VerifyWise LLM Leaderboard Methodology

Version 2.0 — January 2026

---

1. Overview

The VerifyWise LLM Leaderboard provides a comprehensive evaluation of large language models (LLMs designed for enterprise use. The leaderboard intentionally combines **practical, real-world evaluation** with **widely recognized academic benchmarks** to give a balanced and trustworthy view of model performance.

The leaderboard consists of two primary components:

1. **VerifyWise Application Score**
   A proprietary metric that evaluates real-world enterprise task performance.

2. **Standard Academic Benchmarks**
   Well-established benchmarks from the research community, sourced from LLMStats.

---

2. VerifyWise Application Score

2.1 Purpose

The VerifyWise Application Score measures how effectively LLMs perform in **real-world enterprise scenarios**, rather than isolated academic tasks. The goal is to assess practical usefulness in business contexts such as structured outputs, retrieval-augmented workflows, coding tasks, and agent-based automation.

Unlike traditional benchmarks that focus on narrow capabilities, this score reflects how models behave when deployed in production-like environments.

2.2 Evaluation Methodology

Each model is evaluated on **44 total tasks**, distributed across **five evaluation suites**.

- Each task is graded on a **pass / fail** basis using strict criteria.
- Scores are aggregated at the suite level.
- The final Application Score is calculated as a **weighted average** of the suite scores.

2.3 Evaluation Suites

2.3.1 Instruction Following (25% weight)

12 tasks

This suite evaluates the model's ability to follow complex, multi-step instructions accurately and consistently.

Example tasks include:

- Producing outputs in strict formats (JSON, XML, markdown tables)
- Handling conditional logic (e.g., "If X, then do Y; otherwise do Z")
- Satisfying multiple constraints simultaneously
- Parsing and executing nested or layered instructions

2.3.2 RAG Grounded Question Answering (25% weight)

8 tasks

This suite measures retrieval-augmented generation (RAG) quality, with a focus on factual grounding and hallucination avoidance.

Example tasks include:

- Answering questions using only the provided context
- Correctly citing sources with page or section references
- Explicitly acknowledging missing or insufficient information

- Distinguishing between supported and unsupported claims

2.3.3 Coding Tasks (20% weight)

8 tasks

This suite evaluates programming capabilities across multiple languages and difficulty levels.

Example tasks include:

- Generating executable code from specifications
- Debugging code containing logical or runtime errors
- Explaining complex algorithms in plain language
- Refactoring code to improve readability or performance

2.3.4 Agent Workflows (15% weight)

6 tasks

This suite assesses agentic behavior, including planning, tool usage, and multi-step task execution.

Example tasks include:

- Executing multi-step workflows involving tool calls
- Recovering from errors and retrying appropriately
- Decomposing complex goals into smaller actionable steps
- Maintaining context across long interaction chains

2.3.5 Safety & Policy (15% weight)

10 tasks

This suite evaluates adherence to safety standards and content policies.

Example tasks include:

- Appropriately refusing harmful or unethical requests
- Handling sensitive topics with care and nuance
- Maintaining professional and ethical boundaries
- Avoiding misleading, unsafe, or dangerous outputs

2.4 Scoring Formula

The VerifyWise Application Score is calculated as follows:

Application Score =
(Instruction Following × 0.25) +
(RAG Grounded QA × 0.25) +
(Coding Tasks × 0.20) +
(Agent Workflows × 0.15) +
(Safety & Policy × 0.15)

Each suite score ranges from **0 to 100**.

2.5 Evaluation Details

Evaluator:
VerifyWise Evaluation Pipeline v2.0

Judging Method:
Hybrid approach using human review and GPT-4.1 as an automated judge

Evaluation Period:
January 2026

Reproducibility:
All evaluation prompts, task definitions, and scoring criteria are available in VerifyWise's open-source evaluation suite.

---

3. Standard Academic Benchmarks

To complement the Application Score, the leaderboard includes three widely recognized academic benchmarks. These scores provide additional context and are **not modified or re-scored by VerifyWise**.

All academic benchmark data is sourced from **LLMStats**, an independent aggregator of LLM benchmark results.

3.1 MMLU — Massive Multitask Language Understanding

Description

MMLU evaluates a model's breadth of knowledge and general reasoning ability across a wide range of disciplines.

Key Characteristics

- 57 subjects spanning STEM, humanities, social sciences, and professional domains
- Multiple-choice question format
- Measures general knowledge and problem-solving ability

Domains Covered

- STEM: Mathematics, Physics, Chemistry, Computer Science, Biology
- Humanities: History, Philosophy, Law
- Social Sciences: Psychology, Economics, Sociology
- Professional Fields: Medicine, Accounting, Engineering

Reference

- Hendrycks et al., 2020
- "Measuring Massive Multitask Language Understanding"

- arXiv:2009.03300

Data Source

- LLMStats

3.2 GPQA — Graduate-Level Google-Proof Question Answering

Description

GPQA measures expert-level reasoning in advanced scientific domains. Questions are explicitly designed to be resistant to simple search-based solutions.

Key Characteristics

- 448 multiple-choice questions
- Written by PhD-level domain experts
- Focused on biology, physics, and chemistry
- Designed to test deep reasoning and conceptual understanding

Reference

- Rein et al., 2023
- “GPQA: A Graduate-Level Google-Proof Q&A Benchmark”
- arXiv:2311.12022

Data Source

- LLMStats

3.3 HumanEval — Code Generation Benchmark

Description

HumanEval evaluates the functional correctness of generated code.

Key Characteristics

- 164 programming problems
- Models generate code from function signatures and docstrings
- Performance measured using automated unit tests
- Reported metric is pass@1 (first-attempt correctness)

Reference

- Chen et al., 2021
- “Evaluating Large Language Models Trained on Code”
- arXiv:2107.03374

Data Source

- LLMStats

---

4. Data Sources and Attribution

4.1 VerifyWise Application Score

- Source: VerifyWise internal evaluation pipeline
- Methodology: Proprietary evaluation suite developed by VerifyWise
- Update Policy: Scores are updated as new models are released and evaluated

4.2 Academic Benchmarks

- Source: LLMStats
- Attribution: Aggregated from official model releases, research papers, and verified third-party evaluations
- Accuracy Note: Some scores may be self-reported by model providers

---

5. Interpreting Leaderboard Scores

5.1 Score Ranges

- **90% and above** — Excellent (Top-tier performance)
- **80–89%** — Very Good
- **70–79%** — Good
- **60–69%** — Fair
- **Below 60%** — Limited

5.2 Important Considerations

1. No single score fully captures model capability.
2. The Application Score emphasizes enterprise usability, while academic benchmarks measure specific competencies.
3. Model selection should be guided by task-specific needs.
4. Scores represent a snapshot in time; models and benchmarks evolve.

---

6. Contact and Feedback

For questions, feedback, or methodology discussions:

- Website: **verifywise.ai**
- GitHub: **github.com/verifywise/verifywise**

Document Version: 2.0
Last Updated: January 2026