A Composable Evaluation System for Reproducible Omni-Modal Foundation Model Evaluation

Hodong Lee[1,2], Sanghee Park[1,3], Dohoon Ryu[1], Jungwhan Kim[1], Junyeob Kim[4*], Soyoon Kim[1], Geewook Kim[1,3†]

[1]NAVER Cloud AI, [2]Korea University, [3]KAIST AI, [4]Seoul National University

{hodong.lee, sang.hee.park, dh.ryu, jungwhan.kim, soyoon.kim, gw.kim}@navercorp.com

juny116@europa.snu.ac.kr

Abstract

Building an omni-modal foundation model means evaluating it across text, image, video, and audio. Excellent evaluation toolkits exist for each modality, but their inference engines, prompt conventions, and metric implementations are mutually incompatible, so practitioners end up maintaining separate environments for every toolchain and still struggle to compare results across them. **OmniEvaluator** grew out of this need in our own model development: rather than reimplementing benchmarks, it connects existing inference engines and curated evaluation libraries at a higher level, exposing four inference backends, four evaluation frameworks, and over a thousand benchmarks through a single interface. Every run is recorded as an artifact capturing the full configuration for exact reproduction, and results flow into a shared dashboard for cross-model comparison. A federated mode shares GPU inference servers across concurrent evaluations, and a built-in *verifier*, small enough to run on CPU, gives a score that varies far less across engines and prompts than rule-based scoring under configuration mismatch, matching cost-efficient commercial LLM judges without their recurring API cost. The system, demo video, and dashboard are publicly available.[1]

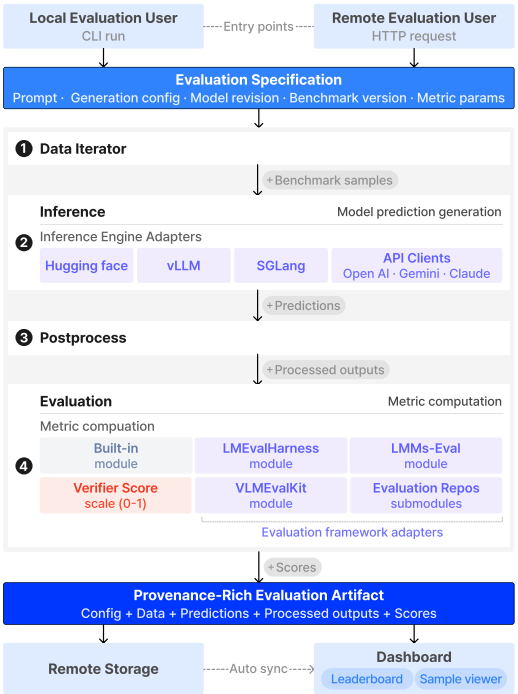


Figure 1: **OmniEvaluator architecture.** Inference and evaluation engines are composed through a unified intermediate schema, enabling modular combination of any supported engine pair. Users can run evaluations locally via CLI or submit requests to a remote evaluation server, both producing provenance-rich evaluation artifacts.

1 Introduction

Foundation models are increasingly *omni-modal*: a single model accepts text, image, video, and audio as input (Gemini Team et al., 2023; OpenAI et al., 2024; Xu et al., 2025a; Fu et al., 2025b; NAVER Cloud HyperCLOVA X Team, 2026). Developing such a model requires evaluating it on benchmarks from every modality it supports. Evaluation infrastructure, however, has not kept pace. Vision–language models are usually evaluated with VLM-specific toolkits (Duan et al., 2024), text models with language-model harnesses (Sutawika et al., 2026), and audio or video capabilities with scripts written for individual benchmarks. Each toolchain has its own prompt conventions, preprocessing steps, and scoring code.

This fragmentation causes two recurring problems for research and production teams alike. First, no single environment covers an omni-modal model. Each toolkit pins its own dependencies; per-toolkit virtual environments avoid version conflicts but still leave every toolkit with its own interface, configuration format, and output layout, all

*Work done while at NAVER Cloud AI.

†Corresponding author.

[1]https://github.com/naver-ai/omni-evaluator

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 738–749
October 24-29, 2026 ©2026 Association for Computational Linguistics

reconciled by hand. In practice, modalities that are hard to set up often simply go unmeasured. Second, scores reported for "the same benchmark" often disagree. Different libraries use different prompt templates, decoding parameters, and evaluator versions (Biderman et al., 2024; Alzahrani et al., 2024), and even metrics with the same name, such as WER (Park et al., 2024) or ANLS (Peer et al., 2024), can hide different normalization rules. As a result, scores are hard to compare across papers, and sometimes even across runs within a team (we quantify this in Table 3). As a further consequence, results become scattered across machines, and GPU resources are split across users rather than pooled.

Our goal is not to build yet another isolated benchmark suite, but to tie together what the community already builds and maintains, so that a single model can be evaluated across every modality it supports, under one configuration, with every score reproducible. We present **OmniEvaluator**, a composable evaluation system that combines existing inference engines and curated evaluation libraries as building blocks (Figure 1). Internally, a shared intermediate schema standardizes the four stages of evaluation: data iteration, inference, post-processing, and metric computation. Connecting N inference engines to M evaluation frameworks directly would require $N \times M$ pairwise integrations. With the schema in between, each engine and each framework needs only one thin adapter, $N + M$ in total, and any engine can then be paired with any framework. Every run also produces a *provenance-rich artifact* recording the full configuration (prompt template, generation parameters, model revision, benchmark version, and metric settings), so a score can be reproduced rather than merely reported.

On top of this pipeline, OmniEvaluator provides:

- **Unified omni-modal evaluation**: one installation and one interface covering four inference backends, four evaluation frameworks, and over a thousand benchmarks across text, image, video, and audio (Table 1)—spanning a substantial portion of the benchmarks reported in the omni-modal tech reports published to date (Table 2)—with any backend able to run any framework's benchmarks. A federated mode further shares inference servers across concurrent evaluations for efficient GPU use (§4.3).

Table 1: **Supported inference and evaluation engines.** Each engine is integrated via a single adapter to a unified schema. **# Bench.** counts the benchmarks accessible per evaluation framework at the time of writing; the built-in engine covers audio and other benchmarks unavailable elsewhere.

Category	Supported Engine	# Bench.
Inference	HuggingFace Transformers (Wolf et al., 2020)	–
	vLLM (Kwon et al., 2023)	–
	SGLang (Zheng et al., 2024)	–
	Off-the-shelf API clients (OpenAI, Google, Anthropic)	–
Evaluation	built-in	182
	`lm-eval-harness` (Sutawika et al., 2026)	1986
	`lmms-eval` (Zhang et al., 2025a)	416
	`VLMEvalKit` (Duan et al., 2024)	375

Table 2: **Benchmark coverage of OmniEvaluator.** Number of benchmarks reported in each representative omni-modal foundation model's tech report that OmniEvaluator supports.

Technical Report	Covered / Total
HyperCLOVAX-SEED-Omni-8B (NAVER Cloud HyperCLOVA X Team, 2026)	28 / 28
MiniCPM-o 4.5 (Cui et al., 2026)	49 / 62
Qwen3.5-Omni (Qwen Team, 2026)	54 / 72
Qwen2.5-Omni (Xu et al., 2025a)	47 / 56
Qwen3-Omni (Xu et al., 2025b)	50 / 65

- **Integrated dashboard**: an interactive view that gathers per-modality results in one place, shows coverage gaps at a glance, and supports cross-model and cross-checkpoint comparison for model selection (Figure 2).
- **Built-in verifier**: a small model, light enough to run on CPU, that judges whether a prediction is semantically correct. Its normalized score varies far less across engines and prompts than rule-based scoring under configuration mismatch, reducing the need for costly API judges (Table 3, §4.2).

OmniEvaluator was used in the development of HyperCLOVA X 8B Omni (NAVER Cloud HyperCLOVA X Team, 2026) and is publicly available with a live demo, demo video, and evaluation dashboard.

2 Related Work

Evaluation tools have matured within each modality; what is missing is a way to evaluate one model across all of them under a single, consistent setup.

Text evaluation. HELM (Liang et al., 2023) popularized holistic evaluation; `lm-eval-harness` (Biderman et al., 2024) provides reusable task implementations and documents common pitfalls; BIG-bench (Srivastava

Table 3: **Native vs. verifier score across evaluation engines and prompt conditions.** Each cell: `lmms-eval`/`VLMEvalKit` on the same outputs, under benchmark-specific (*Sp.*) and uniform (*Un.*) prompts; $[0, 100]$ scale, Δ=max−min. *Sp.* uses each framework's own format-constraining instruction (e.g. "Answer the question using a single word or phrase."), while *Un.* replaces it with a single format-agnostic instruction for all benchmarks; full prompts in Table 9.

			lmms-eval		VLMEvalKit		
Bench.	**Model**	**Score**	**Sp.**	**Un.**	**Sp.**	**Un.**	**Δ**
GQA	Qwen2.5-Omni-3B	Native	61.6	28.1	70.4	29.3	42.3
		Verifier	64.8	66.4	64.7	66.6	1.9
	Qwen2.5-Omni-7B	Native	60.9	28.5	70.1	0.9	69.2
		Verifier	63.3	63.9	63.1	66.0	2.9
MMStar	Qwen2.5-Omni-3B	Native	57.1	51.0	54.2	53.2	6.1
		Verifier	55.2	53.7	54.4	53.5	1.7
	Qwen2.5-Omni-7B	Native	62.7	47.9	62.1	61.1	14.8
		Verifier	61.5	48.0	62.3	61.5	14.3
OCRBench	Qwen2.5-Omni-3B	Native	82.2	82.1	25.6	25.6	56.6
		Verifier	84.3	84.3	84.6	84.2	0.4
	Qwen2.5-Omni-7B	Native	80.5	83.2	25.2	25.2	58.0
		Verifier	86.0	84.9	85.1	84.8	1.2
POPE	Qwen2.5-Omni-3B	Native	88.8	59.2	87.6	87.9	29.6
		Verifier	88.8	88.8	91.3	91.1	2.5
	Qwen2.5-Omni-7B	Native	88.4	0.0	87.6	86.4	88.4
		Verifier	88.4	87.1	91.5	87.8	4.4
RealWorldQA	Qwen2.5-Omni-3B	Native	64.6	64.4	62.6	63.8	2.0
		Verifier	64.7	64.6	62.6	63.8	2.1
	Qwen2.5-Omni-7B	Native	69.2	69.9	69.7	69.5	0.7
		Verifier	69.3	70.1	69.7	69.5	0.8

et al., 2023) and the Open LLM Leaderboard (Fourrier et al., 2024) broadened coverage and showed centralized evaluation at scale. Evalverse (Kim et al., 2024) is closest to our design, unifying several text evaluation frameworks behind one interface, but at the time of writing it does not extend beyond text.

Vision evaluation. VLMEvalKit (Duan et al., 2024) unifies evaluation across a wide range of vision–language models, and VHELM (Lee et al., 2024) extends holistic evaluation to the vision–language setting. Benchmarks such as MMMU (Yue et al., 2024), MMBench (Liu et al., 2025b), and Video-MME (Fu et al., 2025a) standardize *what* to measure, but *how* they are run (prompts, answer parsing, scoring) still differs from framework to framework.

Multi-modal and audio evaluation. LMMs-Eval (Zhang et al., 2025a) extends coverage beyond image–text. For audio, dedicated toolkits such as AudioBench (Wang et al., 2025) and UltraEval-Audio (Shi et al., 2026) sit alongside benchmarks such as LibriSpeech (Panayotov et al., 2015), CoVoST2 (Wang et al., 2021), and VoiceBench (Chen et al., 2026). Each of these tools covers part of the modality space; evaluating an omni-modal model still means combining several of them.

Answer verification vs. preference judging. *LLM-as-a-judge* (Li et al., 2024) asks a model to rate the quality of a response, often without a reference answer. A *verifier* solves a narrower problem: given the question, the reference answer, and the prediction, decide whether the prediction is correct. Existing verifiers were built to check the answers of reasoning models on text benchmarks, or to provide verifiable rewards for reinforcement learning (Chen et al., 2025; Liu et al., 2025a; Zhang et al., 2025b). OmniEvaluator instead uses a verifier as part of its evaluation infrastructure, applying one lightweight, text-based model to benchmarks from all four modalities (§4.2).

Gap addressed by this work. Many studies show how fragile evaluation scores can be: prompt changes (Mizrahi et al., 2024; Sclar et al., 2024), the choice of evaluation examples (Maia Polo et al., 2024), answer-choice ordering (Pezeshkpour and Hruschka, 2024), and cross-framework differences (Zhang et al., 2025a; Alzahrani et al., 2024) can each shift results substantially. OmniEvaluator responds at the system level: a shared intermediate schema makes heterogeneous evaluators interoperable, every run yields a reproducible artifact, the verifier score holds steady when engine or prompt configurations differ, and a federated pipeline shares GPU servers across evaluations. To our knowledge, no existing framework offers this combination.

3 OmniEvaluator

OmniEvaluator is a unified evaluation framework that centers all factors influencing evaluation outcomes—prompt template, generation configuration, model revision, benchmark version, and metric parameters—within a single, inspectable configuration. The framework is guided by two design goals. *Multi-modal generality*: support benchmarks across text, image, video, audio, and cross-modality settings in a single pipeline, primarily by integrating entire evaluation frameworks—rather than individual benchmarks—to leverage existing implementations and keep pace with upstream releases; benchmarks that external frameworks do

not cover, such as audio, omni-modal, and tool-calling tasks, are supported through a built-in engine. *Visualization and interpretability*: enable consistent cross-modal model comparison despite heterogeneous metric systems, with an integrated dashboard that visualizes capability profiles and trade-offs across modalities.

3.1 Composable Architecture through Intermediate Schema

OmniEvaluator decomposes evaluation into four independent stages—data iteration, inference, post-processing, and metric computation—and mediates all inter-stage exchange through a unified intermediate schema (Figure 1). Each record in this schema is a structured object containing the benchmark sample, the raw model prediction, the post-processed output, and the computed metric score. Below is one such record, for a video benchmark.

```
{
  "benchmark": "mvbench_test_64frames",
  "messages": [{
    "role": "user",
    "content": [
      { "type": "video",
        "value": ".../action_sequence__0.mp4",
        "num_frames": 64,
        "sampling_strategy": "uniform" },
      { "type": "text",
        "value": "What happened after ...?" }
    ]
  }],
  "label": ["A"],
  "output": {
    "text": {
      "prediction": ["A. Ate the medicine."],
      "prediction_postprocessed": ["A"]
    },
    "reasoning_content": null
  },
  "metrics": { "exact_match": 1.0 }
}
```

The record shape does not change with modality: a text, image, video, or audio sample differs only in the `content` entries it carries and the modality-specific fields those entries add—`num_frames` and `sampling_strategy` for video, a sampling rate for audio—which the inference engine reads and the evaluation framework never has to know about. Postprocessing writes to its own slot beside the raw prediction, so a metric can read either one without knowing which transformation produced it; the schema is described further in Appendix A. This common representation lets any inference engine feed into any postprocessor and metric module: the same multiple-choice extraction or ASR normalization logic is reused across all benchmarks of that task type, and shared metrics such as exact match are computed by a single implementation regardless of which framework defined the benchmark.

Each inference engine and evaluation framework listed in Table 1 is connected to this schema through a thin adapter that translates between its native data format and the common representation, reducing integration cost from $O(N \times M)$ to $O(N+M)$: a single adapter makes any newly released upstream framework composable with all existing engines and modules. Because postprocessing and metric modules are shared across frameworks, scoring discrepancies that would otherwise arise from differing *metric implementations*—such as two frameworks reporting "ANLS" with different edit-distance thresholds—are structurally eliminated. Full per-modality evaluation results obtained by combining these components are provided in Table 10 (Appendix).

3.2 Evaluation as a Reproducible Specification

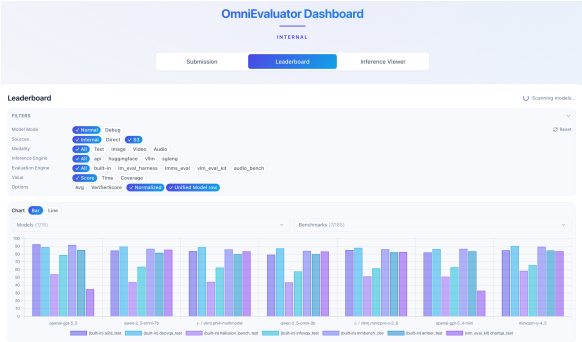
Every evaluation run emits a self-contained artifact that bundles: (i) the evaluation configuration (prompt template, generation parameters, model revision, benchmark version, and metric settings), (ii) all intermediate outputs (raw predictions and postprocessed results), and (iii) the final scores. This artifact is the unit of reproducibility: given the same artifact, any user can re-run the identical evaluation or inspect every decision that influenced a score; in practice, teams use artifacts as baseline anchors across training stages, checkpoint references for regression testing, and verifiable records for external releases. The dashboard (§4.1) automatically ingests these artifacts, enabling cross-experiment comparison without manual data wrangling.

3.3 Quick Start

OmniEvaluator supports local CLI evaluation for fine-grained control and a remote mode in which users submit requests to a persistent server. Both modes produce the same provenance-rich artifacts (§3.2) and stream results to the dashboard on completion; installation and server-launch details are given in Appendix B.

Local evaluation. Users run evaluation directly via the CLI; an existing artifact can also be given as input to reproduce its results exactly:

```
python run.py \
  --inference_engine=vllm \
  --url={vllm_url} \
  --exp_name=hyperclovax-seed-4b \
  --evaluation_engine=builtin \
  --benchmarks=haerae_vision_test
```



#	Model	(built-in) ai2d_test	(built-in) docvqa_test	(built-in) hallusion_bench_test	(built-in) infovqa_test	(built-in) mmbench_dev
Metric		exact_ma	anls	judge_binary	anls	exact_match
1	anthropic-claude-haiku-4-5	62.2	78.2	54.9	35.5	82.3
2	anthropic-claude-opus-4-8	89.8	45.6	63.2	17.5	90.4
3	anthropic-claude-sonnet-4-6	67.7	61.4	60.1	23.1	84.6
4	google-gemini-3_1-flash-lite	90.7	88.2	57.0	69.8	90.3
5	google-gemini-3_1-pro-pr...	15.2	36.0	28.4	35.7	13.3
6	hyperclovax-omni-8b	80.2	88.3	42.3	62.0	85.3
7	hyperclovax-vision-v2-see...	83.3	88.6	42.0	63.1	86.6
8	minicpm-o-2_6	85.2	88.0	51.3	61.7	86.2
9	minicpm-o-4_5	84.8	90.3	58.5	65.8	89.4
10	openai-gpt-5_4-mini	82.0	86.4	50.9	63.1	86.7
11	openai-gpt-5_5	92.5	88.9	54.0	78.8	91.8
12	phi4-multimodal	83.7	88.9	44.3	62.6	85.9
13	qwen-2_5-omni-3b	79.1	87.3	43.7	57.7	84.1
14	qwen-2_5-omni-7b	84.5	89.5	43.9	63.7	86.7

Figure 2: **OmniEvaluator dashboard (leaderboard view).** The dashboard synchronizes with evaluation artifacts produced by OmniEvaluator, providing an integrated view of benchmark results across experiments. *Left*: composable filters (modality, inference engine, evaluation engine) and cross-model benchmark comparison charts. *Right*: the consolidated leaderboard; missing entries explicitly surface modality coverage gaps (https://github.com/naver-ai/omni-evaluator).

Remote evaluation. Users submit an evaluation request to a persistent server via a single REST call, with no local installation required—useful when a model undergoes multi-stage training and evaluation environments change frequently:

```
curl -X POST http://{host}:{port}/add_job \
 -H "Content-Type: application/json" \
 -d '{"arguments": {
  "inference_engine": "huggingface",
  "model_name_or_path": "Qwen/Qwen2.5-Omni-3B",
  "exp_name": "qwen2.5-omni-3b",
  "evaluation_engine": "builtin",
  "benchmarks": "librispeech_test_clean"
 }}'
```

The server records each request with its evaluation specification (§3.2) for server-side reproducibility; results appear on the dashboard upon completion.

Table 4: **Verification accuracy on our human-verified held-out test split** (n=1,566)**.** All models use an identical text-only configuration. Per-group best in **bold**, second underlined.

Type	Verifier	acc
Proprietary (API)	GPT-5.5	90.2
	GPT-5.4-mini	82.8
	Claude-Opus-4.8	**90.4**
	Claude-Haiku-4.5	84.3
	Gemini-3.1-Pro	89.8
	Gemini-3.1-Flash-Lite	83.5
Open-source	OmniEval Verifier	**85.0**
	Qwen3.5-9B	79.9
	Qwen3.5-4B	79.8
	Qwen3.5-0.8B	64.2
	Qwen3-4B-Instruct-2507	76.2
	Qwen3-0.6B	56.1

4 Supported Features

4.1 Integrated Dashboard

The dashboard provides an integrated view of benchmark results across experiments and modalities (Figure 2); evaluation artifacts are automatically synchronized via a remote storage backend (§3.2), so results become browsable as soon as a run completes. Rather than crowning a single best model, the dashboard is designed to visualize cross-modal capability profiles and trade-offs, and it explicitly surfaces *modality coverage gaps*: benchmarks a model has not been evaluated on appear as missing entries, making it immediately apparent which modalities remain untested.

The dashboard also offers interactive visualizations—including radar charts that overlay multiple models on user-selected benchmark axes—and side-by-side inspection of inference samples for contrasting predictions across training stages or against baselines to identify modality-specific error patterns (demonstrated in the live demo). To support model selection, it further reports the mean of the **verifier score**—a single normalized value that our built-in verifier assigns to every benchmark on the same $[0, 100]$ scale (§4.2)—across benchmarks: because it varies far less across evaluation engines and prompt conditions, its average is comparable where raw native metrics are not (measured on disparate, unbounded scales), and visualizing this single signal across models and checkpoints turns per-benchmark results into an actionable decision aid.

Table 5: **API-equivalent cost of one full evaluation pass.** Avg. In/Out Tok. are the mean input/output tokens per sample. Cost is in USD under public pricing.

Modality	# Bench.	# Samples	Avg. In Tok.	Avg. Out Tok.	GPT-5.4-mini	GPT-5.5	Claude-Haiku-4.5	Claude-Opus-4.8	Gemini-3.1-Flash-Lite	Gemini-3.1-Pro
Image	8	21,433	44.3	29.2	3.53	23.52	4.08	20.39	7.06	16.94
Audio	8	33,293	23.1	11.1	2.24	14.93	2.62	13.08	4.48	10.75
Video	8	31,260	69.1	15.2	3.76	25.06	4.54	22.68	7.52	18.04
Text	8	43,640	133.8	177.0	39.14	260.92	44.46	222.30	78.28	187.86
Total	**32**	**129,626**	**75.0**	**70.9**	**48.67**	**324.43**	**55.69**	**278.60**	**97.34**	**233.59**

4.2 Verifier

Rule-based metrics such as exact match, WER (Park et al., 2024), and BLEU (Papineni et al., 2002) compare strings, and two situations complicate this comparison. First, the score depends on how the evaluation is run: the same model outputs, scored under different prompt and metric configurations, swing by up to 88 points on a $[0, 100]$ scale (Table 3). Each framework's benchmark-specific prompt constrains the output shape—"a single word or phrase", "the option's letter"—and its metric implementation is written for that shape (Table 9), so the prompt is part of the harness rather than a neutral wrapper: when a model does not follow it closely, the metric no longer measures what it was written to measure. How much this matters depends on the parser, which also differs across engines: on OCRBench the two frameworks disagree by more than 50 points under the same prompt, while on RealWorldQA both parsers accept the answer in either form and the prompt barely matters. Second, even when a model follows the prompt exactly, string comparison misses answers that are semantically correct but phrased differently from the reference or buried in reasoning traces, and a corpus-level average can be dominated by a few such cases. A common remedy is to have an external LLM API judge the responses, but this exposes internal data to a third party, drifts as the judge version changes, and accumulates cost over repeated evaluations (Table 5).

Our remedy is to build the semantic check into the system itself. We train and release **OmniEval Verifier**, a compact model that reads a `(question, reference, prediction)` text triple and returns a rationale and a binary verdict on whether the prediction answers the question correctly. Training and data details are in Appendix C. Built on `Qwen3-0.6B` and distributed as an 8-bit (Q8) GGUF model, it runs on CPU-only machines via `llama.cpp`[2], so OmniEvaluator ships it as a default scorer alongside native metrics, adding no external API calls and near-zero marginal cost. Per-sample verdicts aggregate into the **verifier score**, a value on a $[0, 100]$ scale computed the same way for every benchmark; on the outputs of Table 3, this score moves far less than the native metric—on GQA and POPE, by a few points where the native metric moves by 40 to 88. Under a benchmark-specific prompt the two largely agree (the *Sp.* columns); the verifier's value is therefore not that it outperforms a well-configured native metric, but that it holds steady when the configuration does not match (the *Un.* columns).

On our human-verified held-out test split, OmniEval Verifier reaches 85.0 accuracy, matching or exceeding cost-efficient proprietary judges (GPT-5.4-mini 82.8, Claude-Haiku-4.5 84.3) and outperforming the open-source models we evaluated under the same text-only configuration (Table 4). Against native metrics, the verifier score closely tracks exact-match-style metrics, and for unbounded metrics such as WER and BLEU it adds a per-sample correct/incorrect view that corpus-level averages do not; where the two disagree, the gap is informative rather than contradictory, since a corpus-level average and a per-utterance verdict measure different things (Table 6). We position the verifier score as a *complement* to native metrics rather than a replacement: native metrics remain the reference under their intended setup, while the verifier adds a signal that survives cross-engine and cross-prompt variation. During model development, its mean proved comparable across benchmarks and training stages and aided model-selection decisions.

4.3 Federated Evaluation

The modular architecture of §3.1 separates inference from evaluation: models are hosted by serving engines such as vLLM, and evaluation clients talk to them over HTTP. **Federated evaluation** builds on this split: inference servers and evaluation clients form a many-to-many pool, decou-

[2] https://github.com/ggml-org/llama.cpp

Table 6: **Comparison of native benchmark metrics and Verifier Score.** Native metrics are EM (exact match) for MMLU-Pro and GSM8K-CoT, PASS@1 for MBPP, WER (%, ↓) for LibriSpeech, BLEU-4 (↑) for CoVoST2 (en→zh), and EM for VocalSound; the Verifier Score is uniformly scaled to $[0, 100]$ (↑). Across these heterogeneous native scales, the Verifier Score provides a single comparable signal.

Modality	Benchmark	Model	Native Metric	Native	Verifier Score
Text	MMLU-Pro	Qwen2.5-Omni-3B	EM (↑)	42.0	41.9
		Qwen2.5-Omni-7B		51.1	51.0
		HyperCLOVAX-SEED-4B		56.8	57.1
	MBPP	Qwen2.5-Omni-3B	PASS@1 (↑)	53.4	83.6
		Qwen2.5-Omni-7B		50.4	86.5
		HyperCLOVAX-SEED-4B		54.2	76.2
	GSM8K-CoT	Qwen2.5-Omni-3B	EM (↑)	32.4	81.3
		Qwen2.5-Omni-7B		70.2	84.8
		HyperCLOVAX-SEED-4B		48.3	51.2
Audio	LibriSpeech	Qwen2.5-Omni-3B	WER (↓)	2.6	71.2
		Qwen2.5-Omni-7B		2.3	63.5
		Phi-4-Multimodal		1.7	82.9
	CoVoST2 (en→zh)	Qwen2.5-Omni-3B	BLEU-4 (↑)	0.0	35.4
		Qwen2.5-Omni-7B		0.0	37.1
		Phi-4-Multimodal		0.0	36.1
	VocalSound	Qwen2.5-Omni-3B	EM (↑)	90.2	90.2
		Qwen2.5-Omni-7B		91.9	91.9
		Phi-4-Multimodal		32.2	36.0

Table 7: **Wall-time speedup of federated over conventional evaluation. Bold** / underline: best / second-best per modality.

Model	Text	Image	Video	Avg.
HyperCLOVAX-SEED-4B	1.41×	2.32×	**1.62**×	1.78×
HyperCLOVAX-SEED-Omni-8B	1.28×	2.29×	–	1.79×
Qwen2.5-Omni-3B	**1.76**×	2.70×	**1.62**×	**2.03**×
Qwen2.5-Omni-7B	1.60×	**2.80**×	1.54×	1.98×
Modality avg.	*1.51*×	*2.53*×	*1.59*×	

pling where benchmarks run from where GPUs are. This decoupling yields two complementary benefits. *First, it improves utilization of fragmented GPU resources.* A single client can spread benchmarks across multiple servers—a spare A100 on one machine, idle V100s on another—while data loading, metric computation, and verifier calls stay on CPU machines. GPUs that would otherwise be too scattered to serve a single job are thus aggregated into one logical evaluation pool. *Second, it minimizes GPU idle time and thereby improves throughput.* Many clients can drive one server, and their concurrent requests are merged by the engine's continuous (in-flight) batching, which keeps the accelerator saturated between requests rather than stalling on the request stream of any single evaluation process. Under the same GPU allocation, these two effects together yield a 1.3–2.8× wall-time speedup over conventional per-process evaluation, in which each framework runs inference from within its own process, with the largest gains on image benchmarks (Table 7).

5 Conclusion

OmniEvaluator turns fragmented omni-modal evaluation into a single workflow: existing engines and frameworks compose through one schema, every run yields a reproducible artifact, and results flow into one dashboard. Cross-framework comparisons (Table 3) show why this matters: identically named benchmarks diverge across engines and prompt configurations, while the verifier score on the same predictions moves far less. OmniEvaluator and the verifier are publicly released at `https://github.com/naver-ai/omni-evaluator` with a live demo and dashboard; a demo video is available at `https://www.youtube.com/watch?v=4Z5VZZWyXqY`.

Limitations

OmniEvaluator wraps upstream evaluators rather than reimplementing them, so a bug in an upstream framework can flow into the scores it reports. The design does not prevent this, but it makes such problems visible: every artifact pins exact framework versions, so re-running anchor models after an upgrade exposes score regressions (the daggered models in Table 10), and running the same benchmark under two engines surfaces disagreements that point to upstream issues (Table 3). The verifier judges only the textual triple and returns a binary verdict; these are the choices that keep it small enough for CPU and its score uniform across benchmarks, and its accuracy already matches or exceeds cost-efficient proprietary judges, though not the strongest ones (Table 4). Consequently, it cannot assess criteria that go beyond textual correctness—such as visual grounding, audio quality, or generation quality—which some omni-modal tasks require; for these, users should fall back on the corresponding native metrics. Finally, this paper covers omni-modal understanding, where predictions are text; extending evaluation and the verifier to multimodal outputs such as image and speech generation is a natural next step.

Acknowledgments

We thank the anonymous reviewers and the program committee for their constructive feedback. We are grateful to the Hyperscale AI team at NAVER Cloud AI for their work throughout the development of HyperCLOVA X 8B Omni and subsequent models, and for their support in building

the evaluation setup this system grew out of. We also thank Huiyeon Yang for help with the figures.

References

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, and 11 others. 2024. Lessons from the trenches on reproducible evaluation of language models. *Preprint*, arXiv:2405.14782.

Ding Chen, Qingchen Yu, Pengyuan Wang, Mengting Hu, Wentao Zhang, Zhengren Wang, Bo Tang, Feiyu Xiong, Xinchi Li, Chao Wang, Minchuan Yang, and Zhiyu Li. 2025. xVerify: Efficient answer verifier for reasoning model evaluations. *Preprint*, arXiv:2504.10481.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. 2026. VoiceBench: Benchmarking LLM-based voice assistants. *Transactions of the Association for Computational Linguistics*, 14:378–398.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, Jiancheng Gui, Luoyuan Zhang, Xian Sun, Fuwei Huang, Moye Chen, Zhuo Lin, Hanyu Liu, Qingxin Gui, Qingzhe Han, and 17 others. 2026. MiniCPM-o 4.5: Towards real-time full-duplex omni-modal interaction. *Preprint*, arXiv:2604.27393.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, Dahua Lin, and Kai Chen. 2024. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 11198–11201, New York, NY, USA. Association for Computing Machinery.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. 2024. Open LLM leaderboard v2. `https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard`.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025a. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24108–24118.

Chaoyou Fu, Haojia Lin, Xiong Wang, YiFan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, Long MA, Xiawu Zheng, Rongrong Ji, Xing Sun, Caifeng Shan, and Ran He. 2025b. VITA-1.5: Towards GPT-4o level real-time vision and speech interaction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2023. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Jihoo Kim, Wonho Song, Dahyun Kim, Yunsu Kim, Yungi Kim, and Chanjun Park. 2024. Evalverse: Unified and accessible library for large language model evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 25–33, Miami, Florida, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Tony Lee, Haoqin Tu, Chi Heem Wong, Wenhao Zheng, Yiyang Zhou, Yifan Mai, Josselin Somerville Roberts, Michihiro Yasunaga, Huaxiu Yao, Cihang Xie, and Percy Liang. 2024. VHELM: A holistic evaluation of vision language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 140632–140666. Curran Associates, Inc.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. LLMs-as-judges: A comprehensive survey on LLM-based evaluation methods. *Preprint*, arXiv:2412.05579.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan,

Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

Shudong Liu, Hongwei Liu, Junnan Liu, Linchen Xiao, Songyang Gao, Chengqi Lyu, Yuzhe Gu, Wenwei Zhang, Derek F. Wong, Songyang Zhang, and Kai Chen. 2025a. CompassVerifier: A unified and robust verifier for LLMs evaluation and outcome reward. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33466–33494, Suzhou, China. Association for Computational Linguistics.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2025b. MMBench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024*, pages 216–233, Cham. Springer Nature Switzerland.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinyBenchmarks: evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 34303–34326. PMLR.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

NAVER Cloud HyperCLOVA X Team. 2026. HyperCLOVA X 8B Omni. *Preprint*, arXiv:2601.01792.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. GPT-4o system card. *Preprint*, arXiv:2410.21276.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. LibriSpeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Chanho Park, Mingjie Chen, and Thomas Hain. 2024. Automatic speech recognition system-independent word error rate estimation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1979–1987, Torino, Italia. ELRA and ICCL.

David Peer, Philemon Schöpf, Volckmar Nebendahl, Alexander Rietzler, and Sebastian Stabinger. 2024. ANLS* – a universal document processing metric for generative large language models. *Preprint*, arXiv:2402.03848.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Qwen Team. 2026. Qwen3.5-Omni technical report. *Preprint*, arXiv:2604.15804.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Qundong Shi, Jie Zhou, Biyuan Lin, Junbo Cui, Guoyang Zeng, Yixuan Zhou, Ziyang Wang, Xin Liu, Zhen Luo, Yudong Wang, and Zhiyuan Liu. 2026. UltraEval-audio: A unified framework for comprehensive evaluation of audio foundation models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 566–577, San Diego, California, United States. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas, Thomas Wang, sdtblck, nopperl, gakada, researcher2, and 11 others. 2026. EleutherAI/lm-evaluation-harness: v0.4.12.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy F. Chen. 2025. AudioBench: A universal benchmark for audio large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4297–4316,

Albuquerque, New Mexico. Association for Computational Linguistics.

Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021. CoVoST 2 and Massively Multilingual Speech Translation. In *Interspeech 2021*, pages 2247–2251.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025a. Qwen2.5-Omni technical report. *Preprint*, arXiv:2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025b. Qwen3-Omni technical report. *Preprint*, arXiv:2509.17765.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2025a. LMMs-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 881–916, Albuquerque, New Mexico. Association for Computational Linguistics.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025b. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024. SGLang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems*, volume 37, pages 62557–62583. Curran Associates, Inc.

A Intermediate Schema

Every benchmark, regardless of modality, is represented by the same `Record` object (§3.1). Its `output` field holds a per-modality sub-output—`output.text` for text-producing benchmarks—carrying both the raw `prediction` and the `prediction_postprocessed` written by the postprocess step, alongside a top-level `reasoning_content` for models that emit a separate thinking trace. Which fields carry data varies by benchmark, but the shape does not, and every engine and framework adapter reads and writes through the same slots. Complete records for text, image, video, and audio benchmarks are available in the repository.[3]

B Installation and Server Launch

OmniEvaluator installs with a single command that clones the framework and its submodules; an administrator then starts the persistent evaluation server:

```
git clone --recursive {repo}
cd OmniEvaluator && pip install -e .

python launch_server.py \
  --port {port} \
  --log_dir="./logs"
```

C Verifier: Data and Training

Data. Training predictions are drawn from our own evaluation artifacts—the outputs of 16 evaluated models across four modalities and 155 benchmarks—to broaden the error distribution the verifier must score. Gold labels come from a multi-teacher API pipeline to reduce single-teacher bias, balanced to a 1:1 positive/negative ratio with limited rule-based augmentation. The held-out test split (n=1,566) is balanced over (modality, task) cells, prioritizes dispute samples, and is fully human-verified and disjoint from training.

Training. The verifier takes a `(question, reference, prediction)` triple and produces a rationale and a binary verdict (`Explanation: ... \nRating: 0|1`); the training mixture spans four modalities, from short exact-match responses to long reasoning traces. Full training hyperparameters are in Table 8.

[3] https://github.com/naver-ai/omni-evaluator/tree/main/demo

Table 8: OmniEval Verifier training hyperparameters.

Hyperparameter	Value
Base model	Qwen3-0.6B
LoRA r / α / dropout	8 / 16 / 0.05
LoRA targets	LM decoder q,k,v,o,gate,up,down
Trainable params	≈5.05M
Supervision	completion-only (target tokens)
Epochs	$\leq$3 (best on validation)
Learning rate	1e-4
Scheduler / warmup	cosine / 0.1
Effective batch	2×8 (accum) $\times$ 8 GPU $=$ 128
Gradient clip	1.0
Precision / quant.	fp32 / none
`max_seq_len`	4096
Hardware	8×V100 (32GB)
Distributed	DeepSpeed ZeRO-2
Seed	42

Failure modes of rule-based scoring. Corpus-level metrics can be dominated by a few pathological samples: on LibriSpeech, a refusal answer yields per-sample WER 266.6 and an off-transcript hallucination 175.0, distorting the aggregate, whereas the verifier simply marks such predictions incorrect.

Cost of API-based verification. Repeated over checkpoints and configurations, API-based verification compounds: at the per-pass cost in Table 5, 50×5 passes come to ≈\$12,000 for GPT-5.4-mini and ≈\$70,000 for Claude-Opus-4.8, whereas OmniEval Verifier runs at near-zero marginal cost.

Verification accuracy. On the human-verified test split, zero-shot inference with the base `Qwen3-0.6B` reaches only 56.1 accuracy; trained as OmniEval Verifier—small enough to run on CPU alone—it rises to 85.0, matching or exceeding cost-efficient proprietary judges (GPT-5.4-mini 82.8, Claude-Haiku-4.5 84.3) and every open-source model we evaluated (Table 4).

D Supported Benchmarks and Full Results

Table 10 reports per-benchmark scores across text, image, video, and audio; full results are continuously updated on the OmniEvaluator dashboard. Coverage spans general knowledge and instruction following, math, VQA and document understanding, video comprehension, speech recognition and translation, and sound and music understanding.

Table 9: **How the task prompt changes the answer format, and the answer format changes the score.** Per benchmark and engine: the instruction under benchmark-specific (*Sp.*) and uniform (*Un.*) prompts, one Qwen2.5-Omni-7B prediction, and its native/verifier score ($\{0, 1\}$). Table 3 reports all five benchmarks.

Bench.	Eval. Engine	Setting	Task prompt	Reference	Prediction	Native	Verifier
GQA	lmms-eval	Sp.	`Answer the question using a single word or phrase.`	`aluminum`	`aluminum`	1	1
		Un.	`Please Answer the question in an appropriate format.`		`The fence is made of aluminum.`	0	1
	VLMEvalKit	Sp.	`Answer the question using a single word or phrase.`	`brown`	`brown`	1	1
		Un.	`Please Answer the question in an appropriate format.`		`The ground in the picture is brown.`	0	1
POPE	lmms-eval	Sp.	`Answer the question using a single word or phrase.`	`yes`	`Yes`	1	1
		Un.	`Please Answer the question in an appropriate format.`		`Yes, there is a snowboard in the image. The person in the image is riding a snowboard down a snowy slope.`	0	1
	VLMEvalKit	Sp.	*(inline)* `Please answer yes or no.`	`Yes`	`yes`	1	1
		Un.	*(inline)* `Please Answer the question in an appropriate format.`		`<points x1="112" y1="112" alt="bottle">bottle</points>`	0	1

Table 10: **Per-modality benchmark results** (accuracy, %; OCRBench 0–1,000; ASR/AST in WER ↓ / BLEU ↑). Representative models shown; full results on the OmniEvaluator dashboard. †Anchor models, re-evaluated on every framework or config update to monitor regressions.

Mod.	Model	Benchmarks							
		MMLU-Redux	MMLU-Pro	IFEval	ARC-C	MATH	AIME'25	HLE	KoBALT
Text	GPT-5.5	96.4	88.2	94.1	96.2	99.1	100.0	41.6	87.4
	GPT-5.4-mini	84.2	77.4	85.8	94.3	89.1	36.7	5.0	39.4
	Claude-Opus-4.8	93.3	88.9	85.0	96.2	98.3	90.0	-	-
	Gemini-3.1-Pro	96.8	74.4	50.1	96.5	98.8	40.0	20.3	87.1
	HyperCLOVAX-SEED-Omni-8B	-	52.3	69.1	-	71.2	-	-	-
	Qwen3-Omni-30B-Instruct†	88.2	-	-	94.7	96.9	56.7	5.4	36.1
	Qwen2.5-Omni-7B†	74.1	51.1	51.9	86.4	71.1	6.7	5.1	20.9
	Phi-4-Multimodal†	69.3	50.4	70.6	82.3	60.8	3.3	5.8	11.3
	MiniCPM-o-4.5	76.0	60.4	83.0	90.0	82.4	16.7	4.3	-
		MMBench	MMMU	MathVista	AI2D	DocVQA	OCRBench	V-MME	MVBench
Image/Video	GPT-5.5	91.8	24.9	52.4	92.5	88.9	808	56.7	36.1
	GPT-5.4-mini	86.7	23.8	44.8	82.0	86.4	793	42.7	28.9
	Claude-Opus-4.8	90.4	55.9	50.8	89.8	-	840	52.1	18.1
	Claude-Sonnet-4.6	84.6	-	48.1	67.7	61.4	839	51.0	28.8
	HyperCLOVAX-SEED-Omni-8B	85.3	38.8	-	80.2	88.3	769	-	-
	Qwen3-Omni-30B-Instruct†	89.9	42.2	51.8	87.6	91.6	774	69.2	36.1
	Qwen2.5-Omni-7B†	86.7	47.2	59.8	84.5	89.5	847	61.1	68.2
	Phi-4-Multimodal†	85.9	47.7	61.2	83.7	88.9	821	33.6	35.5
	MiniCPM-o-4.5	89.4	48.2	51.8	84.8	90.3	832	71.3	63.3
		LibriSp. clean ↓	LibriSp. other ↓	Fleurs ↓	CV15 ↓	CoVoST2 en-zh ↑	VocalSound	ClothoAQA	MuchoMusic
Audio	Whisper-large-v3	3.96	2.01	4.03	24.35	-	-	-	-
	Qwen2-Audio-Instruct	38.2	38.2	52.8	29.4	0.1	78.6	62.4	18.6
	Voxtral-Small	32.7	35.2	35.9	22.0	5.1	45.7	53.1	43.4
	Voxtral-Mini	39.1	44.0	32.1	20.5	3.3	0.1	36.8	50.8
	HyperCLOVAX-SEED-Omni-8B	2.6	4.5	7.6	-	-	-	-	-
	Qwen3-Omni-30B-Instruct†	2.8	1.7	3.4	11.7	2.8	90.8	75.0	78.2
	Qwen2.5-Omni-7B†	4.3	4.3	5.0	15.4	2.5	85.4	73.6	75.5
	Phi-4-Multimodal†	28.5	30.7	29.3	13.1	7.2	27.8	48.9	48.1
	MiniCPM-o-4.5	3.9	1.7	3.9	13.3	0.7	74.8	54.1	73.4