LoLeakBench: Measuring Memorization via Isomorphic Variants in Linguistics Olympiad Benchmarks Across Competitions

Denis Siminiuc
Massachusetts Institute of Technology
[TODO: email]

January 20, 2026

Abstract

[TODO: 2–5 sentences: what problem, what you built, what you found.]

We present LoLeakBench, a reproducible benchmark suite built from Linguistics Olympiad problems (IOL, NACLO, UKLO, AUSLO) with a focus on *leakage*: the possibility that large language models (LLMs) solve public problems by memorization rather than rule induction. Our key contribution is an evaluation protocol based on *isomorphic variants*—surface-form transformations that preserve the underlying logical structure of a puzzle. We define a memorization gap metric, $\mathsf{MemGap}=$ Acc(Orig) - Acc(Iso), and report results for a set of open-weight models runnable on consumer hardware.

1 Introduction

Linguistics Olympiad problems are designed to be self-contained puzzles: given a small set of examples in an unfamiliar language, solvers must induce rules and answer questions (e.g., matching, translation, segmentation). Recent work has proposed benchmarks derived from the International Linguistics Olympiad (IOL) to evaluate LLM reasoning in a knowledge-neutral setting [?].

However, knowledge-neutral *task structure* does not guarantee a knowledge-neutral *evaluation*. Many Olympiad problems and official solutions are publicly available online. As a result, strong performance may partially reflect exposure to the exact problems or near-duplicates during pretraining ("leakage"), rather than robust rule induction.

Goal. We aim to quantify leakage in Linguistics Olympiad benchmarks by comparing performance on original problems to performance on *isomorphic variants* that preserve the underlying reasoning structure while perturbing surface form.

Contributions.

- **Cross-competition dataset.** We curate a suite of auto-gradable subproblems from IOL, NACLO, UKLO, and AUSLO in a unified schema.
- **Isomorphic-variant protocol.** We define a family of transformations that preserve correctness while breaking exact-match memorization.
- **Leakage metric.** We propose the memorization gap $\mathsf{MemGap}=$ Acc(Orig)-Acc(Iso) and analyze it by task type, year, and competition.

- **Open reproducibility.** We provide a runner and cached outputs for open-weight models that can be executed locally (e.g., on an Apple M-series laptop) with deterministic settings.

Paper roadmap. Section 3 describes dataset construction; Section 4 defines isomorphic variants; Section 5 details metrics and grading; Section 6 describes models and protocols; Section 7 reports findings.

2 Related Work

Reasoning benchmarks. A large literature evaluates LLMs on reasoning problems; many datasets risk conflating reasoning with memorized knowledge. [TODO: Add 2–4 citations relevant to general reasoning benchmarks.]

Linguistics Olympiad benchmarks. IOLBench [**?**] digitizes IOL problems and evaluates frontier LLMs on multiple answer types. Our work complements this line by (i) extending to additional competitions and (ii) explicitly measuring leakage via isomorphic variants.

Data contamination and memorization. Prior work has studied training-data contamination in benchmark evaluations. [TODO: Add 2–3 citations on contamination/memorization analysis.]

3 Dataset Construction

Sources. We collect problems from publicly available archives of IOL, NACLO, UKLO, and AUSLO. To enable reliable evaluation, we focus on *auto-gradable* subproblems (matching, multiple-choice, and short structured outputs).

Unified schema. Each item is stored as a JSON record containing: (i) the problem statement and structured example tables, (ii) a set of questions, (iii) canonical gold answers, and (iv) metadata (competition, year, task type, and links). [TODO: Add a short schema figure or example listing in the appendix.]

De-duplication. We remove exact duplicates across sources and perform near-duplicate checks within the dataset. [TODO: Describe method: hashing + fuzzy matching.]

Task types.

- **Matching:** map numbered strings to lettered glosses.
- **Multiple-choice:** choose among options A–D.
- **Short structured output:** constrained strings (e.g., a single word/phrase or JSON mapping).

Train/dev/test. We report results on a held-out test split stratified by competition and year. [TODO: Decide split strategy and add counts.]

4 Isomorphic Variants and Leakage Measurement

Definition. Given an item x with gold answer y, an *isomorphic variant* is a transformation T that produces $x' = T(x)$ such that there exists a deterministic induced mapping T_y with $y' = T_y(y)$ and (x', y') preserves the underlying reasoning structure. In short: the puzzle is "the same," but surface strings change.

Transformations. We use compositions of:

- **Label permutation:** permute the letter labels of multiple-choice or matching lists.
- **Example reordering:** reorder the presentation of examples without changing content.
- **Lexeme renaming:** consistently rename the unknown-language tokens and/or English gloss words via a bijection.
- **Format-preserving noise:** whitespace and punctuation perturbations that do not alter the intended reading.

We avoid transformations that change linguistic difficulty (e.g., adding/removing distractors).

Memorization gap. We define

$$\mathsf{MemGap} \;=\; \mathrm{Acc}(\mathsf{Orig}) - \mathrm{Acc}(\mathsf{Iso}),$$

where $\mathrm{Acc}(\mathsf{Orig})$ is accuracy on original items and $\mathrm{Acc}(\mathsf{Iso})$ is accuracy on isomorphic variants. A large positive MemGapsuggests sensitivity to surface form consistent with memorization or brittle pattern matching.

Variant generation protocol. For each test item, we generate k variants (default $k = 3$) with fixed random seeds for reproducibility. [TODO: Add details: which spans are variantable and how you sample bijections.]

5 Evaluation

Deterministic grading. We design prompts that force machine-checkable outputs: JSON for matching, a single letter for multiple-choice, and normalized strings for short outputs. We grade with strict parsers plus lightweight normalization (case-folding, whitespace normalization).

Metrics. We report:

- Accuracy by task type and overall,
- Memorization gap MemGapoverall and by task type,
- Accuracy by competition and by year bins (to probe temporal effects).

Robustness checks. We include protocol-level controls: temperature $= 0$, fixed max tokens, fixed context window, and cached outputs. [TODO: Add prompt ablations if time permits.]

6 Models and Experimental Protocol

Model set. We benchmark a suite of open-weight instruction-tuned LLMs runnable locally. [TODO: Fill in final list: e.g., Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, Mistral-7B, Gemma-2-9B, Phi-3-mini.]

Table 1: Accuracy on original items (Orig), isomorphic variants (Iso), and memorization gap (MemGap). [TODO: Fill with real numbers.]

Model	Acc(Orig)	Acc(Iso)	MemGap
[TODO: Model A]	[TODO: .00]	[TODO: .00]	[TODO: .00]
[TODO: Model B]	[TODO: .00]	[TODO: .00]	[TODO: .00]

Inference stack. We run models via a local inference server (e.g., Ollama/llama.cpp), using identical prompts and decoding settings. [TODO: List exact versions and quantization formats.]

Prompting. We use a single prompt template per task type. The prompt specifies output format strictly ("JSON only", "single letter only"). We do not provide exemplars unless explicitly stated (zero-shot by default).

Caching. All raw model outputs are cached and re-gradable to support reproducibility.

7 Results

Main finding. [TODO: One paragraph: headline gap sizes + what it suggests.]

By task type. [TODO: Add a small table or figure showing gaps for matching vs MCQ vs short-output.]

By competition and year. [TODO: Summarize whether older problems exhibit larger gaps (possible contamination proxy).]

8 Discussion

Interpreting the memorization gap. A non-zero MemGapcan arise from memorization, brittle heuristics, or genuine sensitivity to superficial formatting. We interpret MemGapas a diagnostic rather than a definitive proof of training-data exposure.

What isomorphic variants do and do not test. Variants aim to preserve structure while breaking exact surface matches. They do not fully control for changes in linguistic difficulty introduced by renaming or reordering. [TODO: Add empirical sanity checks: human solvability unchanged, etc.]

Implications for benchmark design. We recommend reporting variant performance alongside original performance for public, web-available benchmarks.

9 Limitations and Ethics

Licensing. We distribute only derived, structured instances and links to source materials where appropriate. [TODO: Confirm permissions/terms for each competition.]

Contamination claims. Our results should not be interpreted as proof of explicit memorization of any particular source. We present aggregate evidence that surface-form perturbations can substantially change outcomes.

Responsible release. We include reproducible scripts and cached outputs. We avoid releasing any private test sets.

10 Reproducibility

We release:

- the dataset in a unified JSON schema,
- deterministic variant-generation code and seeds,
- an evaluation harness with strict parsing and normalization,
- runner scripts for local inference,
- cached model outputs with hashes.

[TODO: Add exact commit hash, model hashes, and environment details.]

A Appendix

A.1 Example item schema

[TODO: Include a verbatim JSON example (short) showing fields used for variants and grading.]

A.2 Prompt templates

[TODO: Include the exact prompts for matching / MCQ / short-output tasks.]