Use of OPIK for Observability and Quality Evaluation of ACHIEVIT

1. Introduction

ACHIEVIT is an adaptive LLM-based academic planning system designed to generate, refine, and monitor long-horizon study plans for assignments, exams, and theses/dissertations. The system is aware of the goal contexts of users to be achieved, and key constraints (Time availability, skill capability, and deadlines) which may affect the goal from being achieved. As an adaptive agentic system designed to help user be productive and achieve their goals, ACHIEVIT's outputs are plans, not factual answers. Because the system is non-RAG, agentic, and planning-focused, traditional evaluation methods (accuracy, BLEU, heuristics) are insufficient in evaluating and ensuring the quality of ACHIEVIT. This design idea introduces three hard problems:

1. No single "correct" answer. The plans are produced based on the goal context and constraint awareness of the system.
2. Quality depends on relevance, grounding, and structure
3. Small prompt changes can cause silent regressions

Based on this understanding, and in order to make Achievit a stable, quality and trustworthy system, design and development are guided by this question:

> Which prompt–model configuration produces the most relevant, grounded, and reliable plans for users to optimally achieve their goal?

Thus, in this project, Comet OPIK was used as the core observability and evaluation backbone to develop Achievit as a quality system that can be trusted by users (researchers and students) to provide them with an adaptive scaffold/roadmap plan to achieve their goals, such as completing a research dissertation within a definite time, passing an exam with a specific grade targe.

2. Methodology

Achievit is designed to use a large language model (LLM) from Gemini as its core reasoning engine. OPIK is employed and utilised for end-to-end development, testing and evaluation so as to transform silent large language model (LLM) failures into measurable signals that directly improve system reliability and output quality. In the context of this project, I have relied on the OPIK cloud platform to achieve the following.

1. Development-Stage Observability: In the development stage, trace-level analysis and agentic system debugging were carried out. In this context, every LLM calls across the three goal resolutions (Exams, Assignment and Thesis/Dissertation) that Achievit support are tracked and logged as trace using OPIK. This ensured full visibility into any issues in the system behaviour under different constraint profiles with respect to:
    - Input acceptance
    - Correct output generation in JSON format
    - Failure cases at the LLM boundary
    - Token usage and cost per output generated
    - System Latency

2. Testing and Evaluation using LLM-as-a-Judge: The quality of Achievit is also ascertained through dataset-based experimentation and LLM-as-a-judge strategy. Two main evaluations I carried out are:
   - Prompt optimisation: Since Achievit is intended as a system to help students and researchers to achieve their stated goal resolution for the year with the LLM agent providing them with roadmap guidance on what to do, and adapting the plan dynamically as they make progress on subtask activities, I considered having a strong and effective working prompt to be essential in ensuring that system output is within the input context of the user. Thus, I crafted two prompt versions (Prompt v1.0 and Prompt v.20), tracked with OPIK decorator and have them compared:

   - Model Optimisation: To ensure that the Achievit is powered by the most reliable model that suits it and does not overgeneralise or take users far from achieving their goals, I compared two LLM models with best performing model selected based on cost, speed, efficiency and output quality. The characteristics of the two Gemini models I selected and evaluated are as follows:

Table 1: Evaluated models

	Token Limits		
	Input Token	Output Token	Uniqueness
Gemini-3-flash-preview	1,048,576	65,536	Offers high-speed, cost-effective performance
Gemini-3-pro-preview	1,048,576	65,536	Provide deeper = reasoning for complex, mission-critical performance

3. Evaluation Metrics: To ascertain the performance of the prompt versions and LLM models, I resort to the use of the LLM-as-a-judge strategy. This approach is based on the idea that the output produced by an LLM-powered agent A (e.g. Gemini-3-flash or Gemini-3-pro) that Achievit uses is considered valid and of acceptable quality if and only if an independent LLM-powered agent B (e.g. OpenAI GPT-4), acting as an evaluator, judges the output to satisfy predefined quality criteria, without any human intervention in the evaluation process. The characteristics of the LLM-as-a-judge that I followed are summarised as follows.

Table 2: Evaluation Metrics

Strategy	LLM-as-a-Judge
Judge Model	OpenAI GPT-4o
Models and Prompt Evaluated	Gemini-3-flash-preview; Gemini-3-pro-preview
Dataset	Custom dataset (N=15);
Evaluation Metrics	Answer Relevance, Hallucination and Moderation

3. Results and Discussion

3.1. Observability Result

The use of OPIK trace-level observability enabled effective debugging of ACHIEVIT, ensuring reliable system behaviour prior to production deployment. The system consistently accepted user goal inputs and produced the expected structured JSON outputs from the LLM across all supported goal types, demonstrating robustness and correctness in model integration.

In total, 432 traces were logged in the OPIK Cloud across system debugging, prompt versioning and optimisation, and model optimisation and evaluation using LLM-as-a-judge**.** Among these traces, 72 errors were recorded, primarily stemming from API rate-limit exhaustion and server overload conditions**.**

OPIK proved advantageous in providing deep visibility into system behaviours and failure modes that could otherwise impact production reliability. These issues were subsequently mitigated by engineering dedicated function blocks to catch and handle such errors within the system logic. Overall, the total cost of all 432 logged traces was $2.569, with an average P50 latency of 35.9 seconds, demonstrating cost-efficient and observable development at scale.

3.2. Prompt Optimisation with OPIK: Same Model, Different Prompts

In the first experiment, an evaluation of prompt quality was carried out using a controlled setup where only the prompt was varied, while the underlying model (Gemini-3-Flash), dataset, and evaluation framework remained constant. All evaluations were performed using OPIK with GPT-4o as an LLM-as-a-judge, measuring average answer relevance, hallucination risk, moderation, latency, and cost. Table 3 summarises the results of the experiment.

Table 3: Prompt v1 vs Prompt v2 (Gemini-3-Flash) (Data source can be assessed on the Opik Cloud workspace)

	Average					
Prompt	Answer Relevance ↑	Hallucination ↓	Moderation	Latency ↓	Total Token	Total Cost ($)
Prompt v1	0.893	0.189	0.00	75.7s	9966.27	0.008
Prompt v2	0.894	0.191	0.00	47.4s	9417.733	0.007
Change (+/-)	+0.11%	+1.06%	0.00%	−37.38%	−5.50%	−12.50%
*LLM-as-a-judge model: OpenAI GPT-4o *model Judged: Gemini-3-flash *Change deviations are calculated and reported in percentage Baseline: Prompt v1 Change = (Prompt v2 − Prompt v1) / Prompt v1 × 100						

The OPIK evaluation reveals that Prompt v1 and Prompt v2 are statistically equivalent in quality, while Prompt v2 delivers substantial efficiency gains:

- Answer relevance remains stable (+0.11%)
- Hallucination risk increases only marginally (+1.06%)
- Latency is reduced dramatically (−37.38%)
- Cost decreases meaningfully (−12.5%)
- No safety regression (moderation unchanged)

Based on the use of this OPIK's LLM-as-a-judge metrics, I was able to confidently determine that Prompt v2 achieves significant performance and cost improvements without degrading output quality. However, qualitative inspection combined with hallucination sensitivity suggests that Prompt v2 slightly favours compression and speed, which may encourage gap-filling behaviour rather than deliberate reasoning in complex academic planning scenarios. For an agentic system like ACHIEVIT, where structured reasoning is critical, this trade-off becomes significant. As a result, Prompt v1 was selected as the quality-preserving baseline for downstream model comparison. The figure below compares the performance of the two prompts.

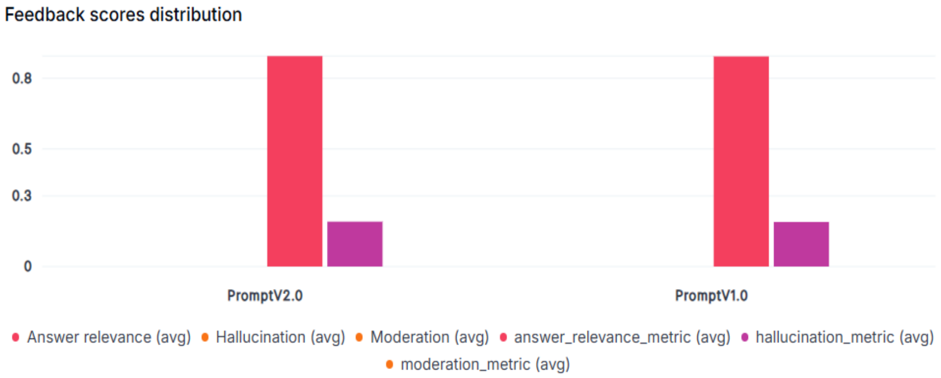

Figure 1: Prompt Version Performance

3.3. Model Comparison with OPIK (Same Prompt, Different Models)

With Prompt v1 fixed, I conducted a model-vs-model evaluation to assess whether a larger, more expensive model would yield better planning quality. Thus, using the same dataset and evaluation setup, I compared Gemini-3-Flash against Gemini-3-Pro. Table 4 below summarises the results of the evaluations

Table 4: Comparison of Model Performance (Data source can be assessed on Opik Cloud workspace)

	Average					
Prompt	Answer Relevance ↑	Hallucination ↓	Moderation	Latency ↓	Total Token	Total Cost ($)
Gemini-3-flash	0.912	0.119	0.00	50.2s	9820.133	0.008
Gemini-3-pro	0.753	0.273	0.00	60.3s	9914.867	0.033
Change (+/-)	−17.43%	+129.41%	0.00	+20.12%	+0.96%	+312.50%
*LLM-as-a-judge model: OpenAI GPT-4o *model Judged: Gemini-3-flash						

	Average					
Prompt	Answer Relevance ↑	Hallucination ↓	Moderation	Latency ↓	Total Token	Total Cost ($)
*Change deviations are calculated and reported in percentage Baseline: Prompt v1 Change = (Gemini-pro – gemini-flash) / Gemini flash × 100						

From the result presented in Table 3, the evaluation metrics show that the larger Gemini-3-Pro model performed substantially worse across every critical dimension except moderation. Particularly:

- Answer relevance dropped significantly (−17.43%)
- Hallucination risk increased sharply (+129.41%)
- Latency increased (+20.12%)
- Cost increased more than 3× (+312.5%)
- No safety advantage observed

While Gemini-3-Pro is more capable in general language tasks, the OPIK evaluation presented above revealed that for Achievit system design, it is less aligned for structured, constraint-driven, agentic academic planning, where concise reasoning and adherence to user intent are paramount. Thus, a decision is made to ensure that the Gemini-3-Flash is used for production. Figure 2 shows the compared model performance.

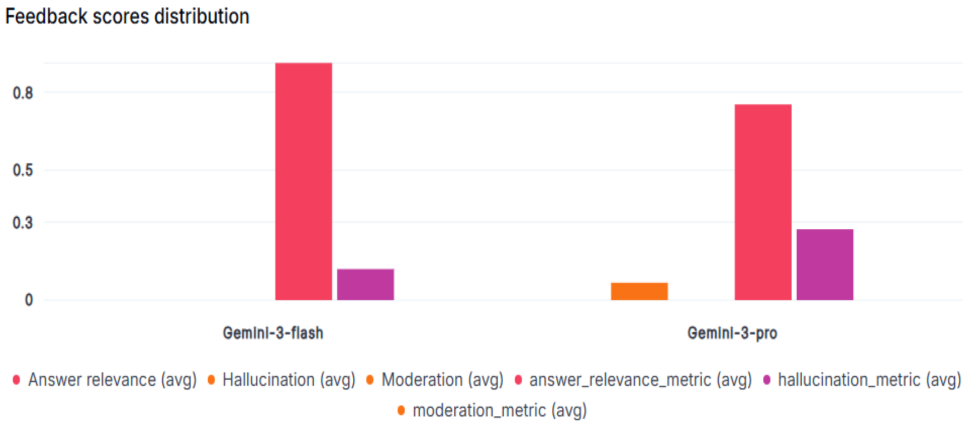

Figure 2: Model's Performance

4. Conclusion

This project demonstrates the use of Comet OPIK as a decision-making system, not merely a monitoring dashboard. In ACHIEVIT, OPIK was used to guide prompt and model selection through controlled, empirical evaluation rather than intuition or model size assumptions. Using OPIK's LLM-as-a-judge evaluation framework, systematic optimisation of ACHIEVIT across quality, latency, and cost dimensions was achieved. Prompt-level experiments showed that while prompt v2 achieved a 37% reduction in latency and 12.5% reduction in cost, plan quality metrics (answer relevance and hallucination risk) remained statistically equivalent to prompt v1. This enabled an informed efficiency–quality trade-off decision that prompts 1 is suitable for real-world deployment.

More importantly, OPIK surfaced a non-intuitive regression during model comparison in this project. Despite being a larger and more expensive model, Gemini-3-Pro increased hallucination risk in Achievit by over 129%, reduced answer relevance by 17%, and incurred higher latency and cost compared to Gemini-3-Flash. Without OPIK's structured evaluation, this regression would likely have gone unnoticed, as both models appeared superficially competent in manual testing.

OPIK's impact in this work can be summarised as follows:

1. Agent system observation and debugging
2. Controlled Experimentation: Prompt-only comparisons were conducted before model-only comparisons, using the same dataset, evaluation metrics, and LLM judge to ensure fair and interpretable results.
3. Correct Use of LLM-as-a-Judge: Rather than relying on surface heuristics, OPIK evaluated semantic quality—relevance, grounding, and hallucination risk—at scale for non-deterministic, agentic outputs.
4. Discovery of Non-Obvious Insights: OPIK revealed that a smaller model outperformed a larger one in structured planning tasks, challenging the assumption that model size correlates with planning quality.
5. Production-Grade Decision Making: Prompt and model selection were justified using empirical quality, risk, latency, and cost metrics, enabling confident deployment decisions.
6. Clear Business and System Impact: The final system achieved lower hallucination risk, reduced operational cost, faster response times, and higher planning reliability for users.

In conclusion, OPIK enabled ACHIEVIT to evolve from a promising agentic system into a measurable, trustworthy, and production-ready planning assistant, demonstrating how observability and evaluation are essential for building reliable LLM-driven applications.