binary entropy* of this fraction is:

$$H(\hat{p}_i) = -\hat{p}_i \log_2 \hat{p}_i \; - \; (1-\hat{p}_i)\log_2(1-\hat{p}_i), \tag{29}$$

with $H(0) = H(1) = 0$ by convention. H reaches its maximum of 1 at $\hat{p}_i = 0.5$ ($c_i = 2$: exactly half the variants correct). **CE** averages this per-sample entropy:

$$\mathrm{CE} = \frac{1}{N}\sum_{i=1}^{N} H\Big(\frac{c_i}{K}\Big). \tag{30}$$

Interpretation: **Lower CE is better.** CE measures how *inconsistent* a model is at the per-sample level. A sample with $c_i \in \{0, 4\}$ contributes $H = 0$ — the model is fully consistent (either always right or always wrong). A sample with $c_i = 2$ contributes the maximum entropy $H = 1$.

- **Low CE** (< 0.25): the model's responses are structured — it consistently succeeds or consistently fails, suggesting genuine reasoning or genuine ignorance rather than noise.
- **High CE** (≥ 0.55): flagged as heuristic or random-looking, indicating the model frequently gets *some but not all* variants right in a way that appears stochastic rather than systematic.

F.11 IRR: INVARIANT RECOVERY RATE

Let $\mathcal{B}^- = \{i : x_{i,\text{base}} = 0\}$ be the set of samples where the model *fails* the base question. IRR conditions on these base-failed samples and asks whether invariant reformulations can *recover*:

$$\mathrm{IRR}_{\text{full}} = P\big(\forall j \in \mathcal{I} : x_{i,j} = 1 \;\big|\; i \in \mathcal{B}^-\big), \tag{31}$$

$$beginequation4pt]\mathrm{IRR}_{\text{part}} = P\big(\exists j \in \mathcal{I} : x_{i,j} = 1 \;\big|\; i \in \mathcal{B}^-\big), \tag{32}$$

$$beginequation4pt]\mathrm{IRR}_{\text{zero}} = 1 - \mathrm{IRR}_{\text{part}} = P\big(\forall j \in \mathcal{I} : x_{i,j} = 0 \;\big|\; i \in \mathcal{B}^-\big). \tag{33}$$

The signed **IRR-delta** compares recovery from base failure against retention from base success (CIR):

$$\mathrm{IRR}_{\Delta} = \mathrm{IRR}_{\text{full}} - (1 - \mathrm{CIR}). \tag{34}$$

Interpretation: A high $\mathrm{IRR}_{\text{full}}$ means that when the model fails the original question, all three reformulations still elicit correct responses — a strong indicator that the *base question itself* is harder than its invariant forms (possible due to base phrasing complexity or training exposure differences). $\mathrm{IRR}_{\Delta} > 0$ means the model finds the base question harder than its invariants; $\mathrm{IRR}_{\Delta} < 0$ means the invariants are harder. High $\mathrm{IRR}_{\text{part}}$ with low $\mathrm{IRR}_{\text{full}}$ indicates that the model can occasionally exploit an invariant phrasing but cannot generalise across all reformulations.

F.12 FG: FRAGILITY GRADIENT

The three invariant types are designed to increase in reformulation difficulty: $\mathcal{I}_1$ (Symbolic & Linguistic) is the most surface-level change; $\mathcal{I}_3$ (Abstraction & Information-Structure) requires the deepest representational re-encoding. A model with genuine understanding should respect this ordering.

Define the *gradient steps*:

$$g_{12} = \mathrm{Acc}(\mathcal{I}_1) - \mathrm{Acc}(\mathcal{I}_2), \qquad g_{23} = \mathrm{Acc}(\mathcal{I}_2) - \mathrm{Acc}(\mathcal{I}_3). \tag{35}$$

The expected direction is $g_{12} \geq 0$ and $g_{23} \geq 0$.

$$\mathrm{FG}_{\text{score}} = \mathbf{1}[g_{12} \geq 0] + \mathbf{1}[g_{23} \geq 0] \;\in\; \{0, 1, 2\}, \tag{36}$$

$$beginequation4pt]\mathrm{FG}_{\text{mag}} = \frac{g_{12} + g_{23}}{2} \quad \text{(signed mean gradient in pp)}, \tag{37}$$

$$beginequation4pt]\mathrm{FG}_{\text{inv}} = \mathbf{1}[\mathrm{FG}_{\text{score}} = 0]. \tag{38}$$

Interpretation: **Higher $\mathrm{FG}_{\text{score}}$ is better** (maximum 2). $\mathrm{FG}_{\text{score}} = 2$ indicates the model respects the expected difficulty ordering: it performs best on the easiest invariant type and progressively worse on harder types. $\mathrm{FG}_{\text{inv}} = \text{True}$ (both gradients inverted, score $= 0$) is a red flag: the model performs *better* on harder reformulations. This is inconsistent with genuine understanding and is attributed to *surface-pattern exploitation* — the harder reformulations may inadvertently contain surface cues (e.g. unusual phrasing, more explicit logical structure) that the model latches on to.

A negative FG_{mag} of large magnitude confirms the gradient is inverted, not merely flat.

F.13 At-Least-k Correct Analysis

Let $\mathcal{A}_k = \{i : c_i \geq k\}$ for $k \in \{1, 2, 3, 4\}$. The **at-least-k-correct** proportion is:

$$P(\geq k) = \frac{|\mathcal{A}_k|}{N} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[c_i \geq k], \qquad k = 1, 2, 3, 4. \tag{39}$$

These four quantities form a nested, monotonically decreasing sequence:

$$P(\geq 1) \;\geq\; P(\geq 2) \;\geq\; P(\geq 3) \;\geq\; P(\geq 4) = \mathrm{IMI}. \tag{40}$$

The complementary *none-correct* proportion is:

$$P(= 0) = 1 - P(\geq 1) = \frac{|\{i : c_i = 0\}|}{N}. \tag{41}$$

Relationship to ICP. The at-least-k statistics are the complementary cumulative distribution function (CCDF) of the correctness count distribution defined in the ICP (Section F.2):

$$P(\geq k) = \sum_{m=k}^{4} P(c = m). \tag{42}$$

Interpretation: All four at-least-k values are **higher is better**; the none-correct proportion is **lower is better.**

- $P(\geq 1)$ — the model answered at least one of the four formulations correctly. This is the loosest possible measure of task-relevant knowledge and has a high ceiling even for weak models.
- $P(\geq 2)$ — at least two formulations correct. A threshold of $P(\geq 2) \approx P(\geq 1)$ (near-flat drop) indicates that the model is consistent once it gets any variant right.
- $P(\geq 3)$ — at least three of the four formulations correct. A large gap between $P(\geq 2)$ and $P(\geq 3)$ signals that the model frequently knows the answer in *some* surface forms but not others.
- $P(\geq 4)$ **= IMI** — the model is fully invariant. The gap $P(\geq 3) - P(\geq 4)$ reflects how often the model passes three of four variants but fails exactly one invariant type.
- $P(= 0)$ — the fraction of questions the model cannot answer in any representation. A high $P(= 0)$ reflects genuine knowledge gaps, not reformulation sensitivity.

Illustrative degradation profile. A model with $P(\geq 1) = 0.92$, $P(\geq 2) = 0.75$, $P(\geq 3) = 0.55$, $P(\geq 4) = 0.41$ shows a steep decline: the 0.17 drop from $P(\geq 1)$ to $P(\geq 2)$, the 0.20 drop to $P(\geq 3)$, and the 0.14 drop to $P(\geq 4)$ indicate that each additional reformulation constraint eliminates a sizable fraction of otherwise-correct samples — a hallmark of surface-level reasoning.

Table 9 provides a consolidated reference for all metrics, their directionality, the question they answer, and their failure interpretation.

F.14 Why these metrics are contamination-aware?

Let s_0 denote the base instance and $s_k = \tau_k(s_0)$ its invariant variants, where τ_k is semantics-preserving (Eq. equation 1). A model that has memorised s_0 can achieve high $\mathrm{Acc}(\text{base})$ without

Table 9: Summary of all invariance evaluation metrics. ↑ = higher is better; ↓ = lower is better.

ID	Metric	Dir.	Question answered	Low value means…
Δ	Invariance Gap	↓	How much does accuracy drop under reformulation?	Small/no degradation
IMI	Invariant Mass Index	↑	What fraction of samples are fully invariant?	Model rarely stays correct across all forms
CIR	Cond. Invariance Reliability	↑	Given base correct, are reformulations also correct?	Correct base knowledge does not transfer
ICR	Invariance Collapse Rate	↓	How often does knowing the base lead to collapse?	Rare catastrophic failure mode
PIDI	Pairwise Inv. Dependency	↓	Are failures correlated across invariant types?	Independent failures (less systematic)
TCIR	Type-Wise CIR	↑	Which invariant type is the weakest, for solvable items?	Specific invariant type is a vulnerability
TCC	Type Collapse Contrib.	↓	Which invariant type drives partial collapses?	That invariant type is rarely the weak link
RIS	Relative Inv. Sensitivity	signed	Is this invariant easier or harder than average?	Harder than the model's global mean
BSS	Bilateral Symmetry Score	↑	Are failures direction-symmetric across base↔inv?	Asymmetric — potential pattern exploitation
CE	Consistency Entropy	↓	How random are per-sample correctness patterns?	Structured, consistent model behaviour
$\mathrm{IRR}_{\mathrm{full}}$	Invariant Recovery Rate	↑	Can invariants recover when base fails?	Model cannot recover through reformulation
IRR_{Δ}	IRR Delta	signed	Is the base harder or easier than invariants?	Invariants are harder than the base
FG	Fragility Gradient	↑	Does performance follow the expected difficulty order?	Gradient violated — possible surface cues
$P(\geq k)$	At-Least-k Correct	↑	How broadly correct is the model across variants?	Model rarely sustains correctness

exhibiting invariant reasoning. However, since $s_k \neq s_0$ at the string level, success on s_k requires recognising the equivalence class $[s_0]_{[\![]}$ rather than recalling a specific form. Consequently, metrics that depend on joint correctness across variants—such as Δ, IMI, and CIR—cannot be inflated by memorisation of s_0 alone. In particular, the gap Δ and consistency-based measures provide a lower bound on representation-invariant competence: any improvement on these metrics necessarily implies the ability to generalise across semantics-preserving transformations, rather than reliance on surface-form recall.

G Additional Experiments and Results

G.1 On Negative Invariance Gaps

In Table 1, a subset of models exhibit non-positive average invariance gaps ($\Delta \leq 0$), indicating that performance on invariant variants is, on average, equal to or slightly higher than the base formulation. While this may appear to contradict the broader claim of invariance degradation, such aggregate behavior does not necessarily reflect improved reasoning. In practice, invariance transformations are heterogeneous: some operations (e.g., distractor removal or simplification) can make instances easier, while others (e.g., abstraction shifts or distractor injection) increase difficulty. As a result, per-sample gains and losses can cancel out when averaged, producing a near-zero or negative Δ despite underlying instability.

Crucially, this phenomenon often reflects inconsistency rather than robustness. A model may fail on the base form of one problem but succeed on its invariant variant, while exhibiting the opposite behavior on another instance. These offsetting errors lead to superficially stable averages without

indicating true invariance. Therefore, negative or near-zero gaps should not be interpreted as evidence of invariant reasoning, but rather as a symptom of unstructured sensitivity to representation.
To address this limitation, we introduce complementary metrics that explicitly capture consistency across variants, such as the Invariant Mass Index (Table 2) and at-least-k correctness profiles. These measures evaluate whether a model can simultaneously maintain correctness across multiple equivalent forms of the same problem, rather than averaging performance across unrelated instances. As shown in our results, models with near-zero Δ often still exhibit low consistency scores, confirming that average accuracy alone is insufficient to characterize invariance behavior.

G.2 On Semantics-Preservation and Perturbation Axes

though all transformation axes preserve the target semantics by construction, they need not impose identical cognitive demands. For example, distractor injection/removal and abstraction-level shifts may change the amount or form of information that must be processed while leaving the underlying answer unchanged. We therefore distinguish between *pure invariance* transformations, which primarily alter surface representation, and *controlled perturbations*, which preserve semantics while potentially modulating task difficulty.

We group variable renaming, entity relabeling, premise reordering, logical/mathematical reformulation, encoding swaps, and paraphrastic variation as pure invariance axes. Distractor injection/removal and abstraction-level shifts are treated as controlled perturbations. We report aggregate and axis-wise results to separate consistency under re-expression from sensitivity to these broader perturbations.

This distinction does not imply that the controlled perturbation axes are invalid invariance tests. Rather, it makes explicit the different factors they probe. Our results remain consistent when focusing on the pure invariance axes, while the controlled perturbations provide additional evidence about how models respond to semantically equivalent inputs that differ in informational structure. We therefore interpret the benchmark as measuring both strict invariance to representational changes and broader robustness to semantics-preserving variation, rather than assuming that every transformation is difficulty-neutral.

G.3 Fragility Gradient and Non-Monotonic Difficulty

Table 3 shows that the Fragility Gradient (FG) rarely attains the ideal monotonic ordering (score = 2), with most models clustering at FG = 1 and even GPT-5.2 at FG = 0, while Human (Expert) also does not achieve perfect monotonicity. This outcome does not contradict the framework but rather highlights an important empirical reality: the assumed ordering $\mathcal{I}_1 < \mathcal{I}_2 < \mathcal{I}_3$ is a coarse, aggregate heuristic rather than a strict per-instance property. In practice, different invariance axes interact with task structure in non-uniform ways, and the relative difficulty of variants can fluctuate depending on the underlying reasoning type. For example, a structural reformulation may be harder than a symbolic renaming in one task but easier in another, breaking global monotonicity even for humans.
Therefore, FG should be interpreted as a diagnostic signal of stability rather than a strict ranking validator. A score of 1 indicates partial ordering consistency, while deviations (FG = 0) reflect heterogeneous sensitivity across invariance axes rather than failure of the metric itself. The fact that even human annotators do not consistently achieve FG = 2 further reinforces that invariance difficulty is not inherently linear, but context-dependent. Retaining FG is still informative, as it captures whether models exhibit coherent degradation trends, but we explicitly avoid over-interpreting it as evidence of a universal difficulty hierarchy.

H Full Per-Task and Per-Subtype Results

The complete per-task and per-subtype results are provided in a separate supplementary material PDF. The full breakdown comprises 84 tables and would substantially increase the length of the main manuscript and appendix PDF; we therefore provide the complete analysis separately for completeness and reproducibility.

I Error and Reasoning Failure Mode Analysis

I.1 Setup and aggregate behavior

We analysed every response of seven contemporary instruction-tuned language models on a 1159-item reasoning benchmark covering three task families — *A. Formal Logic & Discrete Structure*, *B. Quantitative Reasoning & Optimization*, and *C. Relational, Planning, Algorithmic and Semantic Reasoning* — each presented in four equivalent surface forms (a `base` item plus three semantically-preserving rephrasings, `inv1`, `inv2`, `inv3`), giving 4636 scored question–response pairs per model. The seven models studied are Claude-Haiku-4.5, DeepSeek-V3.1, GPT-5, Gemma-4-26B, Llama-3.3-70B, Olmo-3.1-32B, and Qwen3.6-Plus. Aggregate failure counts are summarized in Table 10.

Table 10: Aggregate failure counts and failure rates for each model and human group across all 4,636 items.

Model	Failures / 4636	Failure rate
GPT-5.2	1479	31.9%
Qwen3.6-Plus	1577	34.0%
Qwen3-8B	2038	44.0%
Claude-Haiku-4.5	2386	51.5%
Gemini-3-Flash	2788	60.1%
Gemma-4-26B-A4B-I	3342	72.1%
Llama-3.3-70B-I	3448	74.4%
Command-A	3635	78.4%
DeepSeek-V3.1	3779	81.5%
o4-mini	3797	81.9%
Olmo-3.1-32B-I	3890	83.9%
Aya-Expanse-32B	3933	84.8%
Ministral-3B-2512	4123	88.9%
Gemma-3-4B-I	4148	89.5%
Llama-3.2-1B-I	4299	92.7%
Human (Base)	462	10.0%
Human (Expert)	17	0.4%

We performed a two-pass qualitative coding of these failures, stratified by model, task sub-type, and invariant axis, and supplemented this with corpus-level lexical analysis of the rubric judges' reasoning notes. The result is the eight-category, 48-subcategory taxonomy summarised in this section. The eight top-level categories partition the space of failure *loci* along the reasoning pipeline: input parsing (E1), formalisation (E2), local computation (E3), chain-coherence (E4), invariance robustness (E5), lexical-semantic selection (E6), output specification (E7), and generation pathology (E8). The taxonomy is intentionally pipeline-aligned: each category corresponds to a stage where a model can fail *even if all preceding stages succeeded.*

I.2 E1. Premise-comprehension failure

E1 failures arise when the model does not correctly internalize the problem statement before reasoning begins. In such cases, constraints are dropped, quantifiers are misinterpreted, the question target is misidentified, or surface-level artifacts such as premise order or distractors distort the internal representation. Since all subsequent reasoning is conditioned on this initial encoding, even small E1 errors propagate and lead to confidently incorrect final answers.

Why it happens. We identify two dominant mechanisms. First, *long-context attention dilution*, where load-bearing constraints are overshadowed in prompts containing many structurally similar tokens (e.g., repeated rates, quantities, or conditions), causing critical modifiers to be weakly encoded in the working representation. Second, *prior-driven solution substitution*, where the model recognizes a familiar problem template and applies a memorized solution pattern while ignoring problem-specific modifications.

Subcategories.

E1.1 Dropped or ignored constraint. A semantically critical constraint is omitted during internalization, leading to a systematically incorrect solution.

E1.2 Quantifier / scope misreading. The model incorrectly interprets logical scope, particularly for universal and existential quantifiers.

E1.3 Goal / question-target misidentification. The model answers a related but incorrect target within the same problem. EXAMPLE. Item `b1d1239a08fe4fbe` (`B5. Matrix/Linear Algebra`) asks which property must hold for a 4×4 matrix A satisfying $A^3=0$. The correct answer is that all eigenvalues are 0. CLAUDE-HAIKU-4.5 instead states that A is not invertible, which is true but not the queried property.

E1.4 Distractor capture. Information introduced as distractors is incorrectly treated as constraint-bearing, leading to corrupted reasoning. This failure mode is most pronounced in OLMO-3.1-32B (11.5% breakage), LLAMA-3.3-70B (14.0%), and GEMMA-4-26B (14.0%).

E1.5 Premise-order brittleness. We observe systematic failure under premise reordering, where logically equivalent problems lead to inconsistent outputs. Per-model breakage rates are: CLAUDE-HAIKU-4.5 (19.6%), OLMO-3.1-32B (10.6%), and LLAMA-3.3-70B (10.6%). EXAMPLE. For item `16aa090e103e4f85`, the *base* version asks for the number of paths from S to T in a directed acyclic graph, and CLAUDE-HAIKU-4.5 correctly outputs 4. However, under invariant transformation `inv2`, where the same edges are presented in a permuted order (e.g., $D{\to}G$, $S{\to}B$, $I{\to}T$, $A{\to}C$, etc.), the model fails to preserve equivalence and incorrectly concludes that there are no valid paths, reasoning that "nothing points to I ... there are 0 total paths." This illustrates sensitivity to superficial ordering despite unchanged underlying structure.

E1.6 Implicit-assumption injection. The model introduces unstated or default assumptions that override explicitly provided values in the problem statement. EXAMPLE. Item `85cf10cde3b247d5` (*base*) considers a vertical circular motion setup with $m{=}4\,\mathrm{kg}$, $r{=}1.5\,\mathrm{m}$, and $v_{\text{top}}{=}6\,\mathrm{m/s}$, asking for the tension at the bottom of the circle. The correct solution is $292\,\mathrm{N}$ using $g{=}9.81\,\mathrm{m/s^2}$. However, CLAUDE-HAIKU-4.5 replaces the specified gravitational constant with $g{=}10$, yielding $296\,\mathrm{N}$, while also noting an alternative value of $\sim 300\,\mathrm{N}$ for $g{=}9.8$. This indicates awareness of the correct convention but a failure to consistently adhere to the problem-specified physical parameters.

I.3 E2. SYMBOLIC / LOGICAL TRANSLATION FAILURE

This category captures errors at the natural-language $\to$ formalism interface, where the model has generally understood the underlying problem but fails to express the solution in the required formal language, or produces a representation that is structurally close yet semantically or syntactically misaligned with the expected form.

Why it happens. A key source of failure is the mismatch between sparsely represented formal logics (first-order, modal, epistemic) and richly represented natural-language reasoning. In addition, rubric constraints enforce canonical conventions (e.g., predicate signatures, quantifier ordering, normal forms) that are not mathematically necessary for correctness but are required for evaluation consistency. As a result, even logically equivalent answers may be marked incorrect. A frequent manifestation is *translation refusal*, where the model retains correct semantics but avoids formal notation and instead responds in natural language.

Subcategories.

E2.1 Variable role swap. Incorrect alignment of argument order, producing expressions such as $F(y,x)$ instead of $F(x,y)$. This is particularly prevalent in A1-style logical tasks across DEEPSEEK-V3.1, OLMO-3.1-32B, and LLAMA-3.3-70B.

E2.2 Quantifier-order / type confusion. Errors involving $\forall/\exists$ scope or ordering, such as confusing $\forall x\exists y$ with $\exists y\forall x$. EXAMPLE. Item `0bcf8d391e1a4985`: "Every driver is certified for at least one route category." The correct formalisation is $\forall d\exists r(\mathrm{Driver}(d) \to \mathrm{Certified}(d,r))$. CLAUDE-HAIKU-4.5 instead produces

$\forall x(\text{Driver}(x) \rightarrow \exists y(\text{Certified}(x, y) \wedge \text{RouteCategory}(y)))$, which is semantically close but introduces an unintended predicate constraint, leading to rubric mismatch.

E2.3 Refusal-to-formalise / natural-language fallback. The model produces a correct paraphrase but fails to emit the required formal expression. EXAMPLE. Item `0b080fc57e474451` (`inv1`, A1): instead of $\neg\exists a\,\forall m\,(\text{Analyst}(a) \wedge \text{Covered}(a, m))$, CLAUDE-HAIKU-4.5 answers: "Every analyst failed to cover at least one market." The semantics are correct, but the required formal representation is omitted.

E2.4 Predicate / signature drift. Introduction of extra predicates or decomposition of a single relation into multiple components, yielding structurally inconsistent but semantically equivalent representations that violate the expected schema.

E2.5 Modal / epistemic operator mishandling. Failures involving $\Box$, $\Diamond$, and epistemic operators such as K_A, K_B. EXAMPLE. Item `db3546f3eeae45a4` (A5): given a set of knowledge constraints, the correct conclusion is inconsistency. CLAUDE-HAIKU-4.5 instead derives $K_A(K_B(P)) \wedge K_A(K_B(\neg Q))$, incorrectly treating distributive knowledge across agents as valid, violating standard epistemic closure principles.

E2.6 Logical-equivalence rejection. The model produces a logically equivalent form in a different representation (e.g., DNF vs. CNF) or a structurally valid but rubric-misaligned classification. EXAMPLE. Item `bb403d5df2cc4c51`: the canonical answer identifies both graphs as isomorphic to the Petersen graph, but CLAUDE-HAIKU-4.5 returns "No," indicating failure to recognize equivalence under the required canonical form.

I.4 E3. COMPUTATION & ARITHMETIC FAILURE

This category captures cases where the high-level reasoning structure is correct, but a local computational step is executed incorrectly. Typical errors include linear-system miscalculation, sign inversion, off-by-one mistakes, incorrect intermediate aggregation, and unit inconsistencies. These failures are generally traceable from the model's intermediate steps and do not require deep semantic interpretation to diagnose.

Why it happens. In current models, arithmetic is not performed via explicit algorithmic execution but through learned pattern completion over latent representations. As a result, intermediate quantities beyond a few significant digits are unstable, and errors accumulate through propagation rather than discrete computation. Additionally, sign and unit inconsistencies arise when the model conflates multiple valid conventions (e.g., coordinate direction, physics sign conventions, or unit scaling), leading to internally coherent but externally incorrect outputs.

Subcategories.

E3.1 System-of-equations solving error. Errors in solving linear systems, particularly prevalent in B1 and B4-style constraint problems involving bundled or coupled equations.

E3.2 Sign / direction error. Incorrect handling of directional or signed quantities, often triggered by representation changes. EXAMPLE. Item `3eb72f494b0c4741` (***C2, lens problem***, `inv2` encoding): the model correctly solves the base form as $v = 100$ cm, but under tabular encoding incorrectly applies $m = -v/u \Rightarrow v = -3u$, resulting in $v = -50$ cm. The transformation preserves structure but flips physical sign convention.

E3.3 Off-by-one / boundary error. Errors in discrete counting or indexing, especially in A2-style cardinality and constraint enumeration tasks. These arise from incomplete boundary tracking during iterative counting.

E3.4 Combinatorial mis-enumeration. Incorrect counting of valid assignments in logical or constraint-satisfaction problems.

E3.5 Floating-point / precision drift. Loss of numerical fidelity due to intermediate rounding. EXAMPLE. Item `39a88038785c42a8`: the correct value is 0.737, but CLAUDE-HAIKU-4.5 returns 0.74, which is rejected by the rubric due to insufficient precision despite being numerically close.

E3.6 Unit / dimensional inconsistency. Mixing or misalignment of measurement units (e.g., cm vs. m, percent vs. decimal form), or inconsistent use of physical constants such as $g = 9.81$ vs.

$g = 10$. This failure mode is widespread across all models, with only minor variation in severity.

I.5 E4. MULTI-STEP REASONING / CHAIN-COHERENCE FAILURE

This category captures failures where individual intermediate steps are locally correct, but the global reasoning trajectory becomes inconsistent. The model fails to maintain a stable latent state across multiple inference steps, causing later operations to violate earlier constraints or overwrite previously correct deductions. As a result, intermediate verifications are either ignored or actively contradicted in subsequent reasoning.

Why it happens. In long chain-of-thought generation, models progressively shift attention from the original input to their own recently generated tokens. This leads to a drift in the effective working memory, where earlier constraints are no longer reliably retained. The failure is particularly pronounced in problems involving conditional logic, where the model must simultaneously track a running computation and a gating condition (e.g., thresholds, discount rules, or validity checks). When these two streams are not jointly maintained, later reasoning steps override earlier correct state estimates.

Subcategories.

E4.1 State-tracking loss in graph traversal. Failure to maintain correct dynamic programming state across nodes in graph or path-counting problems. EXAMPLE. Item `24d0727b6d73472a` (*C3 base*): the correct number of paths from S to T is 5. CLAUDE-HAIKU-4.5 correctly computes intermediate paths up to node E (yielding 2 paths) but incorrectly re-binds state at node F, double-counting a path via $D \to E \to F$, and outputs 8.

E4.2 Conditional-rule sequencing error. Incorrect ordering or application of constraints that depend on prior conditions. EXAMPLE. Item `9ea2e088806a4b4c` (*B2 base*): a discount applies only when quantity ≥ 7. Although CLAUDE-HAIKU-4.5 correctly solves the system and identifies a consistency check failure, it proceeds by overriding the condition and returns 196 instead of the correct 164.

E4.3 Bracket/tier transition error. Failures in maintaining correct segment boundaries in piecewise or tiered systems (e.g., tax brackets or pricing tiers). EXAMPLE. Item `907c047889ef4c64` (*C4*): correct computation yields 75.50. The model initially produces this value, then re-segments the brackets mid-response and outputs 82.50, before finally stating 65.50 as the reported answer, indicating unstable tier tracking.

E4.4 Self-contradiction / mid-chain reversal. The model explicitly revises a correct intermediate step without justification, leading to degraded final outputs. This pattern is frequently observed in CLAUDE-HAIKU-4.5, with 26 documented cases where a correct computation is followed by "Wait, let me recalculate," resulting in a worse final answer.

E4.5 Verification-failure ignored. The model correctly identifies an inconsistency during self-checking but proceeds regardless. EXAMPLE. A computed equality check shows $2(14) + 21 + 3(11) = 82 \neq 74$, yet the model continues with the incorrect branch, discarding its own verification signal. This reflects a failure in using verification as a control mechanism rather than generation-only behavior.

E4.6 Unjustified expert-name shortcut. The model prematurely assigns a known structure without full structural verification. EXAMPLE. Item `d9bd5ae675384c7c` (*A3 base*): after correctly verifying properties consistent with multiple candidate graphs (including 3-regularity and edge count), CLAUDE-HAIKU-4.5 asserts the graph is the Möbius–Kantor graph or similar, and concludes equivalence with H, despite the canonical solution identifying it as the cuboctahedral graph with distinct structural properties.

I.6 E5. INVARIANCE BRITTLENESS

This is the defining phenomenon studied in the benchmark: a model produces different answers for semantically equivalent reformulations of the same underlying problem. As shown in Table 11, no

model is fully invariant. Even the strongest system, GPT-5, exhibits a non-trivial loss of approximately 7–12% of base-correct answers under at least one invariance axis, indicating that correctness is not stable under representation-preserving transformations.

Table 11: Breakage rate (%): base item passed but the corresponding invariance variant failed. DEEPSEEK-V3.1 is omitted from the table because its "base pass" set is small (n=859) and rates fluctuate, but its breakage profile is qualitatively similar to OLMO-3.1-32B. CLAUDE-HAIKU-4.5 is highest along nearly every axis — a function of its high *base* pass rate creating more opportunities to break, not necessarily of more fragile reasoning.

Invariance axis	Breakage rate (base pass → inv fail)					
	GPT-5	Qwen	Claude	Llama	Gemma	Olmo
Abstraction Level Shift	11.6	14.9	21.0	14.9	15.2	14.6
Distractor Inj./Removal	10.8	9.1	14.3	14.0	14.0	11.5
Entity Relabeling	9.2	7.2	14.5	14.1	8.2	6.9
Equiv. Logical Reform.	7.4	7.1	13.0	11.8	16.7	8.6
Equiv. Math. Reform.	10.7	12.3	17.6	10.2	11.9	6.1
Perspective / Role	8.4	10.1	19.4	13.2	14.1	8.8
Premise Reordering	9.7	9.7	19.6	10.6	9.0	10.6
Redundant Restatement	3.1	5.6	10.1	7.0	8.1	6.1
Repr./Encoding Swap	11.9	11.5	21.1	10.1	8.8	7.0
Style/Register	9.8	8.1	14.0	9.8	13.3	8.8
Synonym/Paraphrase	8.8	10.0	15.8	9.6	11.9	11.2
Variable Renaming	9.4	9.4	14.5	11.4	11.4	10.0

Why it happens. The failure arises from the interaction of two mechanisms. First, *shallow surface anchoring*, where the model conditions its reasoning path on lexical patterns that change under paraphrase, leading to a different implicit solution strategy even when the underlying problem is unchanged. Second, *role-by-position binding*, where renamed or reordered entities are re-associated based on surface order rather than structural constraints, causing systematic role misalignment under transformations such as entity relabeling or permutation.

Subcategories.

E5.1 Premise-reordering brittleness. Sensitivity to clause or premise permutation without any semantic change. Up to 19.6% of base-correct answers in CLAUDE-HAIKU-4.5 are lost under simple reordering. EXAMPLE. Item `16aa090e103e4f85` (see E1.5), where permutation of identical constraints changes the model's conclusion despite identical logical content.

E5.2 Entity-relabeling brittleness. Failure under renaming of objects, agents, or variables while preserving structure. EXAMPLE. Item `464ad049f9aa427d` (*B2*): the same bundle-pricing problem under `inv2` causes CLAUDE-HAIKU-4.5 to flip from 172 (correct) to 178, purely due to entity renaming.

E5.3 Variable-renaming brittleness. Sensitivity to purely symbolic renaming (e.g., $A, B, C \rightarrow P, Q, R$ or $n \rightarrow m$). Most affected models are CLAUDE-HAIKU-4.5 (14.5%), DEEPSEEK-V3.1 (12.5%), and LLAMA-3.3-70B (11.4%), indicating weak abstraction over variable identity.

E5.4 Abstraction-level shift brittleness. The most widespread invariance failure, occurring when the same problem is expressed at different abstraction levels (e.g., narrative ↔ formal notation). Across models, 11.6%–21.0% of base-correct solutions break under this shift, with CLAUDE-HAIKU-4.5 showing the highest sensitivity (21.0%).

E5.5 Representation / encoding swap. Failure under conversion between equivalent representations (e.g., prose to table or equation form). This effect is especially pronounced when structural cues are redistributed across formats, as illustrated in the lens problem example in E3.2.

E5.6 Perspective / role reframing. Instability under changes in agent perspective or role assignment, where constraints remain identical but viewpoint changes. Error rates reach 19.4% for

CLAUDE-HAIKU-4.5 and 14.1% for GEMMA-4-26B, indicating inconsistent role grounding under reframing.

I.7 E6. LEXICAL / SEMANTIC SELECTION FAILURE

This category captures failures on reading-comprehension and word-choice tasks (primarily C5), where the model selects a near-neighbour of the correct answer that is semantically plausible but not rubric-acceptable. The error manifests as substitution with a synonym, antonym, thematic associate, incorrect analytical framing, or mismatched register. This failure mode dominates performance degradation on C5 across all evaluated models.

Why it happens. Many SAT-style and cloze-form items admit multiple semantically related candidates, but only one is pragmatically licensed under the rubric. Models trained on large corpora tend to prefer high-likelihood lexical completions, which often correspond to frequent synonyms (e.g., "reflected in", "consistent with") rather than the precise target phrase (e.g., "manifest in", "co-extensive with"). This bias is amplified in negation-heavy or polarity-sensitive contexts, where embedded logical structure is not reliably tracked. As a result, models optimize for fluency rather than rubric-constrained semantic precision.

Subcategories.

E6.1 Synonym substitution. Replacement of the target phrase with a semantically close but non-equivalent synonym. EXAMPLE. Item `92ecc3eefc044b01`: "Morris's repudiation of industrialization is ___ the Kelmscott editions' use of handmade materials." Correct answer: "manifest in". OLMO-3.1-32B answers "consistent with", while CLAUDE-HAIKU-4.5 answers "reflected in", both failing rubric specificity.

E6.2 Antonym / polarity inversion. Selection of a semantically opposite term due to polarity misinterpretation. EXAMPLE. Item `b1f7163280af43eb`: correct answer is "homogeneous". OLMO-3.1-32B returns "heterogeneous", inverting the intended density condition and violating the underlying inference.

E6.3 Wrong-axis inference. Correctly understanding the domain but selecting an incorrect abstraction level or response type (e.g., instance vs. category). EXAMPLE. Item `5efcb6c6d66a4fe2`: the correct classification is "Free from". LLAMA-3.3-70B instead provides a descriptive statement ("contains no allergen X"), which is semantically related but fails to match the required categorical output.

E6.4 Document-cross-reference failure. Inability to integrate constraints across multiple sources or passages. EXAMPLE. Item `6d5f1072770543d9`: combining Doc1 and Doc2 yields a correct total of 52%, implying disagreement with Doc2 alone. OLMO-3.1-32B answers "Yes" by relying solely on Doc2, ignoring cross-document reconciliation.

E6.5 Idiomatic / register mismatch. Selection of an answer with incorrect stylistic or register alignment (e.g., informal vs. formal usage), even when semantically similar to the correct option.

E6.6 Pragmatic-cue blindness. Failure to incorporate discourse-level qualifiers such as "typically", "independently", or "with little in common", leading to systematic overgeneralization of the candidate answer space.

I.8 E7. FORMAT / SPECIFICATION COMPLIANCE FAILURE

In this class of errors, the model often arrives at a correct or near-correct solution, but fails at the level of output specification. The final answer is rejected not due to reasoning mistakes, but due to violations of formatting, structural, or response-contract constraints imposed by the benchmark rubric.

Why it happens. This failure mode arises from two main sources. First, many evaluation rubrics operate as strict-match or format-sensitive graders, where mathematically equivalent expressions may still be marked incorrect if they deviate from the expected representation. Second, models carry strong default output priors (e.g., simplifying fractions, rounding to fixed precision, or preferring

natural language explanations) that may conflict with dataset-specific output requirements, leading to mismatches even when the underlying reasoning is sound.

Subcategories.

E7.1 Decimal vs. fraction mismatch. EXAMPLE. Item `043c9dee9e8b436b`: computing $P(\text{div by } 8 \mid \text{div by } 4)$ for $n \in [1, 100]$. The correct value is 0.48 (equivalently $12/25$). GPT-5 returns $12/25$, which is mathematically identical but depends on whether the judge accepts fractional form equivalence, highlighting sensitivity to grading policy.

E7.2 Multi-part answer incompleteness. Item `ff22b72bc81e4cba` requires both the number of defective items and the identification of which retailer reports first. Several models correctly solve one component but omit the second, resulting in partial but invalid submissions under the rubric.

E7.3 Spurious extra element. EXAMPLE. Item `f002a3f1e73c4b08`, symmetric difference of $((L \cup M) \cap N)$ and O. The correct answer is $\{1, 4, 5, 6, 7\}$. CLAUDE-HAIKU-4.5 outputs $\{1, 4, 5, 6, 7, 12\}$, propagating an extraneous element from an intermediate step without re-verification.

E7.4 Notation / format drift. Structural equivalence holds, but notation differs from the required canonical form (e.g., determinant expansions written in permuted index order rather than the expected Levi-Civita expansion format), leading to rejection despite correctness.

E7.5 Refusal / "insufficient information" on solvable items. EXAMPLE. Item `0fa6761d40d942d6` (*C2*, heel-diameter problem). CLAUDE-HAIKU-4.5 responds that the heel diameter cannot be determined, citing missing geometric constraints, despite the problem being fully specified. This reflects overcautious abstention rather than true underdetermination.

I.9 E8. GENERATION / OUTPUT PATHOLOGY

The output trajectory itself fails. The model may loop, garble, truncate, abandon a recalculation, or return no usable response at all. These failures are distinct from classical reasoning errors: in these cases, the model does not produce a complete, well-formed answer that can even be meaningfully evaluated by the rubric.

Why it happens. These are primarily decoding and generation-stability pathologies rather than reasoning defects. Token-level corruption in LLAMA-3.3-70B on long structured prompts (e.g., bundle-pricing or multi-constraint algebra items) suggests entropy collapse under high-complexity contexts. Phrase-level loops in CLAUDE-HAIKU-4.5 (e.g., item `d940bd577e104035`: "the 1-UIP is $(\neg Z \vee \neg A \vee \neg B)$ is wrong, the 1-UIP is $(\neg Z \vee \neg A \vee \neg B)$ is wrong…") indicate a failure mode where the sampler remains trapped in a locally high-probability repetition cycle after the model has internally lost a stable continuation path. Mid-step truncation, in contrast, reflects hard token-budget constraints during aggressive enumerations, where the output is structurally correct but cut off before completion.

Subcategories.

E8.1 Token-level garble. Concentrated in LLAMA-3.3-70B, with 24 visible instances primarily on B1/B2 items involving multilingual interference and escape-character corruption.

E8.2 Phrase-level loop. Characterized by repetitive or cyclic re-generation of partial phrases without progression toward a final answer. This failure is most prevalent in CLAUDE-HAIKU-4.5 (77 visible cases) and also appears in LLAMA-3.3-70B (33 cases), typically manifesting as repeated restatements of intermediate reasoning steps.

E8.3 Abandoned recalculation. The model initiates a corrective step (e.g., "Wait, let me recalculate") but fails to complete the recomputation or output a final result. This is strongly model-specific to CLAUDE-HAIKU-4.5, with 26 visible cases, and reflects incomplete recovery after self-detected inconsistencies.

E8.4 Empty / null output. Rare failure mode in which the model produces either an empty response or a structurally null completion. Observed sporadically in both CLAUDE-HAIKU-4.5 and LLAMA-3.3-70B, typically under high-context or multi-step constraint-heavy items.

I.10 OVERALL OBSERVATIONS

Reasoning-frontier vs. mid-tier behaviour diverges qualitatively. The two frontier models in our evaluation, GPT-5 and QWEN3.6-PLUS, predominantly fail along E5 (invariance brittleness) and E6 (lexical selection). In other words, their failures are largely *post-reasoning distortions*: they generally reach the correct solution in their native representation, but lose accuracy under surface-form changes or fine-grained lexical constraints. By contrast, mid-tier models such as CLAUDE-HAIKU-4.5 exhibit a different failure profile dominated by E1, E4, and E8, where errors emerge earlier in the reasoning trajectory. These include premature constraint loss (E1), unstable multi-step coherence (E4), and generation-level breakdowns such as recalc-abandon and looping (E8.4). Lower-tier models (GEMMA-4-26B, LLAMA-3.3-70B, DEEPSEEK-V3.1, OLMO-3.1-32B) are primarily governed by E1 and E3 failures, indicating that they often fail at constraint integration and basic computation before higher-order reasoning effects become visible.

The most consistent invariance break is abstraction shift. Across all evaluated models, *Abstraction Level Shift* remains the most systematically damaging transformation, with breakage rates consistently in the 11–21% range. This exceeds all other invariance axes and holds even for top-performing systems. The consistency of this effect provides strong evidence that the benchmark is not probing memorization artifacts, but rather genuine representational robustness: models that correctly solve a problem in one representational form frequently fail when the same underlying structure is expressed at a different level of abstraction.

The most informative single test is the bundle-pricing problem under entity relabeling. A canonical 3-by-3 linear system with an embedded conditional discount, when combined with entity relabeling between `base` and `inv` variants, jointly stresses multiple failure modes simultaneously. This single sub-type (B1/B2/B4) activates E1.1 (dropped constraints), E2.1 (variable re-binding), E3.1 (linear-system computation error), E4.2 (rule sequencing failure), and E5.2 (entity-relabeling brittleness). Empirically, performance on this task cluster correlates nearly monotonically with overall model ranking, making it one of the strongest single predictors of aggregate invariance robustness in the benchmark.

I.11 ROOT CAUSE ANALYSIS AND MITIGATION STRATEGIES

Across the eight error families identified in INVARIANTBENCH, we observe that failures are not isolated artifacts but emerge from a small set of recurring systemic weaknesses: (i) fragile input representation binding, (ii) weak cross-form consistency, (iii) limited multi-step state persistence, (iv) over-reliance on surface priors, and (v) decoding instability under structured constraints. Below we outline targeted mitigation strategies that directly correspond to these failure modes, rather than treating errors as independent phenomena.

RC1. Explicit invariant training across surface forms. A primary driver of E5 (invariance brittleness) is the lack of exposure to semantically identical but structurally diverse inputs. Training pipelines should enforce multi-view supervision, where each problem is paired with all invariant transformations during both supervised and reinforcement phases. This reduces reliance on surface anchors and encourages representation-level encoding. Empirically, this directly targets abstraction shift and entity-relabeling failures. It is most effective when invariants are treated as first-class training objects rather than augmentation artifacts.

RC2. Constraint-aware attention regularization. E1 failures arise from dropped or ignored constraints. A mitigation is to explicitly regularize attention mass toward constraint-bearing tokens using auxiliary supervision signals or gradient-based attribution alignment. This ensures that numerically or logically critical modifiers are not diluted in long contexts. Such mechanisms reduce long-context thinning and improve robustness in multi-constraint algebraic and logic tasks. This is especially relevant for E1.1 and E1.2 subcategories.

RC3. Structured state tracking for multi-step reasoning. E4 failures indicate breakdowns in global state persistence. Introducing an explicit scratchpad or latent state memory that is updated deterministically at each reasoning step can mitigate state-tracking loss. Unlike

free-form chain-of-thought, structured intermediate representations enforce consistency checks across steps. This reduces errors in graph traversal, conditional logic, and recursive dependency tasks. It directly addresses E4.1 and E4.4 failure modes.

RC4. Decoding-time constraint enforcement. Many E7 and E8 failures are not reasoning errors but decoding violations. Introducing constrained decoding or post-generation validation layers can enforce format compliance, completeness, and termination conditions. For example, verifying that all required fields are present or that numerical outputs satisfy constraints before finalization. This reduces format drift, empty outputs, and mid-chain truncation without modifying the underlying model.

RC5. Adversarial paraphrase and relabel training. To address E5.1–E5.3, models should be trained on adversarially generated paraphrases, including entity renaming, variable swaps, and structural reordering. This forces abstraction over symbol identity and reduces role-by-position binding. The key objective is to ensure that semantic roles are grounded in relational structure rather than lexical position. This approach is particularly effective for linear systems and logic-based tasks.

RC6. Dual-path verification mechanisms. E3 and E4 failures often persist because intermediate verification is ignored. A dual-path architecture, where one branch performs reasoning and another independently verifies intermediate results, can reduce arithmetic and logical drift. Disagreement triggers recomputation rather than continuation. This mechanism directly targets system-of-equations errors and combinatorial miscounting.

RC7. Calibration of uncertainty-aware abstention. E7.5 and parts of E6 indicate overconfident responses under ambiguity. Introducing calibrated uncertainty estimation allows models to abstain or request clarification rather than hallucinate or overcommit. This reduces incorrect categorical selection and improper completion of underdetermined tasks. Importantly, calibration should be task-conditioned rather than global to avoid excessive refusal rates.

RC8. Token-level entropy stabilization. E8.1 and E8.2 suggest instability in decoding dynamics under long structured inputs. Techniques such as entropy penalties, repetition suppression, or nucleus sampling scheduling can reduce loop formation and token garbling. Stabilizing decoding trajectories prevents collapse into repetitive or noisy sequences. This is particularly important for long reasoning chains in constrained combinatorial tasks.

RC9. Canonical form normalization layer. To reduce E2 (translation failures), outputs should be mapped through a canonicalization layer that converts semantically equivalent expressions into a standard form before evaluation. This reduces false negatives caused by predicate ordering, notation drift, or equivalent logical reformulations. It aligns model outputs with rubric expectations without penalizing valid alternative representations.

RC10. Hierarchical difficulty-aware prompting. Many failures arise when models misallocate effort across reasoning steps. Hierarchical prompting that explicitly separates coarse reasoning from fine computation can reduce overload in multi-constraint problems. This structure helps prevent premature commitment to incorrect intermediate conclusions and reduces cascading errors in E4-type chains.

RC11. Multi-view consistency training across invariance axes. Rather than training on isolated transformations, models should jointly optimize consistency across all invariance families (symbolic, structural, abstraction). This encourages global invariance manifolds in representation space rather than axis-specific robustness. It directly addresses the systematic degradation observed in abstraction shift and encoding swap scenarios.

RC12. Post-hoc reasoning trace alignment. Finally, aligning intermediate reasoning traces with final answers using supervision or preference optimization reduces self-contradiction. This discourages "correct-then-abandon" behaviors seen in E4.4 and improves coherence between verification steps and final outputs. It ensures that once a correct intermediate state is reached, it is preserved through to finalization.

Collectively, these interventions suggest that robustness in INVARIANTBENCH is not a single capability but an interaction between representation stability, constraint preservation, and decoding control. Improvements in any one dimension are insufficient unless accompanied by cross-axis consistency enforcement.

J Details on Improving Invariant Reasoning Capabilities

To enhance reasoning performance on our diverse benchmark, we explored SFT and GRPO (RL).

J.1 Supervised Fine-Tuning (SFT)

Setup and Implementation. We fine-tuned Gemma-3-1B-IT using the TRL library with default training hyperparameters unless otherwise noted. Training followed a parameter-efficient LoRA protocol applied to attention and feed-forward projection layers, with the model loaded in 4-bit precision to accommodate GPU memory constraints. We constructed two training regimes to disentangle memorization from generalization. In the *All Invariants* setting, a stratified 40:60 train–test split is applied across all task sub-types, and all four surface forms (`base`, `inv1`, `inv2`, `inv3`) are included during training. In the more constrained *Base + Inv1* setting, training uses only `base` and `inv1` variants, and evaluation is additionally conducted on the held-out `inv2` and `inv3` forms, which the model has never seen, providing a clean probe of whether fine-tuning induces genuine cross-form invariance rather than variant-specific pattern fitting. All models are trained for five epochs with accuracy on each invariant form tracked independently at every epoch checkpoint.

SFT Results and Analysis. The learning curves in Figure 3 reveal a stark divergence between the two training regimes. Under *All Invariants* training, performance rises sharply and uniformly across all four surface forms, climbing from a shared baseline of approximately 5.8% to 54.5% (Base), 54.4% (Inv1), 54.2% (Inv2), and 53.8% (Inv3) by epoch 5. This near-identical convergence pattern is consistent with memorization of problem content through cross-variant leakage rather than internalization of invariant structure. The tight alignment of all four curves provides strong evidence that the model is exploiting the statistical overlap between training variants and test instances, and that the observed accuracy gains do not necessarily reflect a generalized capacity for invariance-robust reasoning.

The *Base and Inv1* setting exposes this limitation directly. While Base and Inv1 performance improves substantially, reaching 42.4% and 42.5%, respectively, by epoch 5, accuracy on the unseen forms Inv2 and Inv3 remains critically low throughout training, plateauing at only 17.2% and 14.2%. This gap of approximately 25 percentage points between seen and unseen invariants confirms that SFT does not robustly transfer reasoning patterns across surface reformulations: the model learns a form-specific mapping that fails to generalize to logically equivalent but syntactically novel variants. The findings collectively indicate that standard SFT, even when applied to diverse invariant forms, primarily induces task-surface memorization rather than the representational abstraction required for invariance-robust reasoning. Full per-epoch accuracy breakdowns are provided in Table 12.

Table 12: SFT performance across epochs under full and partial invariant training settings.

Epoch	Setup	Base	Inv1	Inv2	Inv3
0	all invariants	5.78	5.44	6.56	5.35
1	all invariants	37.10	39.34	39.95	39.52
2	all invariants	50.04	47.89	49.70	51.86
3	all invariants	50.73	48.49	50.47	51.16
4	all invariants	53.93	53.06	51.86	53.06
5	all invariants	54.53	54.36	54.18	53.84
0	Base and Inv1	5.78	5.44	6.56	6.13
1	Base and Inv1	23.38	27.96	7.94	6.56
2	Base and Inv1	37.53	34.60	11.65	14.06
3	Base and Inv1	39.43	36.41	10.44	17.69
4	Base and Inv1	42.54	41.33	14.75	16.13
5	Base and Inv1	42.36	42.45	17.17	14.24

J.2 Reinforcement Learning via GRPO

Setup and Implementation. We trained Gemma-3-1B-IT using Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as implemented in TRL, with default hyperparameters. The reward signal is binary correctness evaluated against the ground-truth answer for each invariant variant, and the

model generates multiple rollouts per problem to estimate group-relative advantages. As with SFT, we train under two regimes: All Invariants (all four surface forms available during training) and Base + Inv1 (training restricted to `base` and `inv1`, with `inv2` and `inv3` held out entirely for evaluation). The same stratified 40:60 split across task sub-types is used in both settings, ensuring that SFT and GRPO results are directly comparable. Training proceeds for five epochs, with per-invariant accuracy tracked at each checkpoint.

GRPO Results and Analysis. GRPO training exhibits a different trajectory from SFT across both regimes, as shown in Figure 3. Under *All Invariants*, accuracy increases steadily but remains lower overall, reaching 32.44% (Base), 32.70% (Inv1), 27.78% (Inv2), and 29.08% (Inv3) at epoch 5. Unlike the tightly clustered SFT curves, Inv2 and Inv3 lag behind Base and Inv1 throughout training, with only partial convergence, indicating gradual and incomplete cross-form generalization.
The generalization effect of GRPO is more visible in the *Base and Inv1* regime. While SFT leaves Inv2 and Inv3 at 17.17% and 14.24%, respectively, GRPO improves them to 19.67% and 20.88% by epoch 5, corresponding to gains of 2.50 and 6.64 percentage points. Base and Inv1 reach 30.46% and 32.01%, respectively, substantially lower than the corresponding SFT performance of 42.36% and 42.45% on the seen variants. These results indicate that reward-driven optimization provides a modest improvement in transfer to unseen forms, particularly for Inv3, but does not substantially close the invariance gap. Full per-epoch results are reported in Table 13.

Table 13: GRPO (RL) performance across epochs under full and partial invariant training settings.

Epoch	Setup	Base	Inv1	Inv2	Inv3
0	all invariants	6.13	5.26	6.38	6.30
1	all invariants	11.13	8.80	11.82	8.54
2	all invariants	18.64	19.67	16.74	14.58
3	all invariants	21.40	21.14	24.50	22.35
4	all invariants	28.82	28.39	30.72	29.59
5	all invariants	32.44	32.70	27.78	29.08
0	Base and Inv1	5.95	5.87	6.21	5.61
1	Base and Inv1	9.06	10.70	5.95	6.99
2	Base and Inv1	16.65	22.43	11.73	6.99
3	Base and Inv1	22.86	26.23	15.01	10.61
4	Base and Inv1	26.66	30.03	17.34	15.62
5	Base and Inv1	30.46	32.01	19.67	20.88

J.3 On SFT/GRPO Diagnostic Scope

A potential limitation is that the SFT and GRPO experiments are conducted on a single model (Gemma-3-1B-IT) without multiple seeds or cross-architecture replication. We emphasize that these experiments are intended as *diagnostic interventions* rather than definitive claims about training paradigms at scale. The goal is to isolate whether standard supervised imitation versus reward-driven optimization exhibits qualitatively different behavior under controlled invariance settings. As such, the focus is on identifying directional effects (e.g., memorization vs. generalization patterns) rather than establishing statistically robust performance trends across models.

Importantly, the observed divergence between SFT and GRPO is large and structurally consistent across training regimes within this controlled setup, making it unlikely to be purely attributable to noise or initialization variance. However, we do not claim that these findings universally generalize across model sizes, architectures, or training seeds. Extending this analysis to larger models (e.g., 7B, 8B, or beyond), multiple random initializations, and alternative architectures remains an important direction for future work.

We therefore position these results as a proof-of-concept demonstrating that training objectives can materially affect invariance behavior, rather than as a conclusive comparison of optimization strategies. A comprehensive study across scales and settings is left for subsequent investigation.

K DISCUSSION

K.1 OVERALL DISCUSSION

✓ **High base accuracy can hide representational dependence.** Our results show that accuracy on a fixed problem formulation is an incomplete measure of reasoning robustness. Across models, performance that is strong on the original instances often drops when the same problems are expressed through semantics-preserving transformations. Because the underlying problem, answer, and reasoning requirements remain unchanged, these drops cannot be attributed to increased task difficulty alone. Instead, they reveal sensitivity to how the problem is expressed. This distinction is important: a model may identify the correct solution when a problem matches a familiar formulation yet fail to recover the same solution after a harmless change in wording, structure, representation, or abstraction. InvariantBench therefore measures a property that conventional accuracy does not isolate: whether a reasoning capability transfers across equivalent views of the same underlying problem.

✓ **Abstraction shifts expose a deeper generalization failure.** Abstraction-level transformations are the most consistently damaging invariance class, producing failure rates of 11–21% even among the strongest models. These transformations preserve the underlying reasoning structure while changing its level of description, for example by replacing a concrete instance with a parameterized formulation. The pattern suggests that models do not consistently separate the reasoning rule from the particular form in which that rule is expressed. A model can therefore succeed when the relevant structure appears at a familiar level of granularity but fail when the same structure is presented more generally or concretely. This makes abstraction shifts especially informative: they test whether a model has learned a reusable reasoning relation rather than a formulation-specific solution pattern.

✓ **Structural changes reveal reliance on positional cues.** The substantial degradation caused by premise reordering provides direct evidence that semantic equivalence alone does not guarantee equivalent model behavior. Reordering independent premises changes neither their content nor their logical relations, yet accuracy drops by as much as 19.6 points in strong models. Similar sensitivity appears when information is moved across textual, symbolic, and tabular representations. These failures indicate that models can attach importance to position, ordering, or presentation format even when these properties are irrelevant to the underlying task. The issue is therefore not limited to paraphrasing: it extends to the organization through which the same constraints are presented. Robust reasoning should recover the relevant relations independent of such presentation choices.

✓ **Scaling changes where models fail, not only how often.** The error analysis shows a qualitative shift in failure modes with model scale. Smaller models frequently fail during premise interpretation or basic computation, so errors arise before invariance can be tested meaningfully. Larger models more often parse and solve the original problem correctly but lose consistency after transformation, with errors concentrated in invariance preservation and fine-grained lexical-semantic decisions. Scaling thus appears to move the dominant bottleneck from *solving the instance* to *maintaining the solution across equivalent instances*. This distinction helps explain why higher base accuracy does not translate proportionally into higher invariant consistency: once basic reasoning becomes sufficiently strong, representational stability becomes the limiting factor.

✓ **SFT can improve accuracy without learning broad invariance.** Our training experiments further show that improving performance on transformed examples is not equivalent to learning invariance. SFT produces substantial gains when the relevant variants are included during training, but when exposure is restricted, performance on unseen invariants retains a gap of roughly 25 percentage points. This indicates limited transfer from observed transformations to unseen but semantically related ones. In other words, exposing a model to more forms can teach it to handle those forms without necessarily making its reasoning independent of form. The distinction is important for training: memorizing a collection of transformation patterns is easier than learning the underlying equivalence that connects them.

✓ **GRPO improves transfer, but does not close the gap.** GRPO shows a stronger transfer profile than SFT under partial invariant exposure, more than doubling performance on unseen variants relative to the supervised baseline while retaining competitive performance on seen variants. This result suggests that reward-based optimization can encourage behavior that generalizes beyond the exact transformations observed during training. However, substantial residual inconsistency remains. Thus, reinforcement learning appears to improve the *transfer* of invariant behavior rather than fully establishing it. The result also suggests that invariance may benefit from objectives that reward consistency across alternative formulations, rather than relying only on correctness for individual examples.

✓ **Humans remain substantially more consistent across equivalent formulations.** Human evaluators maintain more than 98% consistency across the tested transformation types, whereas even the strongest models remain below approximately 48% full consistency. This gap is notable because the transformations do not change the intended reasoning problem. Human performance therefore provides evidence that the transformations themselves are generally understandable and that the observed model failures are not simply an unavoidable property of the tasks. At the same time, the comparison should be interpreted as a behavioral reference rather than evidence about the internal mechanisms of human reasoning. The central finding is narrower and more testable: humans are far less sensitive than current models to the representational changes introduced by our transformations.
✓ **Invariance provides a stronger test of memorization-based performance.** Semantics-preserving transformations also make conventional contamination concerns less decisive. A model that has memorized a seed problem may still fail when the same problem is presented through an unseen equivalent formulation. Consequently, performance on transformed variants probes whether knowledge of a particular instance transfers beyond its observed surface form. The resulting base-to-invariant gap is therefore useful as a complementary measure of generalization. It does not eliminate contamination concerns—models may still have encountered equivalent reasoning patterns—but it makes simple instance memorization less sufficient to explain strong performance.

K.2 Future Work Directions

■ **Training for joint rather than isolated invariance.** Our SFT and GRPO results suggest that improvements do not reliably transfer across invariance axes. A natural next step is to train models on multiple equivalent views of each problem while explicitly rewarding agreement in both final answers and intermediate reasoning states. Rather than treating each transformation as an independent augmentation, training could define a consistency objective over the set of valid formulations of a seed problem. This would encourage the model to preserve the underlying reasoning relation while changing its surface realization. An important challenge is to prevent the objective from becoming a collection of transformation-specific shortcuts; evaluation should therefore include held-out transformation types and compositions.
■ **Preserving constraints throughout reasoning.** The failure patterns suggest that some invariance errors begin when critical constraints are weakened or misrepresented early in processing. Future models could explicitly track quantities, conditions, quantifiers, bounds, and logical dependencies that determine the solution. Such mechanisms need not impose a particular reasoning format; their purpose would be to preserve task-defining information as wording and structure change. This may be especially useful for multi-constraint problems, where losing a single condition early can produce a fluent but systematically incorrect solution.
■ **Persistent intermediate states for multi-step reasoning.** Longer reasoning chains introduce another source of instability: an early correct deduction may be ignored, overwritten, or contradicted later. Explicit state tracking or structured working memory could preserve intermediate conclusions and allow later steps to check against them. This is particularly relevant to planning, combinatorial reasoning, and other tasks where correctness depends on maintaining several constraints over many steps. Such mechanisms could also make invariance failures easier to localize by distinguishing errors in initial interpretation from errors introduced during subsequent reasoning.
■ **Independent verification for arithmetic and logical consistency.** Some failures arise from incorrect intermediate computations or logical deductions that are not corrected during generation. A separate verification process could check intermediate results, constraint satisfaction, and logical consequences before they are propagated. When a contradiction is detected, the system could recompute the affected step rather than continuing from an invalid state. This would complement invariance training by addressing a different failure source: a model may represent an equivalent problem correctly yet still lose consistency because an intermediate computation is wrong.
■ **Representations that survive abstraction changes.** The strong effect of abstraction-level transformations motivates methods that explicitly connect concrete and abstract formulations of the same reasoning pattern. Training could pair instantiated problems with parameterized versions and require consistent solutions across the pair. More broadly, curricula could vary the level of detail while keeping the underlying relation fixed. Success on such pairs would provide a direct test of whether the model has separated the reasoning rule from the particular values, entities, or wording used to express it.
■ **Breaking position and identity shortcuts.** Premise reordering and entity relabeling suggest that models can associate semantic roles with where an item appears or which tokens name it. Training

can directly challenge these shortcuts through systematic permutation, relabeling, and representation changes. The goal is not to make models insensitive to all order information—order can itself carry meaning—but to distinguish meaningful order from arbitrary presentation order. Evaluation should therefore target transformations for which the relevant relational structure is provably unchanged.

■ **Improving fine-grained semantic precision.** Lexical-semantic failures point to a complementary limitation: a small wording change can alter truth conditions even when two expressions appear nearly synonymous. Future work should focus on distinctions involving negation, quantifiers, scope, and other operators whose interpretation determines the answer. Contrastive training with minimally different formulations could help models learn when a lexical substitution preserves meaning and when it changes the problem. This is particularly important for invariance evaluation because a valid transformation must change presentation without accidentally changing semantics.

■ **Separating semantic correctness from output format.** Finally, evaluation itself should distinguish reasoning errors from harmless differences in answer representation. Canonical normalization of mathematical, symbolic, and structured outputs can map equivalent answers to a common form before scoring. This reduces false failures caused by formatting preferences and allows invariance metrics to focus on the property they are intended to measure. Such normalization is especially important when comparing models that express equivalent solutions using different but valid output conventions.

K.3 Broader Impacts

INVARIANTBENCH provides a diagnostic for whether reasoning systems remain reliable when the same problem is expressed in different but semantically equivalent ways. This can improve evaluation and model selection for applications where inputs naturally vary in wording, structure, or format, including education, scientific reasoning, and decision support. At the same time, optimizing specifically for benchmark transformations may encourage "invariance gaming" without improving robustness to unseen forms. The benchmark's failure analysis could also reveal exploitable weaknesses in deployed models. We therefore position INVARIANTBENCH as a complementary evaluation tool rather than a standalone optimization target, and recommend combining it with broader robustness and safety evaluations.

L Representative Benchmark Samples with Model Answers and Evaluations

A2. Set Algebra + Cardinality with Constraints

ID: 83994f0c8a9643d8 | Base

Question: Let S = {x in N | 1 <= x <= 100000, x mod 41 = 23, x is prime}. How many elements are in S?

Answer: 17

Claude Haiku 4.5: By the prime number theorem for arithmetic progressions and computational verification, the actual count is: 80.

DeepSeek-V3.1: 108.

Gemma-4-25B-A4B-I: 237.

GPT-5.2: 240.

Llama-3.3-70B-I: 0.

Olmo-3.1-32B-I: 0.

Qwen3.6-Plus: 240.

A3. Conditional Override Transformation Systems

ID: 8d3aeea181294a0d | Base

Question: At a music shop, Bundle A contains 2 strings, 1 reeds, and 3 sticks for 86; Bundle B contains 1 strings, 3 reeds, and 2 sticks for 82; Bundle C contains 3 strings, 2 reeds, and 1 sticks for 90. If an order includes at least 6 reeds, each such unit is discounted by 5. One order contains 5 strings, 6 reeds, and 5 sticks. What is total charge?

Answer: 198

Claude Haiku 4.5: 198.

DeepSeek-V3.1: 337.

Gemma-4-25B-A4B-I: 261.

GPT-5.2: 198.

Llama-3.3-70B-I: 134.

Olmo-3.1-32B-I: 244.

Qwen3.6-Plus: 198.

Figure 5: Example from the benchmark and evaluation results. Red denotes incorrect and Green denotes correct answer marked by the judge model.

More such examples are provided in the separate supplementary material PDF.

A4. Constraint Satisfaction in Propositional Logic

ID: 7e19ebe57e9d4eee | Base

Question: Four switches connect to a motor: Q, R, S, T. Rules:
1. **If Q is ON, then R is OFF (Q ⇒ ¬R)**
2. **If R is OFF, then S is ON (¬R ⇒ S)**
3. **T is ON if and only if Q is ON (T ⇔ Q)**
4. **S is OFF (¬S)**

Find the true state of the switches.

Answer: Q=OFF, R=ON, S=OFF, T=OFF

Claude Haiku 4.5: Q: OFF, R: ON, S: OFF, T: OFF.

DeepSeek-V3.1: Q off R on S off T off.

Gemma-4-25B-A4B-I: Q is OFF, R is ON, S is OFF, T is OFF.

GPT-5.2: Q OFF R ON S OFF T OFF.

Llama-3.3-70B-I: Q is OFF R is ON S is OFF T is OFF.

Olmo-3.1-32B-I: Q OFF R ON S OFF T OFF.

Qwen3.6-Plus: Q and S and T are off and R is on.

A5. Inductive Rule + Formal Deduction

ID: 476db78e286a4519 | inv1

Question: In Boolean logic, the rule p ∧ (p ∨ q) is known as the "Absorption Law." If you simplify this rule, what does it exactly equal?

Answer: p

Claude Haiku 4.5: p.

DeepSeek-V3.1: The same as the first thing being true.

Gemma-4-25B-A4B-I: It is just the first thing being true.

GPT-5.2: Just the first true thing A.

Llama-3.3-70B-I: It is true.

Olmo-3.1-32B-I: True.

Qwen3.6-Plus: That something is true.

Figure 6: Example from the benchmark and evaluation results. Red denotes incorrect and Green denotes correct answer marked by the judge model.

B2. Optimization over Coupled Algebraic Equations

ID: ae7024cda8854d76 | inv1

Question: Let the production rates of three workers be x, y, and z. The combined rates are x + y = 32, y + z = 28, and x + z = 24. Calculate the total production if worker x works 3 hours, worker y works 2 hours at a rate reduced by 3 units/hour, and worker z works 3 hours.

Answer: 102

Claude Haiku 4.5: 102.

DeepSeek-V3.1: 84.

Gemma-4-25B-A4B-I: 75.

GPT-5.2: 102.

Llama-3.3-70B-I: 102.

Olmo-3.1-32B-I: 63.

Qwen3.6-Plus: 102.

B3. Probabilistic Reasoning with Conditional Dependence

ID: 07710e3bde024aae | inv2

Question: Evaluate the following conditions for a competitive event: Format: strictly single-elimination; Total participants: exactly 64 players; Skill level: perfectly equal (every match is a 50/50 coin flip); Seeding: initial bracket placements are assigned completely at random. Based strictly on these parameters, what is the exact mathematical probability that Player Alpha and Player Beta will face each other in the final championship match?

Answer: 1/2,016

Claude Haiku 4.5: 1/32.

DeepSeek-V3.1: 1/63.

Gemma-4-25B-A4B-I: 1/1024.

GPT-5.2: Exact probability Player Alpha and Player Beta meet in final is 1/2016.

Llama-3.3-70B-I: 1/63.

Olmo-3.1-32B-I: 1/63.

Qwen3.6-Plus: 1/2016.

Figure 7: Example from the benchmark and evaluation results. Red denotes incorrect and Green denotes correct answer marked by the judge model.