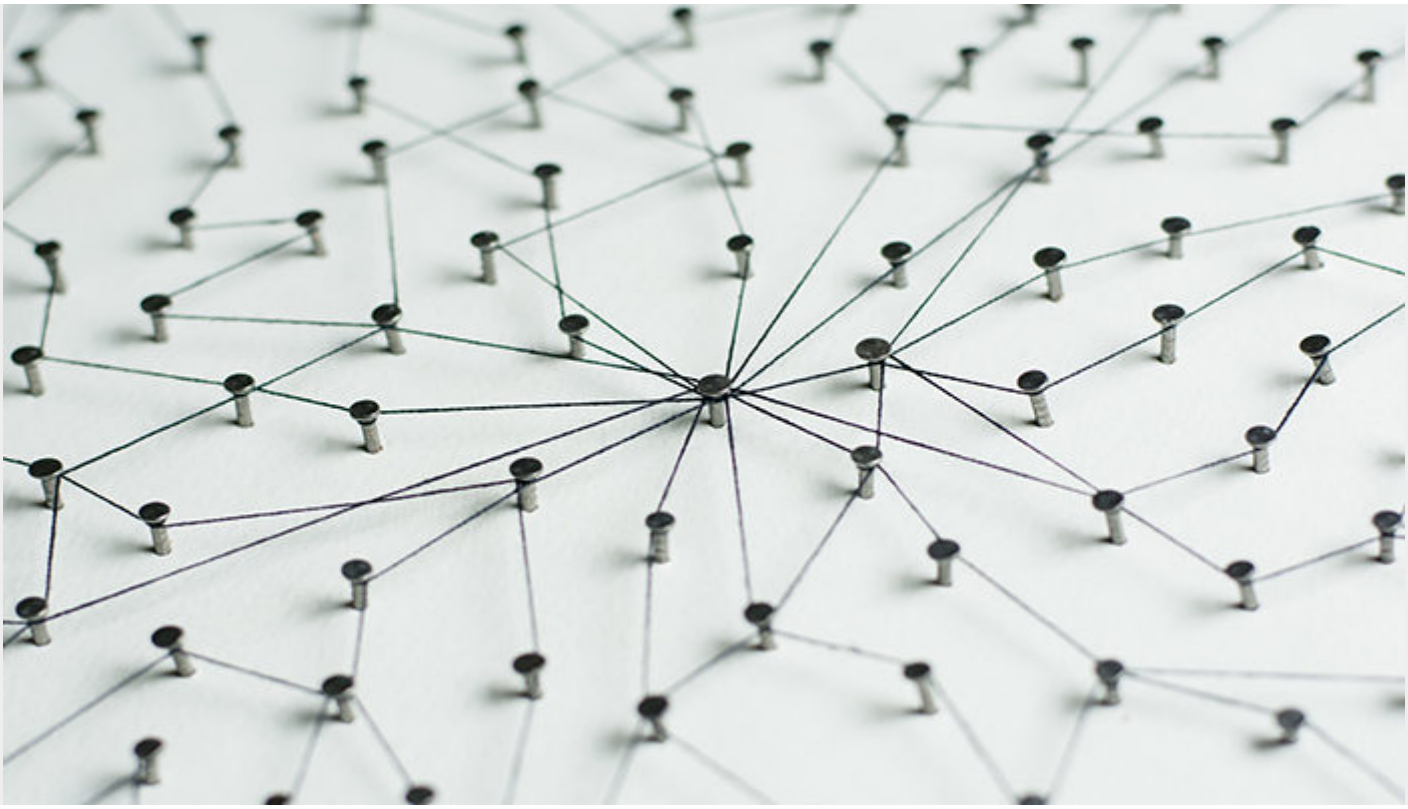
BMC HELIX LLM OBSERVABILITY HELPS OPTIMIZE MODEL TRAINING AND APPLICATION PERFORMANCE



We have seen an explosion in large language models (LLMs) and generative AI (GenAI) techniques over the past two years, and the transformative power they bring to a wide range of applications is only beginning to be realized. For IT operations (ITOps), it's hard to ignore the significant advantages that individuals and businesses can gain—from generating incident summaries and pinpointing root causes to recommending remediation steps. However, running LLMs in production and at an enterprise scale can be a challenge for IT, developers and data engineering teams as they tackle:

- **Computational resources:** Training and running large LLMs require significant computational power, which can be expensive.
- **Scale:** LLMs generate large volumes of data and may suffer from model drift. Monitoring this volume of data at scale is no easy task.
- **Performance**: LLM applications are complex, and it's not always easy to identify request errors or latency bottlenecks.
- **Quality and accuracy**: Is the LLM application performing as expected and generating high-quality outputs?
- **Bias, toxicity, and hallucinations:** These models can generate fabricated content by blending fact and fiction in their outputs, or generate biased or offensive content.

BMC Helix's new LLM observability functionality helps organizations solve these challenges. It provides LLM application workflow tracing and dashboards to help data scientists and AI engineers monitor model quality and efficacy while enabling IT and developers to better understand LLM application performance and behavior.

Improving LLM model quality and efficacy

The BMC Helix LLM observability dashboards help data scientists and AI engineers monitor the model training metrics that impact its quality and efficacy. These metrics provide insights into model accuracy, detect drift, and measure and reduce hallucination. The configurable dashboard allows users to add new training metrics as needed.

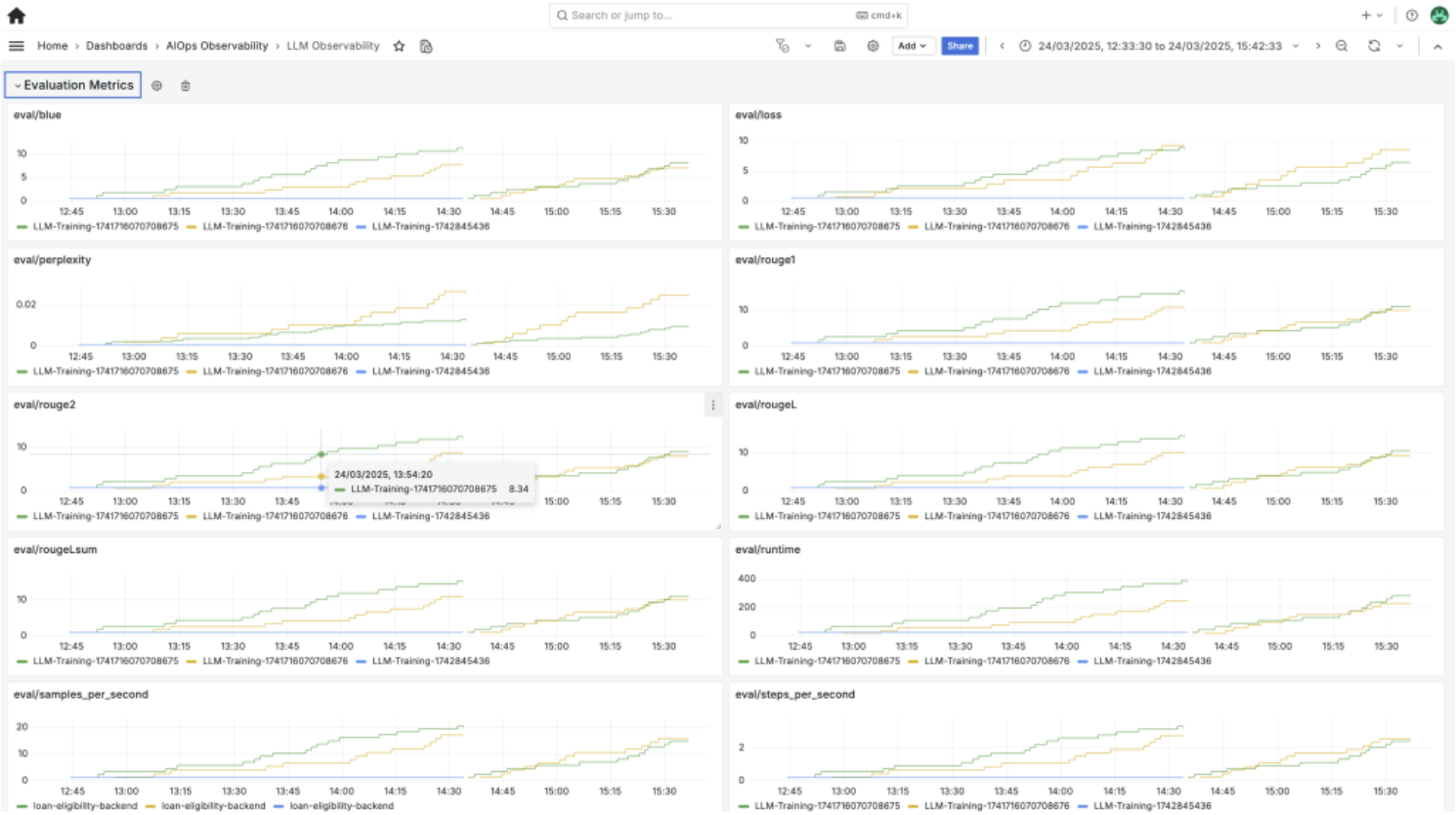


Figure 1. LLM model quality and evaluation metrics and evaluation metrics

Understanding key training metrics

LLM training metrics quantify and assess the performance of a language model, enabling develops and AI engineers to identify strengths and weaknesses, optimize training, and ensure the model meets quality standards for production applications.

Key training metrics to monitor include:

Quality and accuracy

- **Loss**: Shows how much the model's predictions differ from the actual answers. A smaller number means the model's predictions are closer to the truth.
- **Valid_mean_token_accuracy**: Shows the percentage of individual words (tokens) the model got right on the validation set. A higher percentage means better accuracy.

- **Bleu:** This score checks how similar the model's generated text is to one or more reference texts by comparing groups of words. It's used to evaluate translations.
- **Perplexity**: Measures how confidently the model predicts the next word in a sentence. Lower perplexity means the model is more certain and makes better predictions.
- **RougeLsum**: This metric is used for text summarization. It looks at how much the model's summary overlaps with a reference summary, focusing on the longest matching sequence of words.

Hallucination

- **Token accuracy and BLEU**: If the model starts "hallucinating" (making up details that aren't in the input), the percentage of correctly predicted tokens (token accuracy) may drop. Similarly, the BLEU score—which checks how similar the output is to a correct reference—will be lower if the generated text includes extra or made-up content.
- **Perplexity**: When the model is unsure and starts generating odd or irrelevant words, its perplexity (a measure of confidence in predicting the next word) will increase. Higher perplexity means the model is less certain about its predictions, which can be a sign of hallucinations. An increase in perplexity over time shows that the model is becoming
- **ROUGE-Lsum (for summarization):** In tasks like summarization, if the model adds information that isn't in the source or misses key details, the ROUGE-Lsum score will fall because it measures the overlap between the generated summary and a reference summary.

Drift

- **Loss:** Drift refers to a gradual change in the model's behavior over time. If the loss value (which shows how far the predictions are from the true answers) starts increasing, it suggests the model is drifting and its predictions are getting less accurate.
- **Token accuracy, BLEU, and ROUGE scores**: If these scores drop over time, it means the model's output is moving away from the expected results. This steady decline is another sign that the model is drifting from its original performance or training data.
- **Perplexity**: An increase in perplexity over time shows that the model is becoming less confident in its predictions, which can be another indicator of drift.

Optimizing GPU costs

The BMC Helix LLM observability dashboard also provides system metrics related to GPUs at training time. GPU metrics monitor the GPU's real-time performance, such as memory usage, speed, and temperature, to ensure that the computational resources are being used efficiently and not overloaded.

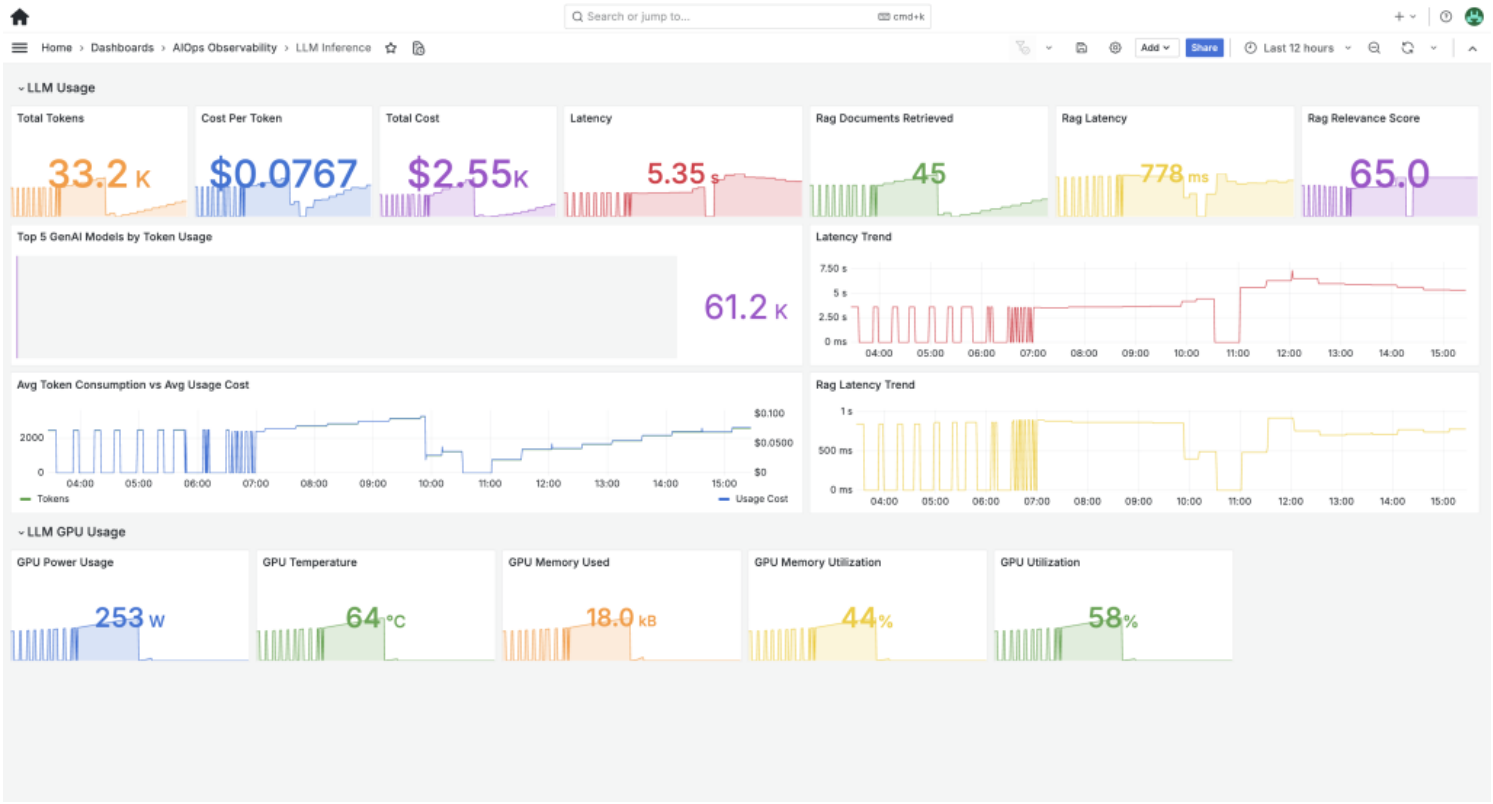


Figure 2. LLM and GPU usage and cost metrics

Key metrics to monitor LLM applications in production

The BMC LLM observability function uses a combination of OpenTelemetry and OpenLLMetry to collect and transmit metrics and traces from LLM applications to understand the sequence of events, making debugging and root cause analysis simpler and more effective. That means we can work with any LLM application as long as we're getting the data in the OpenTelemetry Protocol (OTLP) format.

Once your LLM application is instrumented with OpenTelemetry or OpenLLMetry, you can visualize and analyze the data to better understand your application's operational performance and behavior. See our documentation to help you get quickly started.

Understanding key signals

While capturing API calls to LLMs through traces is crucial for debugging and root cause analysis, it's also important to analyze metrics such as requests, tokens, and costs to optimize performance and manage costs.

Key signals to understand and monitor include:

- **Total successful GenAI requests:** Tracks the total number of times a user or application interacts with a GenAI model by submitting a prompt or request.
- **LLM request rate:** Quantifies how many requests are being made to an LLM model over a given timeframe to help identify bottlenecks and areas where the LLM might be struggling to handle requests.
- **LLM cost:** Helps users estimate the cost of using LLM APIs. LLM cost is measured by tracking

token usage and applying relevant pricing per token or the specific model and provider.

- **Total successful vector DB requests:** Tracks the total number of times a LLM application successfully sent a request to the database and the database processed it without any errors, returning the expected data or confirmation. A successful request ensures that your application can rely on the vector database to perform its intended tasks, like storing data, retrieving information, or making predictions.
- **Vector DB request rate:** Measured as queries per second (QPS), this is a crucial performance metric. Higher QPS values imply better query processing capability and system throughput.
- **LLM latency:** Measures the time delay between a user's input and the model's response, impacting the perceived speed and efficiency of LLM applications.
- **LLM tokens total:** The number of tokens a model can process at a time— its context window—directly impacts how it comprehends, generates, and interacts with text.
- **Total usage tokens:** Each token processed by the LLM incurs a cost, making it crucial to monitor and optimize token usage to manage expenses effectively.

Monitor your large language model and applications with BMC Helix

While organizations rush to implement LLMs into their products or create new products with GenAI, IT, data, and AI leaders need to ensure these models are used cost-effectively, responsibly, and safely. Check out this article to learn more about LLM observability.

By monitoring your LLM model quality and efficacy and LLM applications for run-time behavior with the BMC Helix LLM observability function, you can monitor various metrics to optimize performance and manage costs. It is generally available for all BMC Helix Observability and AIOps suite customers. See our documentation for more information to start saving on LLM observability.