

An abstract graphic in the top right corner of the slide. It features several thin, curved lines in cyan, yellow, white, and purple. These lines intersect at various points, some of which are marked with small circles of the same color. The lines generally trend from the top left towards the bottom right, creating a sense of flow and connectivity.

Evaluating AI agent applications

Introduction

As teams push to bring AI applications into production, AI agents are taking center stage. However, you can't confidently deploy AI agents without rock-solid evaluations to ensure your applications behave as expected—for both you and your users. Rigorous evaluations are essential because AI agent applications are inherently non-deterministic.

In this whitepaper, we'll guide you through running rigorous evaluations to enhance the performance of AI agent applications—helping you move quickly and deploy with confidence.

While our focus is on evaluating AI agents, the techniques outlined here are equally effective for any LLM-powered application such as chatbots.

How Weights & Biases built the state-of-the-art AI programming agent

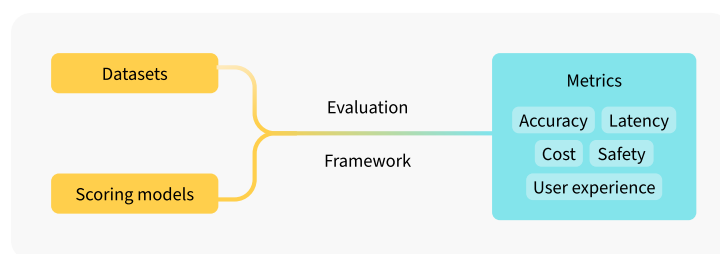
Weights & Biases built the state-of-the-art AI programming agent capable of resolving 64.6% of the issues it encounters using OpenAI's o1 model and W&B Weave. The AI programming agent functions as an autonomous programmer, switching between reading, writing, and testing code—until it determines the issue is solved. The AI programming agent significantly outperformed OpenAI's published results, which relied on a basic agent framework. So what made the difference?

One word: iteration. Using the Weave toolkit for AI agent tracing and evaluations, we made 977 iterations in just 8 weeks before achieving the top ranking. That's over 17 iterations per day. You can learn more in [our blog](#).

What is an evaluation?

When building an AI agent, your goal is to test it rigorously to achieve the best possible output. Traditional test cases and “vibe checks” that work well for software validation are insufficient for the complexities of AI applications. This is why evaluations are a critical component of the AI agent development workflow.

Evaluations enable you to measure and iterate on your AI agent’s performance. By establishing an evaluation framework and scoring tools, you can assess the impact of improvements across multiple dimensions, such as accuracy, latency, cost, safety, and user experience. An evaluation framework aggregates scores for each



evaluation, allowing you to compare results side by side. Additionally, it enables you to drill down into individual examples within an evaluation to identify specific areas—such as prompt structure or model configuration—that need refinement. This systematic approach helps ensure continuous improvement and readiness for deployment.

Why rigorous evaluations are critical

Traditional software testing methods fall short when applied to AI applications for several key reasons. At the core of this difference is the non-deterministic nature of Large Language Models (LLMs), which power AI applications. Unlike traditional software, where conditional logic is sufficient to ensure predictable outputs, LLMs can produce varying responses to the same input. This variability necessitates a new approach: evaluations for

LLMs function as unit tests for AI applications, enabling developers to pinpoint areas for improvement, enhance consistency, and ensure accurate, reliable responses for all users. Consider how application testing has evolved. Building a simple AI chatbot is now a quick and accessible task. What once required significant time and resources can now be accomplished with no-code/low-code tools, basic Python scripts, or even code generated by ChatGPT.

Without proper evaluations, AI assistants could negatively impact the customer experience

The challenge lies beneath the surface. While the chatbot may look polished and perform adequately in limited scenarios, its behavior might occasionally be unpredictable and, in some cases, potentially harmful. Preparing such an AI application for real-world use requires rigorous hardening and testing. Robust evaluation is necessary to ensure the chatbot delivers accurate, credible responses in diverse scenarios. Relying on informal “vibe checks,” no matter how frequent, cannot guarantee a production-quality application. Without systematic evaluation of common, corner, and edge cases, developers lack the visibility needed to understand and control their agent’s behavior.

The need for rigorous evaluation doesn’t end after initial testing. Changes to the AI application or updates to underlying LLMs requires continuous monitoring and evolving evaluation strategies to ensure the AI application is reliable, accurate, and safe in production. Let’s examine the implications of this fundamental difference between traditional software development and AI agent development.



Hi, valued customer! What can I help you with today?



How long do I have to return a laptop computer purchased last week?



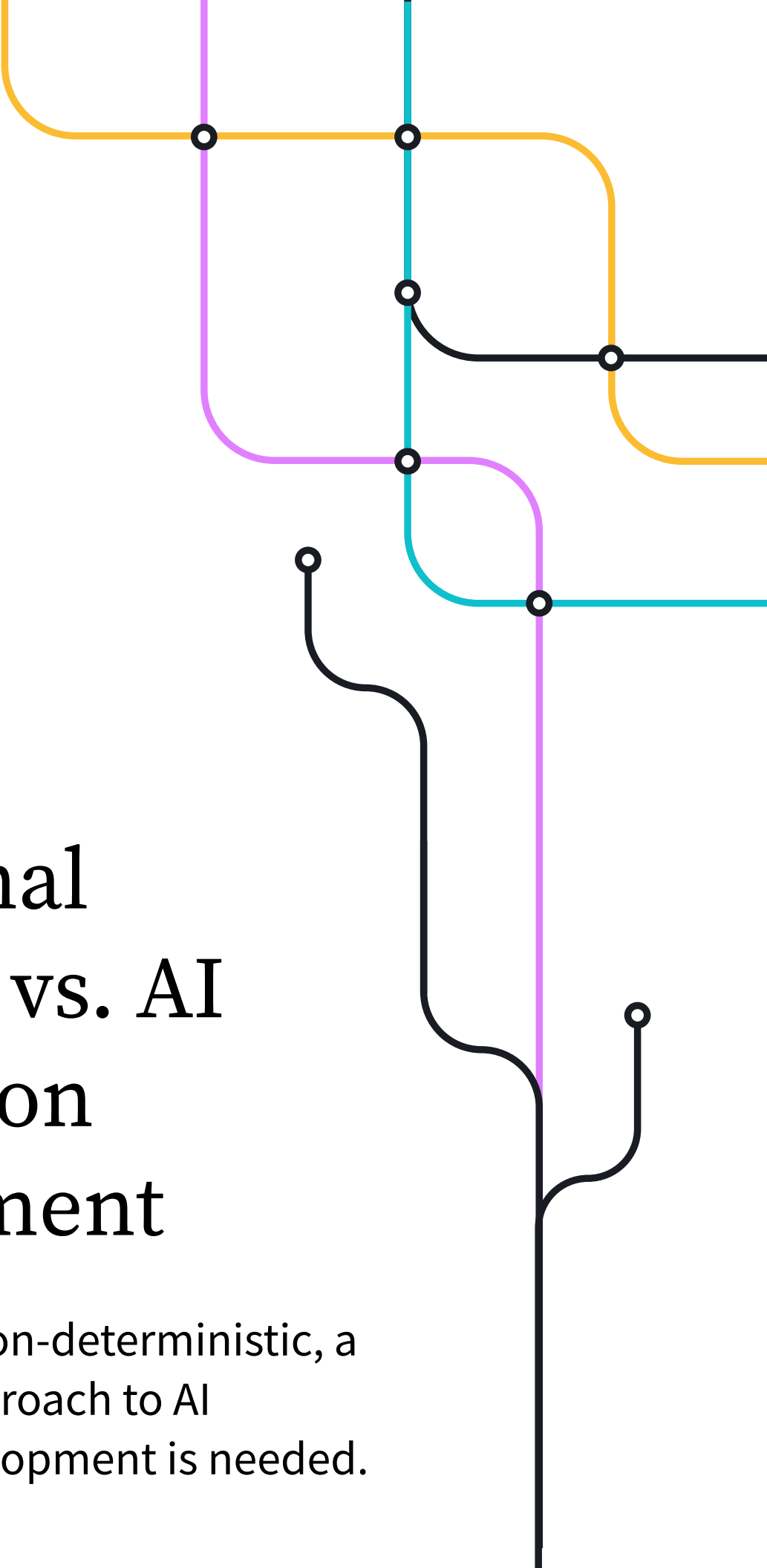
Please refer to the return policy on our website.



Will it help if I provide the order number on the receipt?



I'm sorry. I don't understand your question.

An abstract graphic in the top right corner of the slide. It features several colored lines (yellow, purple, teal, and black) that run vertically and horizontally, connected by small white circles. The lines have rounded corners and some end in small circles, creating a network-like structure.

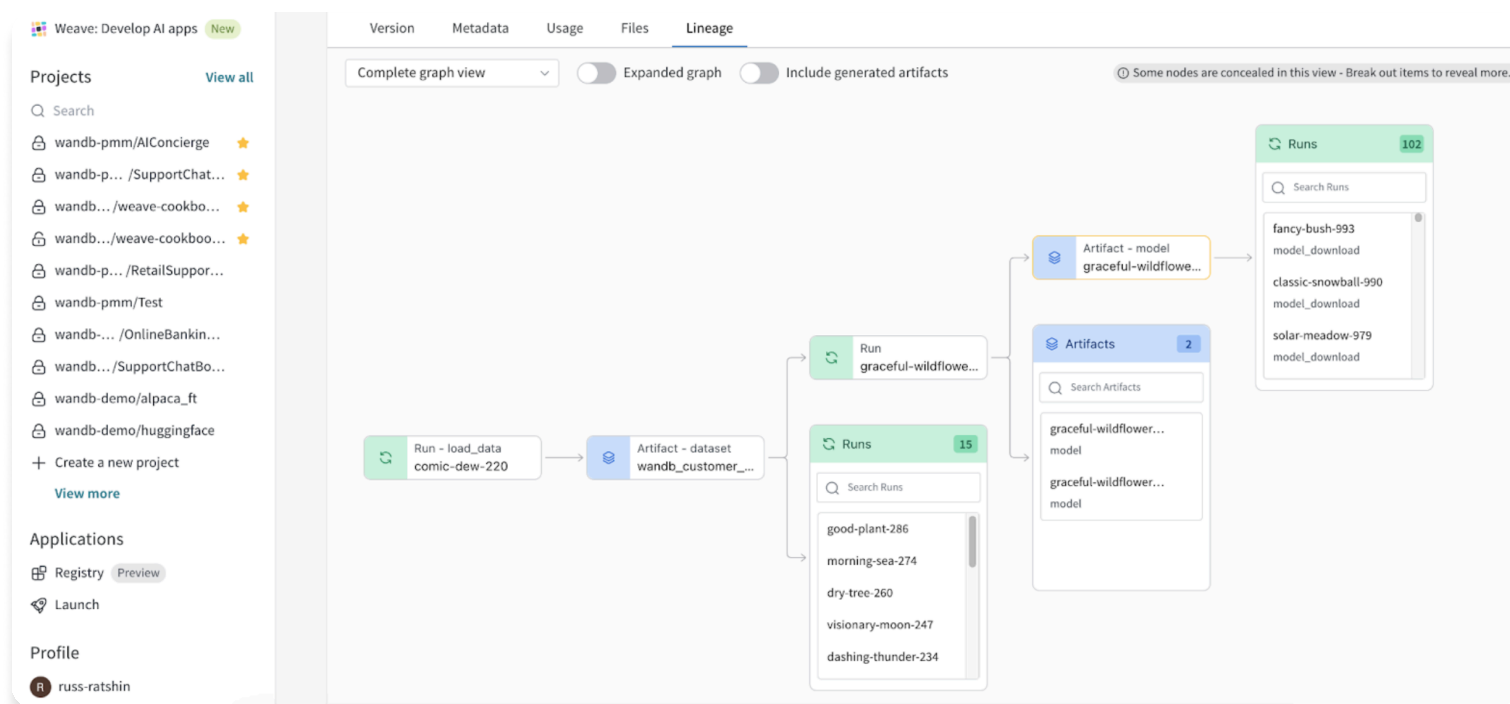
Traditional software vs. AI application development

Since LLMs are non-deterministic, a new iterative approach to AI application development is needed.

Reproducibility

When your agent's performance fluctuates—whether it “feels” worse this week or better next month—it's essential to maintain clear records of what was modified. This includes model updates, prompt or code changes, and dataset revisions. Detailed logs enable you to identify the changes, evaluate their impact, and decide whether to roll back or proceed.

Additionally, when collaborating with team members, the ability to reproduce the exact agent version, configuration, and code is crucial for building on each other's work. This level of reproducibility is equally vital for troubleshooting production issues, such as consistent failures or hallucinations, ensuring a systematic approach to identifying and resolving problems.



Track model lineage

Measuring incremental improvements

“Vibe checks” cannot capture small but crucial gains. For instance, consider adopting a new LLM expected to outperform your current model. An accuracy improvement from 70% to 73%, for example, may not be noticeable when casually testing prompts.

This could lead you to mistakenly conclude there was no improvement, causing you to miss out on incremental advances that, over time, add up to significant gains. Data-driven evaluations are essential to detect these subtle gains and ultimately achieve production-level quality.



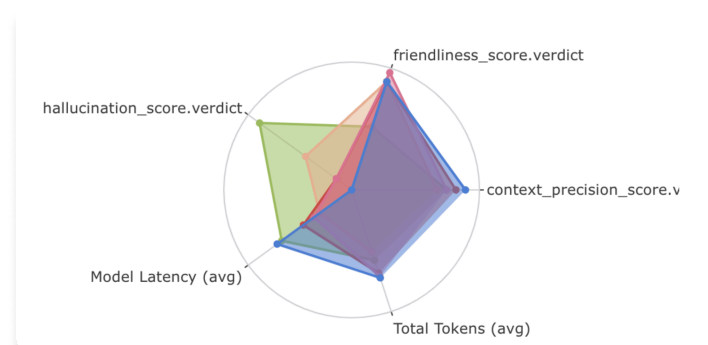
Evaluate AI applications across key metrics

Multiple dimensions of performance

AI agents are evaluated across multiple dimensions, including accuracy, cost, latency, and safety. Even within a single dimension, the definition of metrics can vary based on the use case. For instance, the concept of accuracy differs significantly between a customer service agent and a healthcare assistant. Accuracy may be assessed using various metrics such as recall, perplexity, or precision, which organizations tailor to specific business contexts. Therefore, agents are typically measured against dozens, if not hundreds, of metrics.

A common pitfall is focusing on a single metric while neglecting others, leading to regressions in performance elsewhere. For example, an effort to improve accuracy by using a larger context window in an LLM might inadvertently slow down latency due to the increased computational load. Without thorough evaluation, this latency regression could go unnoticed, especially if the testing focuses solely on accuracy. Over time, the impact may compound, with users experiencing slower response times.

By the time the issue becomes evident, it may be nearly impossible to pinpoint the change that caused the regression, resulting in time-consuming debugging and rework. Achieving improvements in one metric without negatively impacting others requires careful and systematic evaluation. Given the complexity and scale of these measurements, rigorous evaluations are essential to ensure progress remains linear, building on prior advancements rather than stagnating in repetitive cycles.



Assess how improving one metric affects others

Protecting reputation and users

Releasing an underperforming agent can severely damage user trust and harm your brand. There are countless examples of overhyped AI launches that failed to meet user expectations. Many of these were promising ideas, but they were released prematurely without the necessary testing.

Beyond safeguarding the company's reputation, evaluations are also critical for protecting users. They help ensure users are shielded from risks such as the leakage of private or personally identifiable data and the generation of harmful content. Rigorous testing not only builds trust, but also establishes a foundation for long-term success and user safety.

Compliance and regulatory audits

Thorough evaluation logs serve as an essential audit trail if you need to prove past performance. This is particularly critical in industries with strict regulatory requirements. In the event of a severe production issue—such as excessive model hallucinations or inaccurate outputs—you must be able to trace the lineage of the model used, including fine-tuning metadata, datasets, system prompts, and the specific tests conducted with that

Reducing single points of failure

In AI applications, your intellectual property extends beyond the final version of the code or model weights—it encompasses the insights gained from the numerous failed versions that informed the final iteration. A rigorous evaluation platform captures this valuable knowledge in a centralized repository, enabling others to build on it effectively.

When key engineers who conducted the original experiments leave, you don't want your intellectual property or institutional knowledge to leave with them. Clear, well-documented evaluations mitigate this risk, make onboarding new team members faster, and enhance business continuity.

version of the application. This allows you to audit whether the validation process was sufficient or needs improvement. As AI regulations evolve, this level of traceability will likely become mandatory, with companies expected to demonstrate proof of rigorous testing and quality assurance to satisfy regulatory standards.

Benefits of rigorous evaluations



Speed

Reduces the time it takes to move from idea to production

1



Confidence

Mitigates regressions and ensures key metrics remain stable

2



Iteration

Quickly test and compare different configurations without duplicating work

3



Collaboration

Centralized evaluations enable reuse

4



Standardization

Consistent evaluations and benchmarks

5

A well structured evaluation framework makes it easy to compare and improve agent performance and support governance.

Anatomy of a rigorous AI agent evaluation

A rigorous evaluation consists of three key components.

1. Agent

An agent is the specific version of the application being tested, encompassing all its components, including LLMs, prompts, RAG pipelines, guardrails, and configuration metadata (e.g., temperature settings).

2. Dataset

A dataset is well-curated data that reflects real-world use cases and tests various scenarios. It typically includes examples—often highlighting failure cases—to evaluate the agent. There are two primary types of datasets:

- a. Questions only: These datasets are used when evaluators generate numerical scores, such as context relevancy or adherence to instructions. If you are building the scorer on your own as opposed to using pre-built scorers from Weights & Biases or third-parties, you will need a separately labeled ground truth dataset to train a classifier or fine-tune an LLM, depending on the technology powering the scorer.
- b. Questions and expected answers: These datasets are used in programmatic evaluators that compare the agent's output against a ground truth to produce a binary pass/fail result.

In both cases, expert-labeled ground truth data is critical for defining what “good” looks like and ensuring automated and model-assisted tests align with human judgment. An ideal process for creating ground truth data involves experts reviewing outputs, providing structured feedback, and refining evaluation methods so they align with expert standards.

- Bravotv: A television network that focuses on reality TV shows, including popular franchises like The Real Housewives.
- BravoWWHL: Stands for Bravo's Watch What Happens Live, a late-night talk show hosted by Andy Cohen that features celebrity interviews, games, and discussions about Bravo's reality TV shows.
- Paris Hilton: A well-known American socialite, businesswoman, and media personality, known for her appearance on the reality TV show The Simple Life and her work as a singer, actress, and entrepreneur.

< Vars

tweet_full_text=YOU STOLE MY GODDAMN HOUSE
#justdoit #juststealit @Bravotv @BravoWWHL
@ParisHilton <https://t.co/8xShKr0SYq>

Prompt >

You will be doing named entity recognition (NER).
Extract up to 3 well-known entities from the following tweet:

YOU STOLE MY GODDAMN HOUSE #justdoit
#juststealit @Bravotv @BravoWWHL @ParisHilton
<https://t.co/8xShKr0SYq>

For each entity, write one sentence describing the person or entity.

👍 GOOD

👎 BAD

3. Scorers (Evaluators)

Scorers are tools or methods used to quantify performance metrics such as accuracy, latency, and safety. The evaluation process begins by selecting the appropriate criteria and metrics, followed by building the scorers to generate these metrics. There are two main types of scorers. By combining these methods, scorers ensure comprehensive and reliable evaluation across diverse performance dimensions.

1. **Automated Checks:** These are fast, programmatic tests that rely on clear pass/fail criteria, such as format validation or checks for required fields.
2. **Model-Assisted Evaluations:** These can use LLMs to evaluate outputs or specialized classifier models trained with algorithms and expert-annotated ground truth data. Model-assisted evaluations are particularly suited for assessing subjective criteria, such as response accuracy, conciseness, and privacy compliance.

W&B Weave provides a flexible and robust framework for running evaluations

01

Use pre-built scorers from Weights & Biases or bring your own. Weave integrates seamlessly with third-party or custom (homegrown) scorers.

02

Build and maintain high-quality datasets derived from production traces for both improving evaluations and fine-tuning underlying LLMs.

03

Weave offers powerful visualizations, automatic versioning, leaderboards, and a playground to precisely measure and rapidly iterate on improvements.

04

With Weave, you can centrally track all evaluation data to enable reproducibility, lineage tracking, and collaboration.

The screenshot displays the Weave interface, which is used for managing and evaluating machine learning models. The interface is divided into two main sections: a list of traces on the left and a detailed view of a specific trace on the right.

Traces List (Left Panel):

Trace	Feedback	Status	callback	frequency_p
process_chat_message	4b85	🟢	<function update_c...	N/A
process_chat_message	e68a	🟢	<function update_c...	N/A
process_chat_message	bddc	🟢	<function update_c...	N/A
weave.completions_create	3382	🟢	N/A	
weave.completions_create	4c93	🟢	N/A	
weave.completions_create	8e15	🟢	N/A	
weave.completions_create	e919	🟢	N/A	
weave.completions_create	9241	🟢	N/A	
weave.completions_create	57d7	🟢	N/A	
memory-list_all_memories	65f8	🟢	N/A	N/A
process_chat_message	a4a8	🟢	<function update_c...	N/A

Trace Detail View (Right Panel):

The detailed view shows the execution of a specific trace, `process_chat_message`, with a status of `🟢`. The trace is associated with the `winston-solve` function, which has a status of `🟢` and a cost of `$0.0089`.

Inputs:

Path	Value
messages	
0	
role	user
content	make me a photo realistic image of Martin Ødegaard dribbling a soccer ball, on a solid red background the with the arsenal logo
callback	<function update_chat at 0x10309c...

Output:

Path	Value
output	
type	plan

An abstract graphic on the right side of the page consists of several vertical and horizontal lines in yellow, purple, teal, and black. These lines are connected by small white circles with black outlines, creating a network-like structure that flows from the top right towards the bottom right.

A practical recipe for running evaluations

Having examined the ingredients necessary for rigorous evaluations, it's time to see how these elements come together in practice. Let's take a look at the recipe in detail.

Step 1: Define success criteria

The foundation of any rigorous evaluation begins with a clear definition of success. This involves specifying the requirements your AI application must meet, establishing acceptable performance thresholds, and identifying critical failures. For instance, in a customer service agent, success might mean accurately answering questions using caller data, while failures could include delivering generic responses or exposing private information. A well-defined success framework ensures the evaluation process aligns with your business goals and user expectations. Incorporating a survey of common metrics for accuracy, latency, cost, and safety provides a structured approach for measuring success and helps you assess performance with consistency and confidence.

Step 3: Create a robust scoring system

Establish performance baselines to measure improvements across future iterations of your agent. Previously, we explored various types of scorers, including using LLMs themselves as judges to evaluate agent outputs for quality, relevance, and adherence to instructions. For domain-specific requirements, specialized machine learning models can be trained to conduct deeper accuracy checks and assess subjective criteria. Both scorers and evaluation metrics should be updated iteratively to foster continuous improvement while maintaining the stability and performance of the production agent.

Step 5: Analyze evaluation results

The results of evaluations serve as a guiding light for selecting the best candidates and techniques. By tracking results in leaderboards, teams can compare different LLMs, identifying the most suitable model for their needs. Similarly, comparing evaluation methods enables the refinement of scoring approaches, ensuring the chosen techniques are robust and effective. This iterative analysis not only enhances the application's performance but also provides a blueprint for future improvements.

Step 2: Comprehensive eval suite

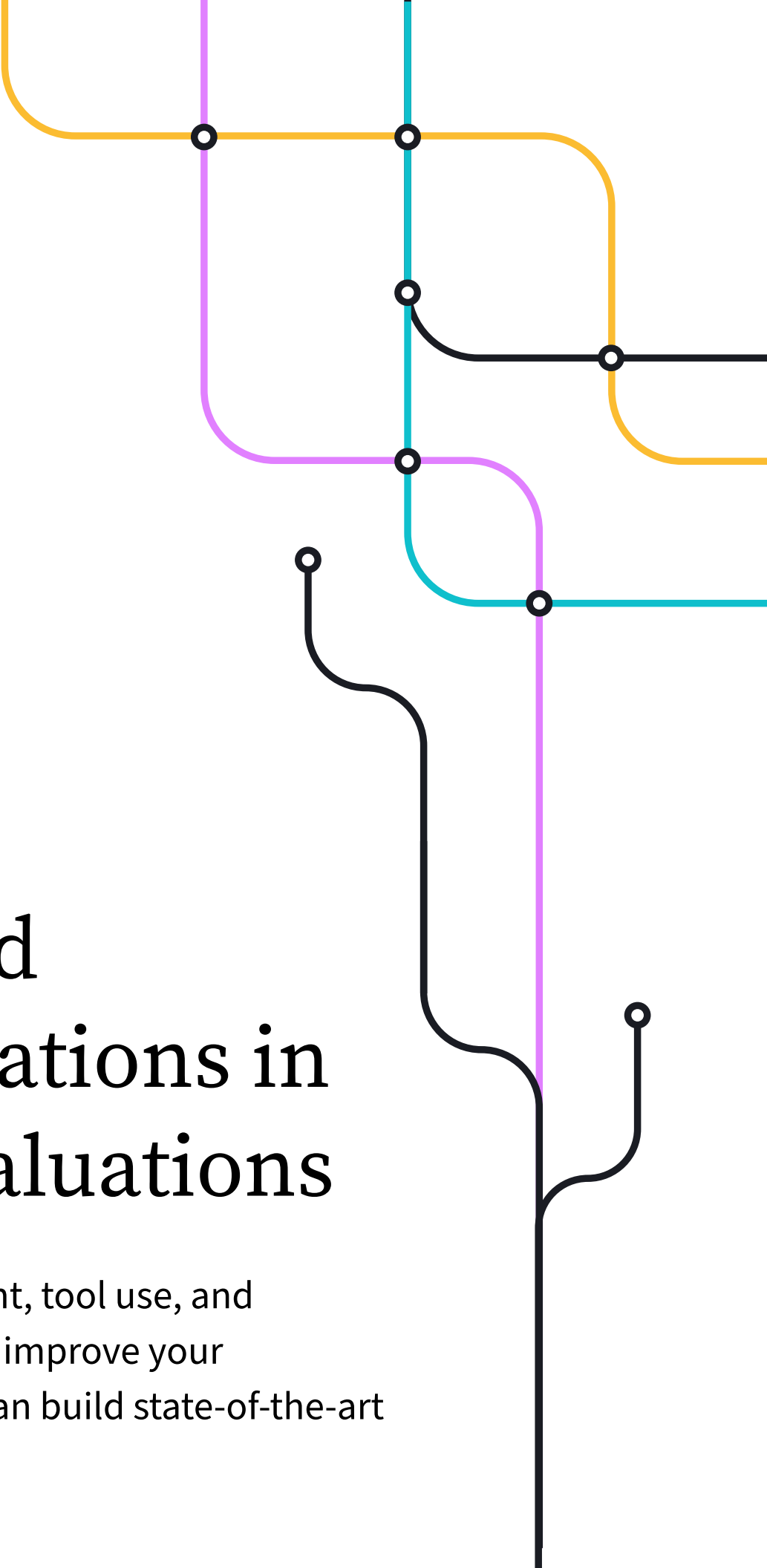
An effective evaluation suite is multifaceted, combining automated and manual approaches. Automated checks are indispensable for validating clear rules, such as format consistency or compliance with predefined requirements. These checks provide fast, repeatable results that can be easily scaled. For nuanced cases, however, expert review is critical. Specialists can assess complex outputs, ensuring that the system's behavior aligns with both domain-specific standards and user expectations. To maintain consistency and traceability, version control should be applied to evaluation code, allowing for iterative improvements without losing historical context.

Step 4: Build a high-quality dataset

A high-quality dataset is the backbone of any evaluation process. Real-world user feedback is invaluable, offering insights into usage patterns and pain points. Synthetic data generated by LLMs can supplement this, with scorers used to filter and identify high-quality examples for inclusion. Expert annotations are crucial for handling domain-specific scenarios, ensuring the dataset accurately reflects the real world. Continuous editing and curation removes noise, captures edge cases, and maintains relevance. Incorporating production traces—such as logs and failure modes—adds authenticity, addressing real-life complexities that might otherwise go unnoticed.

Summary

A well-executed evaluation strategy, following this recipe, ensures your AI agent meets high standards of quality, reliability, and safety while continuously adapting to evolving requirements and user needs.

An abstract graphic on the right side of the page consists of several colored lines (yellow, purple, teal, and black) that form a network of paths. These paths are connected by small white circular nodes. The lines have rounded corners and some end in small open circles, creating a sense of flow and connectivity.

Advanced considerations in agent evaluations

Memory management, tool use, and planning can further improve your evaluations so you can build state-of-the-art agents quickly.

Memory management

Memory management helps the agent retain context over long conversations or large data sets. Memory management for agents involves structuring, storing, and retrieving relevant information to optimize performance, minimize resource consumption, and ensure an effective user experience. Proper memory management ensures that agents are efficient, responsive, and scalable while maintaining user privacy and regulatory compliance. Key metrics include:

- **Recollection accuracy:** Measures the agent's ability to accurately recall and reproduce past interactions or experiences.
- **Relevance:** Assesses how pertinent the recalled information is to the current context or task.
- **Faithfulness:** Evaluates the degree to which the recalled information aligns with the actual events or stored knowledge, ensuring the information is not fabricated or distorted.
- **Generalization:** Evaluates how well the agent can apply learned experiences or memories to new, unseen scenarios.
- **Compression ratio:** Looks at the agent's ability to summarize or compress long-term memories effectively without losing critical information.
- **Latency:** Examines the time it takes for the agent to retrieve or use memory when needed.
- **Memory Usage:** Tracks how efficiently the memory system uses resources, focusing on storage space and computational requirements.
- **Forgetting rate:** Monitors the agent's ability to forget unimportant information over time while retaining essential details.

Tool use

Evaluating an AI agent's ability to effectively use different tools and functions involves assessing its ability to invoke, interact with, and derive value from external resources (e.g., APIs, libraries, databases, or services). This evaluation ensures the agent integrates tools correctly, selects the right ones for specific tasks, and optimizes their use. Key metrics include:

- **Success rate:** Measures the percentage of tool or function calls that are successfully executed without errors.
- **Error rate:** Indicates the frequency of failed or incorrect tool or function invocations.
- **Latency:** The time taken for the agent to invoke a tool or function and receive a response.
- **Context integration:** Evaluates the agent's ability to incorporate relevant contextual information into tool inputs for better outputs.
- **Tool selection accuracy:** Assesses how often the agent selects the most appropriate tool for the given task.
- **Resource utilization:** Assesses the computational and memory resources consumed during tool or function usage.
- **Task completion time:** The total time required to complete a task that involves invoking tools or functions.
- **Output accuracy:** Evaluates how correct the outputs generated by the tools or functions are.
- **Relevance of results:** Measures how closely the tool's output aligns with the intended task or user query.
- **Fallback success rate:** Measures the agent's ability to recover and succeed using alternative approaches when a tool or function fails.

Planning

Assessing multi-step reasoning or task decomposition in an agent involves evaluating the agent's ability to break down complex tasks into manageable sub-tasks, solve them systematically, and deliver coherent results. It's important to evaluate multi-step reasoning and task decomposition capabilities of your AI agent to ensure reliable performance. Key metrics include:

- **Goal achievement rate:** Measures the percentage of plans that successfully accomplish the intended objectives or goals.
- **Action optimality:** Evaluates whether the actions in the plan represent the most efficient path to achieving the goal.
- **Planning time:** Assesses the time taken by the agent to generate a complete plan from the initial input.
- **Dynamic replanning time:** Tracks how quickly the agent can adapt and create a new plan in response to changes or disruptions.
- **Plan feasibility:** Measures the practicality and executability of the generated plan in the given environment or context.
- **Error recovery rate:** Indicates the agent's ability to revise and adjust its plan to recover from errors or unexpected situations.

Multi-agent systems

Assess the ability of agents to collaborate effectively, share information, and complete tasks collectively. Evaluations ensure the agents operate efficiently, minimize conflicts, and deliver coherent outcomes. Key metrics include:

- **Task completion rate:** The percentage of tasks successfully completed through coordinated efforts among agents.
- **Inter-agent alignment:** Measures the degree of coherence and synchronization between the actions and decisions of multiple agents.
- **Conflict resolution rate:** The frequency with which agents successfully resolve conflicts or overlapping responsibilities during collaboration.
- **Dependency management success:** Assesses how effectively agents handle interdependencies between tasks assigned to different agents.
- **Plan consistency:** Evaluates whether individual agent plans align and integrate into a cohesive overall strategy.
- **Error propagation rate:** Measures the extent to which errors in one agent's output affect the performance of other agents in the system.
- **Scalability:** Assesses the system's ability to maintain coordination efficiency as the number of agents or task complexity increases.

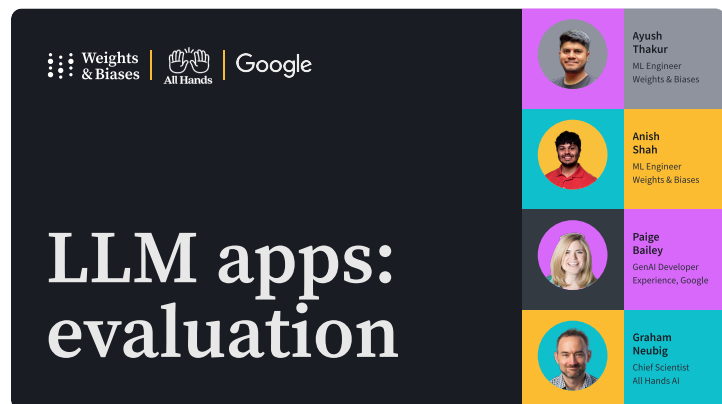
Human-agent interactions

Evaluating human-agent interactions in an agent ensures the agent delivers a positive, effective, and user-centered experience. Key metrics include:

- **Context tracking score:** Evaluates the agent's ability to remember and effectively utilize conversation history, ensuring responses remain coherent across multiple turns.
- **User turn-to-repair ratio:** Compares the number of user utterances spent clarifying or correcting the AI to total user utterances, revealing how often the user must "fix" the conversation.
- **Missed clarification rate:** Tracks the frequency with which the agent fails to ask a question when it should have, resulting in confusion or errors, identifying under-questioning behavior.
- **Over-questioning score:** Quantifies instances where the agent's questions are excessive or irrelevant, highlighting when it inappropriately prolongs the interaction with unnecessary queries.
- **Non-redundant query ratio:** Looks at how many questions are not repeated or rephrased unnecessarily, ensuring the agent avoids looping or posing the same query in different words.

Resources to get started

Explore Weights & Biases courses, consulting, and more to get started with AI agents quickly.

**01**

Free online course: Created in collaboration with experts from Google, All Hands AI, and Weights & Biases, this [free online evaluations course](#) is the easiest way to get started.

02

Watch our agents webinar: In this [webinar](#), gain valuable insights into all things agents, including a detailed look at evaluating them. Our AI expert explains how to build and evaluate an agent.

03

Run a Proof of Concept (POC): Begin with a small-scale pilot and scale swiftly as value is demonstrated.

04

Attend a Weights & Biases workshop: Bring your use case to an expert session for hands-on guidance in designing and running evaluations.

05

Engage our experts: [Weights & Biases AI Advisory Services](#) provide specialized consultative help and guidance in developing and productionizing your agent.

Conclusion

2025 will be the year AI meets business. Agents offer unprecedented capabilities, but the biggest challenge remains effective evaluations. With our expertise—derived from building AI workflows for OpenAI, Meta, NVIDIA and the likes—we're here to help you set up a robust evaluation process that enables faster, more confident deployments. Don't wait—contact us for a [free demo](#) and a deep dive into your agent use case.