Feature	Lighteval	HELM	OpenAI Evals	LM Evaluation Harness
Developer	Hugging Face	Stanford CRFM	OpenAI	EleutherAI
Primary Goal	Flexible, all-in-one LLM evaluation across multiple backends	Holistic, transparent evaluation with broad coverage and multi-metric analysis	Framework for building and running custom evals for OpenAI models	Few-shot evaluation with standardized, reproducible benchmarks
Ease of Use	High: Python API, CLI, and integration with HF ecosystem	Moderate: CLI-driven, config files, less plug-and-play	Moderate: Requires coding custom evals, integrates with OpenAI API	High: CLI-based, simple setup, but less GUI-friendly
Supported Models	HF Transformers, vLLM, TGI, Nanotron, OpenAI API, local models	Broad range: Open, closed, and limited-access models (e.g., GPT-4, LLaMA, Claude)	Primarily OpenAI models, extensible to others with custom setup	HF models, OpenAI API, vLLM, local models, adapters (e.g., LoRA)
Benchmarks/Tasks	Extensive: 100+ tasks (e.g., MMLU, GSM8K), custom tasks supported	42 scenarios (16 core + 26 targeted, e.g., MedQA, LegalBench), expandable	Custom tasks defined by user (e.g., SQL generation, Q&A)	100+ tasks (e.g., HellaSwag, ARC, GSM8K), supports custom tasks
Metrics	Accuracy, exact match, custom metrics, extensible	Multi-metric: accuracy, robustness, fairness, bias, toxicity, efficiency, calibration	Custom metrics (e.g., string match, model-graded), user-defined	Accuracy, log-likelihood, custom metrics, extensible

Customization	High: Custom tasks, metrics, and prompt templates	Moderate: Modular toolkit for new scenarios/metrics, less focus on custom prompts	High: Fully custom evals, but requires coding	High: Custom prompts, metrics, and tasks via YAML or Python
Backend Support	Multi-backend: HF Accelerate, vLLM, TGI, OpenAI API, Nanotron	Primarily local or API-based, less focus on inference optimization	OpenAI API, extensible to local models with effort	HF Transformers, vLLM, OpenAI API, local inference
Evaluation Style	Few-shot, zero-shot, CoT, detailed sample-by-sample results	Zero-shot, few-shot, multi-metric across scenarios	Custom (zero-shot, few-shot, etc.), depends on user implementation	Few-shot, zero-shot, CoT, reproducible with public prompts
Output Storage	HF Hub, S3, local storage, detailed logs	Publicly browsable results, raw prompts/completions on website	Local JSON logs, no built-in cloud storage	Local files, optional logging of samples
Speed/Optimization	High: vLLM for fast inference, multi-GPU support via Accelerate	Moderate: Focus on thoroughness over speed, no specific optimization	Moderate: Depends on OpenAI API or local setup	High: vLLM support, multi-GPU via Accelerate, batch size optimization
Transparency	High: Open-source, detailed results, active development	Very High: All prompts/completions public, living benchmark	Moderate: Open-source, but results depend on user's setup	High: Public prompts, reproducible, used in Open LLM Leaderboard
Use Case	Researchers/developers needing	Comprehensive analysis of LLM capabilities,	Developers building custom	Researchers benchmarking LLMs

	flexible, fast eval with HF integration	limitations, and trade-offs	evals for OpenAI-based apps	on standard tasks with reproducibility
Strengths	Multi-backend, speed, HF ecosystem, customizability	Broad coverage, multi-metric, transparency, standardized scenarios	Flexible custom evals, OpenAI integration	Simplicity, reproducibility, wide task support, few-shot focus
Weaknesses	Evolving API, some prompt inconsistencies	Heavyweight, less focus on speed or ease of use	Requires coding, limited built-in tasks	CLI-heavy, less holistic than HELM, shadow APIs
Community Adoption	Growing, tied to HF ecosystem	Strong in academia, used for HELM Lite and multimodal extensions	Moderate, used by OpenAI devs, less broad adoption	Very high: Backend for HF Open LLM Leaderboard, widely cited

Detailed Comparison

1. **Lighteval**
    a. **Overview**: An open-source toolkit from Hugging Face, evolved from LM Evaluation Harness, with inspiration from HELM. Focuses on flexibility and speed.
    b. **Best For**: Users in the Hugging Face ecosystem needing fast, customizable evaluations across diverse backends (e.g., vLLM for speed, Accelerate for scale).
    c. **Unique Feature**: Multi-backend support and seamless storage (HF Hub, S3).
2. **HELM (Holistic Evaluation of Language Models)**
    a. **Overview**: A Stanford-led framework emphasizing transparency and a multi-metric approach across diverse scenarios.
    b. **Best For**: Researchers seeking a comprehensive, standardized benchmark with deep insights into accuracy, fairness, robustness, etc.

   c. **Unique Feature**: Publicly browsable raw data and a modular toolkit for extending scenarios/metrics.

3. **OpenAI Evals**
   a. **Overview**: A framework for creating custom evaluations, primarily for OpenAI models, with a focus on iterative development.
   b. **Best For**: Developers building LLM-based applications with OpenAI APIs who need tailored evals (e.g., SQL correctness, JSON parsing).
   c. **Unique Feature**: Model-graded evals (using an LLM to judge outputs) and integration with OpenAI's ecosystem.
4. **LM Evaluation Harness**
   a. **Overview**: An EleutherAI project for few-shot evaluation, widely adopted for its simplicity and reproducibility.
   b. **Best For**: Researchers and practitioners benchmarking LLMs on standard tasks (e.g., MMLU, GSM8K) with minimal setup.
   c. **Unique Feature**: Powers the Hugging Face Open LLM Leaderboard, extensive task library, and vLLM support for speed.

Choosing the Right Tool

- **Lighteval**: Pick this if you want a modern, fast, and flexible solution with strong Hugging Face integration and multi-backend support.
- **HELM**: Ideal for a thorough, transparent evaluation across multiple dimensions (e.g., fairness, bias), especially in academic or auditing contexts.
- **OpenAI Evals**: Best for custom, application-specific evaluations, particularly if you're using OpenAI models and need iterative testing.
- **LM Evaluation Harness**: Go-to for quick, reproducible few-shot evaluations on standard benchmarks, widely trusted in the research community.