Evaluation Framework Comparison - Phase I

Objective

Large Language Models (LLMs) have demonstrated exceptional capabilities in language understanding and generation, fueling a wide range of Lyric applications from Global Search to ADP Assist. However, ensuring the reliability, transparency, and trustworthiness of LLM-based systems remains a critical challenge. Implementing a robust LLM evaluation framework is essential to addressing these challenges, showing how tracing and evaluation tools can support transparency, reliability, and continuous improvement in our applications.

Evaluation frameworks for LLM observability and tracing have rapidly evolved to meet the growing demand for transparency, reliability, and performance monitoring in generative AI systems. Tools like Arize Phoenix and Langfuse provide comprehensive observability, supporting both LLM and traditional ML workflows, with features such as token-level analysis, deep tracing, and prompt management. Confident AI / DeepEval focuses on test-based evaluation, enabling CI/CD-style checks to ensure model quality and consistency. Platforms like LangSmith and Traceloop offer tight integration with popular frameworks and structured logging for robust debugging. Lightweight tools such as PromptLayer and Helicone specialize in OpenAI-specific logging, cost tracking, and prompt analytics.

This project shows the first phase of our evaluation initiative, aimed at comparing three leading LLM observability frameworks, Langfuse (13.9k GitHub stars), Arize Phoenix (6.4k GitHub stars), and Confident AI / DeepEval (9.3k GitHub stars), focusing on core capabilities including comprehensive tracing, prompt management, and dataset management, evaluation and visualization.

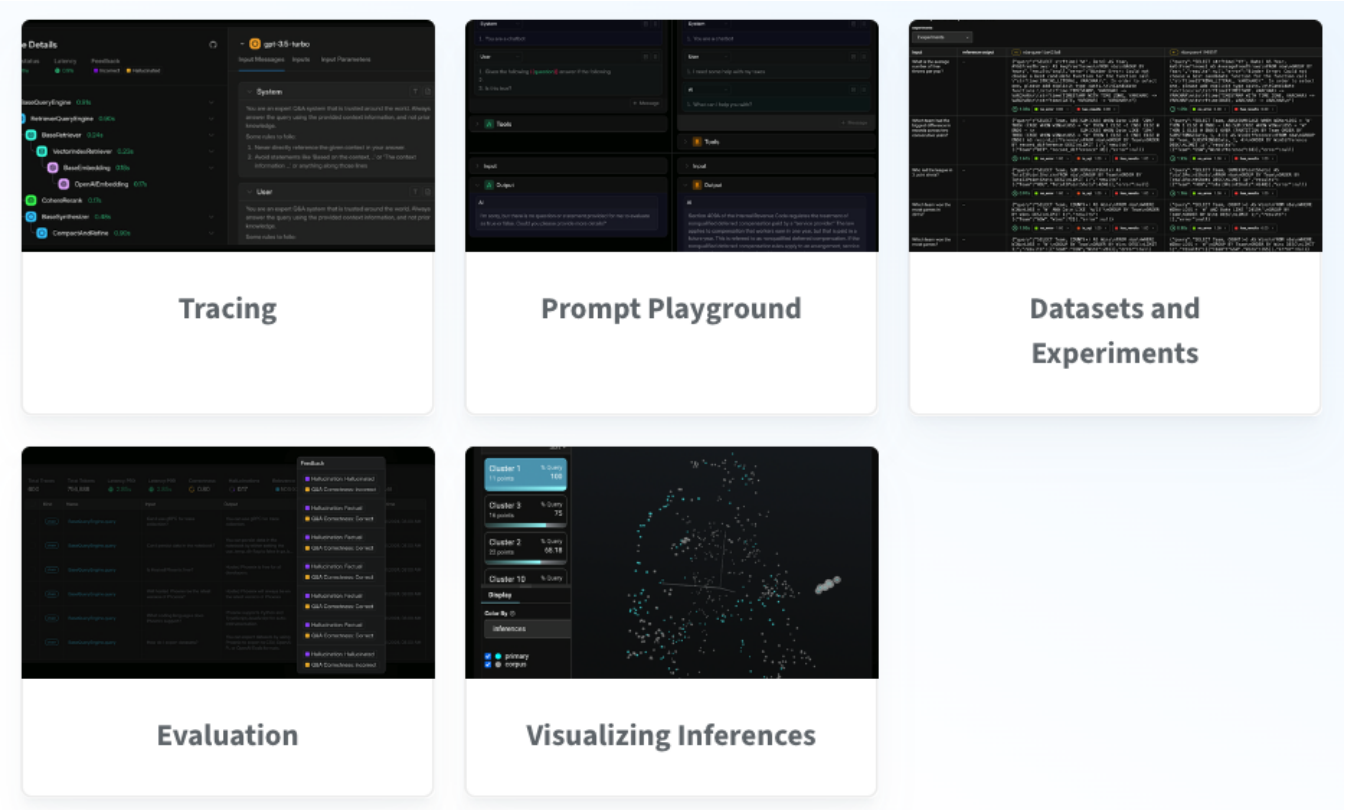

Figure 1 - Arize Phoenix provides tracing, prompt and data managements, evaluation and visualization. Both Langfuse, Confident AI / Deep Eval provide similar observability and evaluation functions.

Deployment Strategies

Both Arize Phoenix and Langfuse provide dual deployment paths, managed cloud for quick scalability and turnkey maintenance, or self-hosted installs for teams that need granular control over infrastructure and data governance. Confident AI / DeepEval, by contrast, centers on a SaaS offering (plus an open-source library) with no self-hosted alternative, leaving gaps in tracing, prompt management, and dash-boarding if you opt for library-only use.

Framework	Cloud (Managed)	Self-Hostable	Notes
Arize Phoenix	Hosted SaaS	Deployable via Docker, terminal, Colab, SageMaker, etc	Easy local start-up with flexible production deployment options.
Langfuse	Fully managed, scalable	Full control over data /security	Langfuse cloud is a fully managed service for ease of use and scalability, but self-hosted option for teams needs greater control over data, security and infrastructure.
Confident AI /DeepEval	Confident AI SaaS*	NOT self-hostable	SaaS meets most needs except for multimodal capabilities. DeepEval library enables offline eval, but lacks traceability, prompt/data management, and dashboard/reporting. ADP access to DeepEval documentation is currently blocked.

Table 1 – Evaluation Framework Deployment Strategies for Arize Phoenix, Langfuse, Confident AI / DeepEval

Comprehensive Tracing

Arize Phoenix

Phoenix automatically logs events, model invocations, prompt inputs/outputs, and metadata at each stage—creating detailed execution graphs. Its integrated tracing interface makes it easy to visualize the processing chain, diagnose issues, track latency, and maintain a complete audit trail of LLM application runs.

Pros

- Deep Visibility: Get granular, step-by-step visibility into how your AI agent works, what tools/functions are called, what the input/output is, and where things slow down or fail.

- Error Tracking and Reproducibility: Quickly see where and why errors happen, and reproduce the exact agent steps for debugging or improvement.

- Visual Dashboard: Powerful visual tools to explore, filter, and analyze traces and agent runs, making debugging much faster.

- Easy Integration: Simple to set up—can be integrated in a few lines, often without code change (if using popular AI agent libraries).

- OpenTelemetry Support: Works with industry-standard tracing systems, so DevOps/SRE teams can monitor agents with their existing tools.

- Works with Multiple Frameworks: Native support for popular frameworks (e.g., LangChain, LlamaIndex), as well as custom Python code.

Cons

- SLA Impact: Tracing adds small runtime overhead. It can matter for high-frequency, low-latency applications.

- Limited Offline Use: Full tracing experience is tied to the Phoenix dashboard, which is a live server app. Traces may not be as accessible offline.

- Possible Volume of Data: Tracing everything in large or long-running agents can generate a lot of data, which can be overwhelming to sift through without filters.

Langfuse

Langfuse provides comprehensive capabilities for tracing, debugging, and exploring traces. This functionality allows for a detailed understanding of the operations at each step, facilitating the visualization of each trace with clarity and precision.

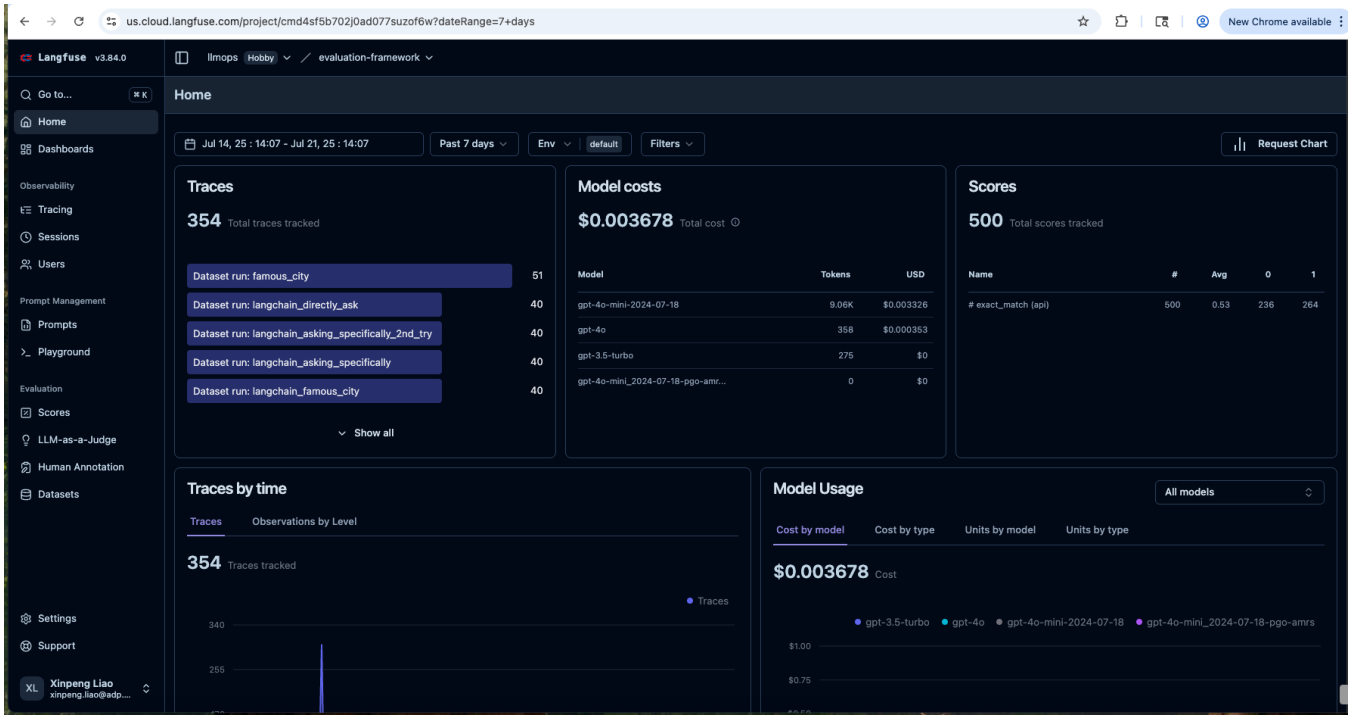

Figure 2 – Langfuse Sampled Tracing Dashboard

Pros

- Capture all the interactions with your LLM-application, provide advanced tracing, including users, sessions, tags, multi-modality, metadata, distributed tracing and agent graphs.

- Provides detailed span attributes and metadata for fine-grained debugging and root cause analysis.

- Supports multi-turn conversations/sessions, maintaining context over time, essential for understanding how lyrical responses evolve.

- Seamless integration with popular frameworks and SDKs (LangChain, LlamaIndex, OpenAI, AWS Bedrock, Mistral) in Python and JavaScript/TypeScript.

Cons

- May introduce complexity in configuring and maintaining trace integrations across diverse workflows.

- Detailed tracing can increase storage and processing overhead, requiring thoughtful management.

- Potential learning curve to distributed tracing and observability tools (e.g. ClickHouse, LangChain & LangGraph).

Confident AI / DeepEval

Pros

- LLM and non-LLM Call: Confident AI's SaaS platform (not just DeepEval OSS) supports end-to-end "inference pipelines" tracing. They explicitly state you can monitor LLM calls with inputs/outputs and attach metadata.

- Multi-turn Context Support: The platform offers session-level monitoring, enabling developers to trace conversations across multiple turns—useful for chat-based applications.

- Custom Metadata: Traces can include custom span-level attributes, giving teams flexibility to tag and organize trace data for deeper analysis.

- Latency and Cost Tracking: Built-in dashboards display API latency, throughput, and estimated cost metrics, helping teams monitor both performance and budget.

- SDK Integration: Offers Python SDK and DeepEval.ts for JavaScript/TypeScript, making it easy to integrate tracing in both backend and frontend applications.

Cons

- Limited Non-LLM Tracing: While LLM spans are supported, other components like database queries or vector store lookups aren't first-class citizens—unlike what Langfuse and Phoenix offer.

- No Multimodal Support: Currently, Confident AI does not provide tracing for image, audio, or other non-text modalities.

- SaaS-Only Model: Full tracing functionality requires using Confident AI's hosted SaaS platform. There is no self-hosted version available.

- Access Restrictions: In ADP environments like ADP, access to DeepEval documentation or services is blocked, limiting usability.

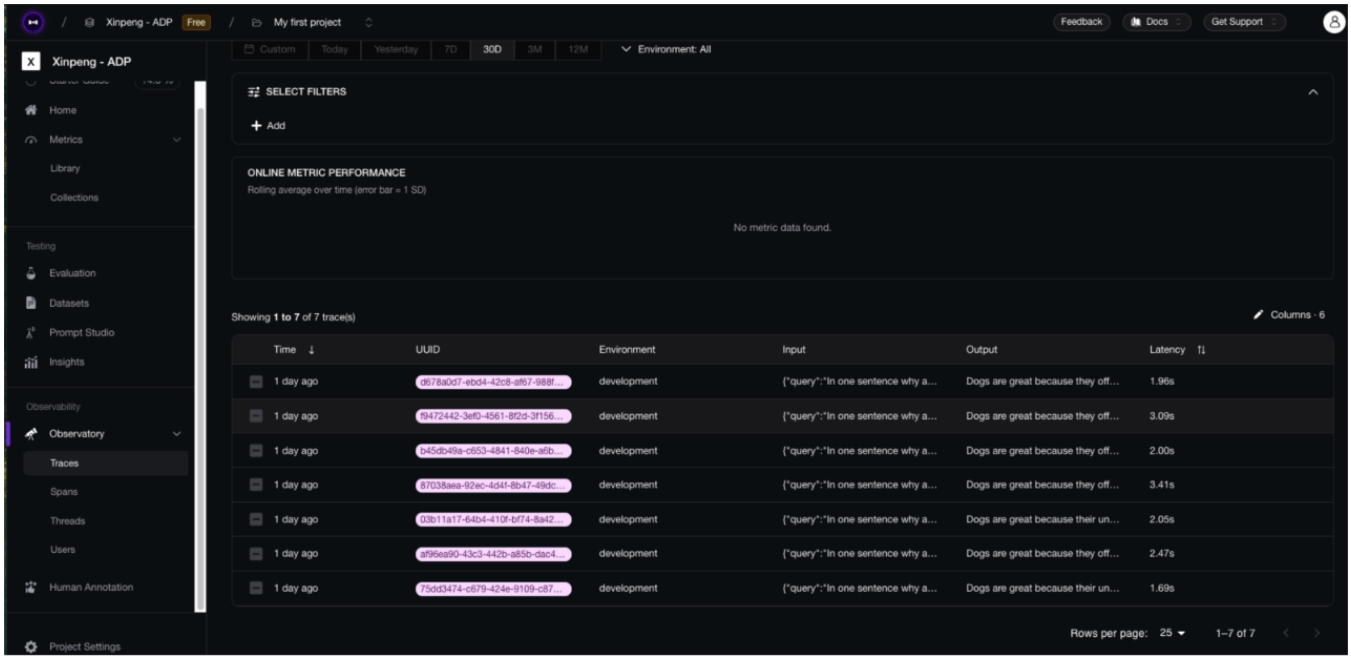

Figure 3 – Confident AI Sampled Tracing Dashboard

Robust Evaluation

Arize Phoenix

Phoenix integrates both automated and manual evaluation workflows. It supports built-in and custom metrics (e.g., BLEU, ROUGE, semantic similarity) as well as human-in-the-loop review. The framework links evaluation results directly to experiments and traces, helping users systematically assess model improvements and understand performance dynamics.

Pros

- Continuous Monitoring: Can be hooked into automated tests/CI for ongoing regression detection.

- Facilitates Error Analysis: Makes it fast to spot edge cases and systematically improve your app's weak spots.

- Integrated Dashboard: Unified place to compare, analyze, and drill down into evaluation results visually.

- Multi-metric Reporting: Evaluate on many axes (accuracy, relevance, harmlessness, etc.) simultaneously.

- Flexible & Rich Evaluations: Supports both automated metrics and human-in-the-loop feedback; extensible to custom use-cases.

- Plug-and-Play: Works with LLMs, chains, agents, and integrates well with many existing workflows.

Cons

- Data and Scaling: Large batch evaluations (especially with human or LLM-in-the-loop metrics) can be costly and slow, particularly with big datasets.

- Limited Out-of-the-Box Metrics: Some evaluation needs (domain-specific, specialized behaviors) require custom implementation.

Langfuse

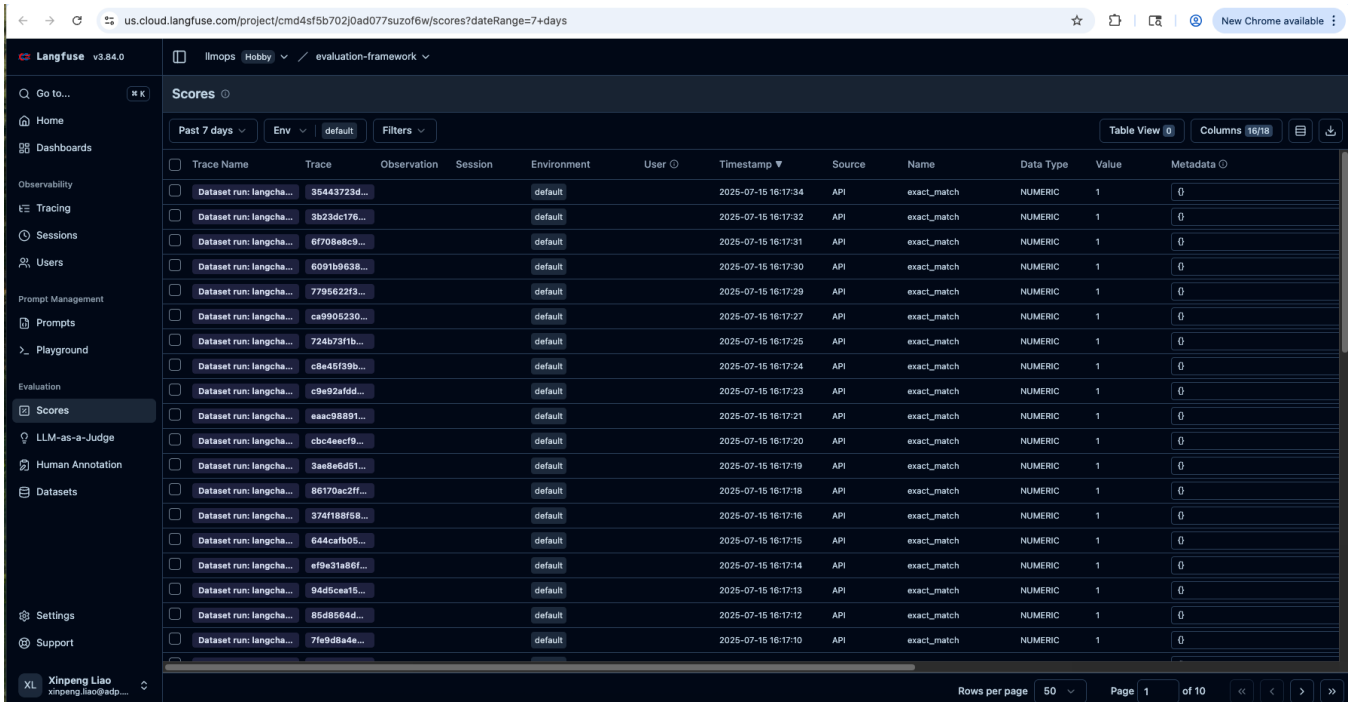

Figure 4 - Langfuse Sampled Evaluation Scores

Pros

- Model-Based Evaluation (LLM-as-a-Judge): Support automated evaluations using large language models to assess output quality, scoring factors such as accuracy, hallucination, and toxicity without human input.

- Custom Scoring Workflows: You can define and ingest your own evaluation metrics via Langfuse's SDK or API. This includes logic for structured format checks, domain-specific scoring, or external evaluation pipeline integration.

- Subjective Metric Support: Flexible metric definition allows you to evaluate subjective aspects (e.g., lyrical creativity or tone), tailoring evaluations to your specific application needs.

- Human Annotation Integration: Manual annotations can be added directly through the Langfuse UI, enabling team-based review processes. These annotations help benchmark automated evaluations and validate models.

- User Feedback Capture: Enables both explicit (e.g., thumbs up/down, 5-star ratings) and implicit (e.g., engagement time, click-throughs, acceptance /rejection) feedback to be tied to traces, providing a rich layer for behavioral evaluation.

- Development and Production Coverage: Evaluations can be run on both curated datasets during development and live application traces in production, enabling iterative, real-world performance tracking.

- Multimodal Readiness: Able to incorporate non-text data into evaluations, which is useful for applications like lyric generation that may involve audio, metadata, or visuals.

Cons

- LLM-as-a-Judge Requires Tuning: Automatic scoring by LLMs can be subjective or inconsistent unless carefully prompt-engineered and validated against human benchmarks.

- Annotation Workflow Can Be Manual: While Langfuse supports manual annotations, scaling this process across large datasets still requires time and team effort.

- Evaluation Requires Customized Setup: Custom evaluations, while powerful, require engineering effort to define metrics, structure feedback ingestion, and build evaluation pipelines.

- No Built-in Dataset Curation Tools: Langfuse assumes you bring your own datasets for evaluation. It doesn't provide tools for dataset creation, labeling, or version control out of the box.

Confident AI / DeepEval

Pros

- Rich Set of Prebuilt Evaluation Metrics: Confident AI provides a wide variety of ready-to-use metrics out of the box—covering summarization, hallucination, bias, toxicity, coherence, RAG performance, agentic tasks, and conversational quality. These are well-documented and can be executed using any LLM, traditional NLP methods, or statistical models.

- Flexible Evaluation Modes: Supports both end-to-end evaluation (e.g., final outputs) and component-level evaluation (e.g., intermediate steps like tool usage or context relevance), making it useful for complex pipelines like RAG or agent frameworks.

- LLM-as-a-Judge Support: Enables model-based scoring with predefined rubrics, allowing LLMs to assess outputs automatically with minimal setup.

- Custom Metric Support: Users can define their own prompt-based or Python-based evaluation logic, which integrates smoothly with the DeepEval ecosystem.

- Web-Based Human Annotation Tool: Confident AI SaaS includes a user-friendly web interface to collect human ratings and compare them with model-generated scores.

- Supports Batch and Real-Time Evaluation: Works well in both development (offline/batch mode) and production (live monitoring), helping teams close the loop on quality feedback.

Cons

- SaaS Dependency for Full Functionality: Many advanced features (like annotation tools and dashboards) are only available through the hosted Confident AI SaaS platform, which may not be suitable for teams requiring self-hosting.

- No Multimodal Evaluation Support: Current documentation makes no mention of support for image, video, or audio evaluation—limiting use in non-text-heavy applications.

- Potential Access Restrictions: In some enterprise settings (e.g., ADP), access to loading datasets from Confident AI SaaS or services may be blocked, limiting usability.

- Inflexibility for All Use Cases: While powerful, each use case entails a separate evaluation metrics or workflow fully customized by Confident AI leaving clients with less control if the number of use cases increases.

- DeepEval only for LLM Evaluation: Without the SaaS platform, we can curate the dataset via code, run off-line evaluations. The default telemetry should be opted out to avoid connecting to 3rd party vendors, e.g., PostHog. The default Judge LLM should be customized to use our LLM providers via AI-gateway – not working so far due to the built-in inflexibility of DeepEval.

Prompt Management

Arize Phoenix

Phoenix provides a centralized way to manage and version prompts. It allows for storing, editing, and experimenting with different prompt formats directly within the framework. You can compare prompt variants, track which prompts were used in each experiment, and efficiently test prompt engineering strategies for better results.

Pros

- Centralization: All your prompts are in one place, making them easy to manage.

- Integration: You can fetch and use prompts easily from your code without hardcoding, supporting agile development.

- Version Control: You can track changes, experiment safely, and avoid losing old working prompts.

- Experimentation Tools: Built-in support for A/B testing and analytics speeds up prompt optimization.

- Feedback Incorporation: Real-time improvement of prompts based on output and user feedback.

- Translate or adapt prompts to the format required by different LLM providers, by integrating LangChain prompt templating capabilities.

Cons

- SLA Impact: Fetching prompts from a central repository (especially remotely) may add latency.

- Dependency on Platform/Tool: Relying on Phoenix's prompt management adds dependency on the Phoenix service to run agent workflow.

- Overhead: Managing prompts in a separate system may seem like extra work.

- Limited Built-in Multi-modal Prompting: As of mid-2025, multi-modal (non-text) prompt management may be less mature than frameworks directly targeting images/audio (though this is rapidly changing).

Langfuse

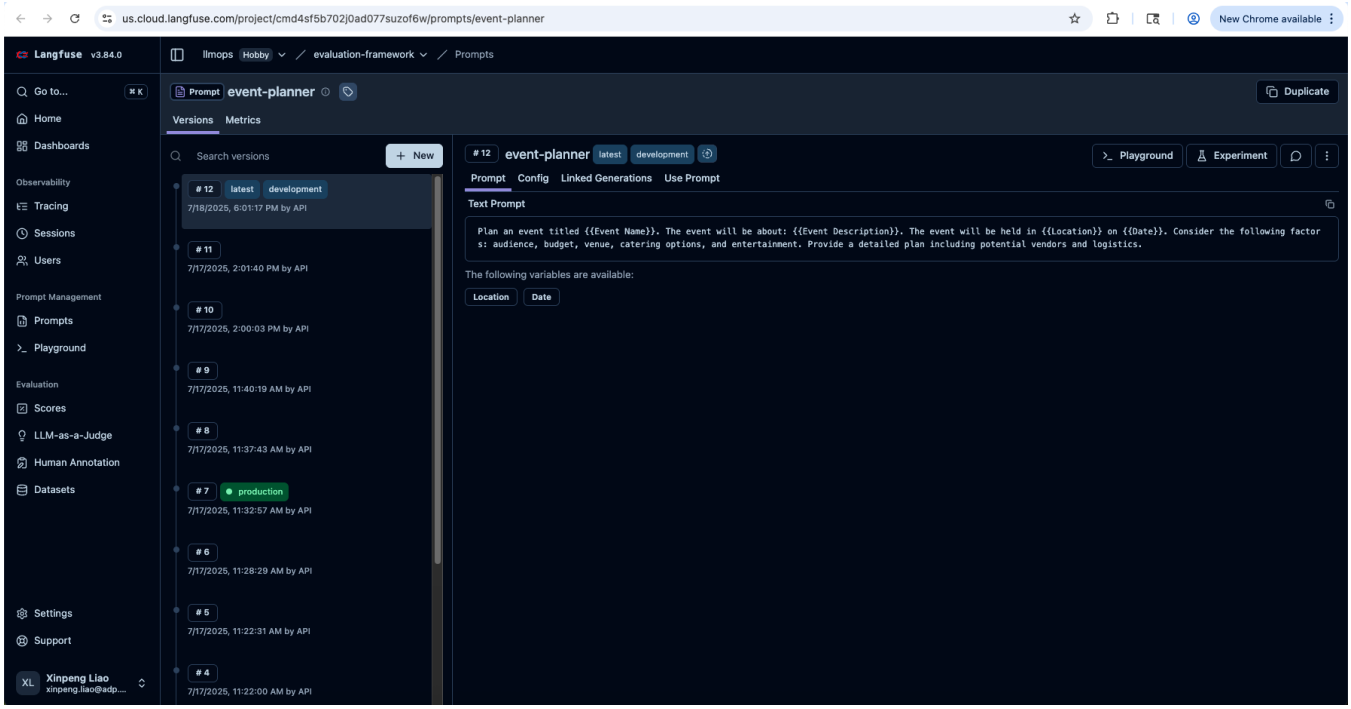

Figure 5 - Langfuse Sampled Prompts

Pros

- End-to-End Prompt Management: Langfuse offers a complete prompt CMS (Content Management System), supporting versioning, editing, labeling, and retrieval, all integrated into your LLM application's lifecycle.

- Tight Integration with Tracing: Every prompt call is logged and linked with the full LLM trace, making it easy to debug, monitor, and analyze prompt performance at a granular level.

- Version Control with Labels: Prompts can have multiple versions with labeled environments (e.g., "production", "staging"). Switching versions for production is as simple as updating a label, with no redeployment needed.

- Decoupled from Code: Prompts are centrally managed and can be updated without touching application code, enabling faster iteration and more flexible experimentation.

- Collaborative Editing UI: Langfuse supports in-platform prompt editing with full change history, allowing non-engineers (e.g., PMs, UX writers) to contribute and suggest improvements safely.

- A/B Testing and Evaluation: Prompts can be tested across datasets using Langfuse's evaluation features to ensure new versions outperform or maintain current quality. This reduces risk during prompt iteration.

- Prompt Analytics: Tracks per-prompt version metrics such as latency, cost, token usage, and even quality scores, enabling data-driven prompt optimization.

- LangChain Integration: Seamlessly works with LangChain and other frameworks; developers can pull latest prompt templates directly into their apps via SDKs.

- Chat Prompt & Local Caching Support: Useful for conversational agents; Langfuse supports dynamic and cached prompt templates to reduce latency and cost.

Cons

- Primarily Built Around Langfuse Ecosystem: While Langfuse integrates well with popular tools, its prompt management features are most powerful when you're also using Langfuse for tracing and evaluation. It's not a general-purpose prompt manager for arbitrary stacks.

- Requires Initial Setup and Workflow Design: Teams must invest time in defining how prompts are organized, versioned, labeled, and evaluated, especially if migrating from hardcoded prompts.

- Limited Support for Visual or Multimodal Prompt Templates: The system is designed for text-based prompt workflows. Managing visual or multimodal inputs may require additional tooling outside Langfuse.

- Still Emerging in Enterprise Settings: While the tool is feature-rich, it may not yet be widely adopted in regulated environments or legacy enterprise stacks that demand formal change controls or compliance workflows.

Confident AI / DeepEval

Pros

- Prompt Versioning Support: Confident AI's SaaS platform tracks prompt version history, allowing teams to review changes and revert when needed—ensuring stability across iterations.

- Interactive Prompt Playground: The Prompt Studio offers a user-friendly environment to test prompts and models interactively, which is helpful for rapid prototyping and experimentation.

- Experimentation Support: The platform allows controlled A/B or multi-variate experiments to compare prompt performance, helping teams choose the best prompt configuration based on results.

- UI-Driven Workflow: Non-technical users (e.g., PMs, designers, QA) can manage and iterate on prompts directly through the web UI, encouraging broader collaboration without engineering bottlenecks.

Cons

- Weak Code-to-Platform Sync: Confident AI lacks robust SDK-based prompt management—there's no clear way to sync prompts defined in code (e.g., LangChain apps) with the SaaS prompt store. This can lead to drift between code and platform versions.

- UI-Centric, Less Developer-Friendly: The prompt management workflow is focused on the web interface, which may slow down teams looking to automate prompt updates or manage them as part of CI/CD workflows.

- Limited Visibility into Prompt-Call Tracing: Compared to Langfuse, Confident AI provides less direct integration between prompt versions and LLM trace data, which could limit debugging and fine-grained analysis.

Dataset Management

Arize Phoenix

Phoenix allows users to import, create, and manage datasets for use in experiments. It orchestrates experiments where multiple prompts, models, or parameters are evaluated across dataset samples, recording input-output pairs. Experiments are tracked and versioned, enabling straightforward comparison of results and reproducibility.

Pros

- AI Evaluation Integration: Evaluate agents directly on managed datasets.

- Easy Ingestion: Multiple convenient ways to bring in data—UI upload, API, integrations, and production logging.

- Annotation & Labeling: Tools to manually or semi-automatically annotate or tag your data for training or evaluation.

- Versioning & History: Enables dataset iteration, comparison, and reproducibility.

- Collaboration: Cloud-based, with options for teams to collaborate on dataset curation and analysis.

Cons

- Limited Traditional Dataset Operations: Mature "ML data ops" features—like large-scale data diff, merge, merge conflicts, or fine-grained access control—may be limited compared to dedicated platforms.

- Mostly LLM/NLP-Focused: Dataset features are tuned for LLM/agent data (traces, prompts, responses) and may not support all data types or tabular use cases well.

- Scalability Limitations: For ultra-large datasets or high-throughput, you may hit performance constraints in the UI or API.

- Less Mature Enterprise Features: Features like granular audit trails, role-based access, or detailed regulatory compliance are still evolving.

- Dependence on Cloud/Hosted Service: Some features might depend on the hosted Arize/Phoenix service, which may be a limitation for data regulation.

Langfuse

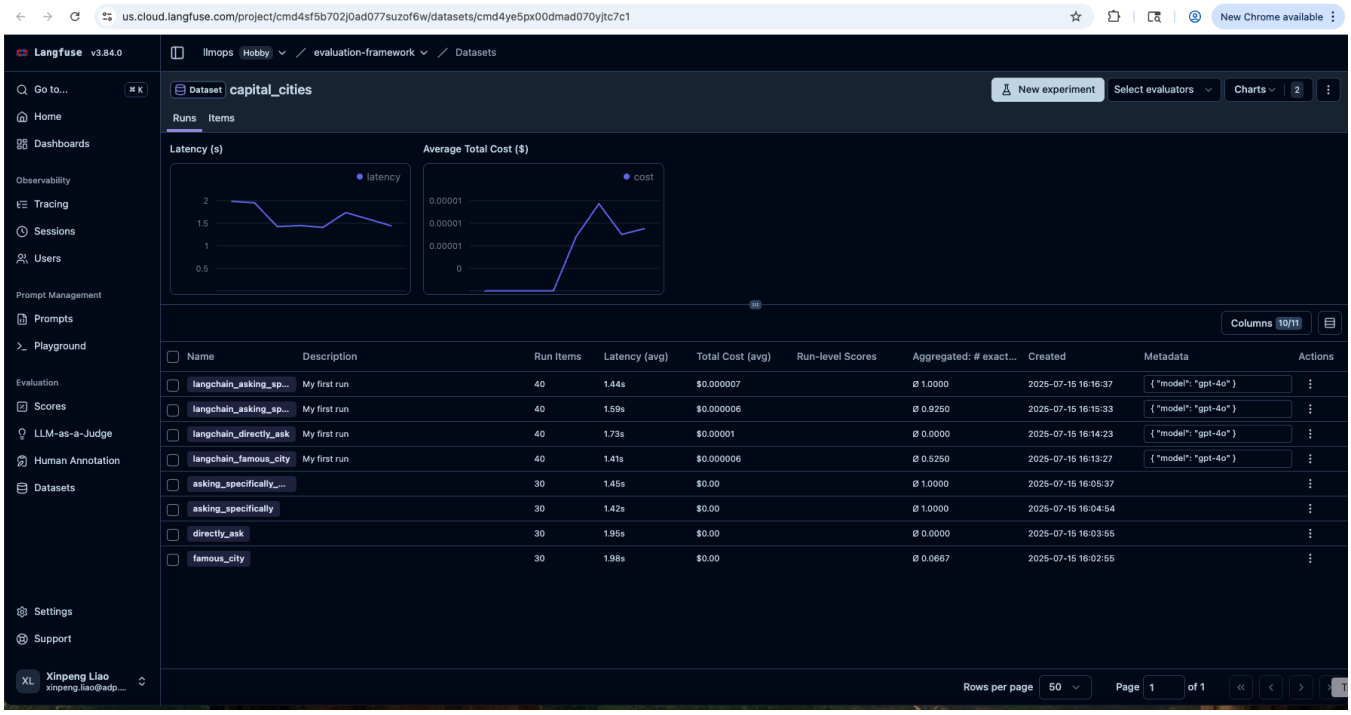

Figure 6 - Langfuse Sampled Datasets

Pros

- Continuous Improvement: Easily generate datasets from real-world production edge cases, enabling targeted improvements to your application.

- Pre-deployment Testing: Benchmark new models or prompt changes against curated datasets before deploying them to production, reducing risk of regressions.

- Structured Testing: Conduct systematic experiments by defining collections of inputs and expected outputs, facilitating repeatable and reliable evaluation processes.

- Flexible Evaluation: Incorporate custom evaluation metrics or leverage automated LLM-as-a-judge methods to assess performance flexibly.

- Strong Framework Integration: Integrates seamlessly with popular frameworks such as LangChain and LlamaIndex, allowing streamlined dataset management within existing workflows.

Cons

- Dashboard Input/Output Truncation: Trace inputs and outputs may be truncated in the Langfuse UI (approximately 10k characters or 1 MB per request), complicating inspection of large or complex data.

- Limited Manual Dataset Upload via UI: Currently, datasets must be imported via API or SDK, as there's no built-in UI support for manual file uploads (e.g., CSV or JSON), potentially slowing workflows for non-technical users.

- No Built-in Throttling or Run Limiting: The platform lacks UI-based controls to throttle dataset processing or limit batch sizes, requiring users to implement custom logic programmatically.

- Lack of Failure Summaries: There is no consolidated view of failed or poorly performing dataset items, forcing users to manually filter and inspect individual records.

- API Rate Limits for High-Volume Ingestion: Cloud-hosted deployments impose dataset API rate limits (e.g., 100 requests/minute on Hobby plans, up to 1,000 requests/minute on Enterprise plans), which can hinder workflows requiring high-frequency data operations. Self-hosted setups bypass this limitation but depend heavily on the user's infrastructure capacity.

Confident AI / DeepEval

Figure 7 - Confident AI Sampled Datasets - Topic Modeling beyond typical dataset management

Pros

- Easy Creation and Versioning: The SaaS platform enables straightforward dataset creation, version control, and management of gold-standard ("golden") examples, streamlining evaluation workflows.

- Flexible Import/Export Options: Supports importing and exporting datasets through CSV, JSON, or API-driven methods, offering compatibility with common workflows and tools.

- Advanced Capabilities (Topic Modeling): Confident AI extends beyond traditional dataset management by offering built-in topic modeling features, enhancing data analysis and insight generation.

- Integration with Evaluation: Seamlessly integrates datasets with robust evaluation processes, providing built-in tools for batch and real-time evaluations.

Cons

- Compatibility Constraints: There are current blockers when using Confident AI's built-in dataset loading functions within certain enterprise environments, such as ADP, potentially limiting operational flexibility.

- SaaS-Dependent: Dataset management features heavily rely on the SaaS offering. There is no self-hosted or offline option, which may not be suitable for teams with strict security or data privacy requirements.

- Limited Customization in Data Processing: While powerful, the prebuilt functionalities and data workflows might restrict teams needing extensive data preprocessing, custom transformations, or more sophisticated versioning logic.

Analytics and Reporting

Arize Phoenix

Phoenix handles Analytics and Reporting by automatically tracking agent performance, user interactions, and task outcomes. It collects data such as response times, user feedback, success rates, and common queries, then provides dashboards and downloadable reports for analysis. This helps developers and stakeholders monitor agent effectiveness and identify trends for improvements.

Pros

- Automated tracking: No manual setup for logging; Phoenix captures key metrics out-of-the-box.

- Actionable insights: Visual dashboards make it easy to spot issues and improvement opportunities quickly.

- Customizability: You can tailor which events or interactions are tracked and reported.

Cons

- Limited deep customization: For very advanced, domain-specific analytics, extra work or external tools may be needed.

- Data latency: Some metrics may not appear in real-time, depending on the reporting intervals.

Langfuse

Langfuse Analytics and Reporting Langfuse offers a powerful, flexible Analytics & Reporting suite for LLM applications, featuring a feature-rich Metrics API, alerting, and comprehensive evaluation tools. It equips teams to monitor cost, latency, and quality, detect regressions, and make continuous improvements. However, maximizing its potential depends on thoughtful configuration and sufficient data volume.

Pros

- Custom report & dashboard generation: With the robust Metrics API, craft tailored reports and dashboards by querying traces, observations, and scores. Customize your insights using selectable dimensions (e.g., model, user, prompt version), metrics (e.g., count, latency p95), filters, and time granularity (hour/day/week/month).

- Performance alerting: Automatically detect regressions in quality or latency by configuring metrics-based alerts, enabling faster operational response.

- Built-in A/B and regression testing: Compare model versions, prompts, or pipeline variants systematically. Just ensure sufficient traffic and data to derive statistically significant conclusions.

- Advanced analytics for NLP pipelines: Go beyond basic metrics with precision, recall, error rates, and latency analysis. This is especially valuable for high-stakes applications.

- Data-driven continuous optimization: Leverage dashboards and evaluation tools (LLM-as-a-judge, user feedback, manual or custom scoring) for iterative model refinement.

Cons

- Dashboard design needs care: Powerful as it is, the Metrics API demands thoughtful construction of views to avoid clutter. A strategic selection of dimensions, metrics, filters, and visuals is essential

- Alert tuning to reduce noise: Default thresholds may result in false positives; fine-tuning is required to balance sensitivity with relevance.

- Statistical validity in A/B Tests: Effective regression testing assumes sufficient data volume; without it, tests could be misleading.

- Initial customization overhead: Setting up meaningful alerts and dashboards requires effort and domain knowledge—but pays off with actionable insights.

Confident AI / DeepEval

Pros

- Integrated KPI Dashboards: Offers built-in dashboards that surface core metrics such as quality, latency, cost, and drift, making it easy to monitor LLM performance at a glance.

- Configurable Performance Alerts: Supports performance alerting to track model regressions or anomalies in production. Critical metrics like latency and cost can trigger alerts when thresholds are breached.

- Experimentation & A/B Testing: Features an experiments dashboard that helps run regression and A/B tests with visible diff reports between prompt or model versions, reinforcing your evaluation and validation workflows.

- Developer-Centric Evaluation Framework: The open-source DeepEval framework integrates easily with test-driven development (e.g., Pytest), allowing embedded, CI/CD-friendly LLM tests and traceability.

Cons

- Batch-Centric, Not RealTime: Optimized for scheduled or batch evaluations rather than live-streaming telemetry; less ideal for systems needing real-time high-traffic monitoring documentation.

- Higher Computational Overhead: Deep, LLM-based metrics (like GEval, DAG, RAGAS) can be resource-intensive, requiring asynchronous execution or dedicated infrastructure.

- Visualization & Alerting Maturity: While Dashboards and alerts are available, ConfidentAI’s UI and alerting experience are reportedly less polished than established ML observability leader.

- Setup and Integration Requirements: Requires developer effort to embed DeepEval tracing and design meaningful evaluation pipelines and alerts before generating actionable insights.

Conclusion

Arize Phoenix shines in early-phase experimentation with minimal setup and strong tracing. It is a powerful open-source tool built with OpenTelemetry that excels during the early stages of development and experimentation. It auto-instruments LLM workflows, capturing spans from frameworks like LangChain and LlamaIndex, allowing easy visualization, dataset creation, and prompt experimentation. It's ideal for prototyping and debugging, especially within teams that already use Arize’s broader platform. However, Arize Phoenix as a newer product is more geared toward development environments, lacks documentation and community support compared to Langfuse.

Langfuse offers end-to-end, production-ready observability with full OpenTelemetry support and A/B evaluation pipelines. It delivers robust, production-grade observability across both LLM and non-LLM components. It supports OTLP ingestion, nested spans, cost tracking, multimodal workflows, prompt versioning, user feedback, A/B testing, and built-in alerting. With polished dashboards and deep integration into developer tools, Langfuse is ideal for complex applications requiring full lifecycle visibility. However, it may require some setup and configuration to self-host and tailor for edge cases.

ConfidentAI / Deep Eval excels in evaluation-heavy workflows with deep quality testing in SaaS, which adds tracing, dashboards, regression tests, and CI /CD integration. However it isn’t included in our project's free-tier testing scope. But its open-source DeepEval can help us write unit-like tests for model outputs using metrics like GEval, RAGAS, and hallucination detection. It’s excellent for ensuring LLM quality before deployment.

As a starting point, we're exploring free open-source versions. Given limitations with number of features provisioned in self hosting by Confident AI / DeepEval, it is decided to drop this framework and continue with Arise Phoenix and Langfuse for further exploration.

Reference

Arize Phoenix Docs

Langfuse Docs

Confident AI Docs

DeepEval GitHub