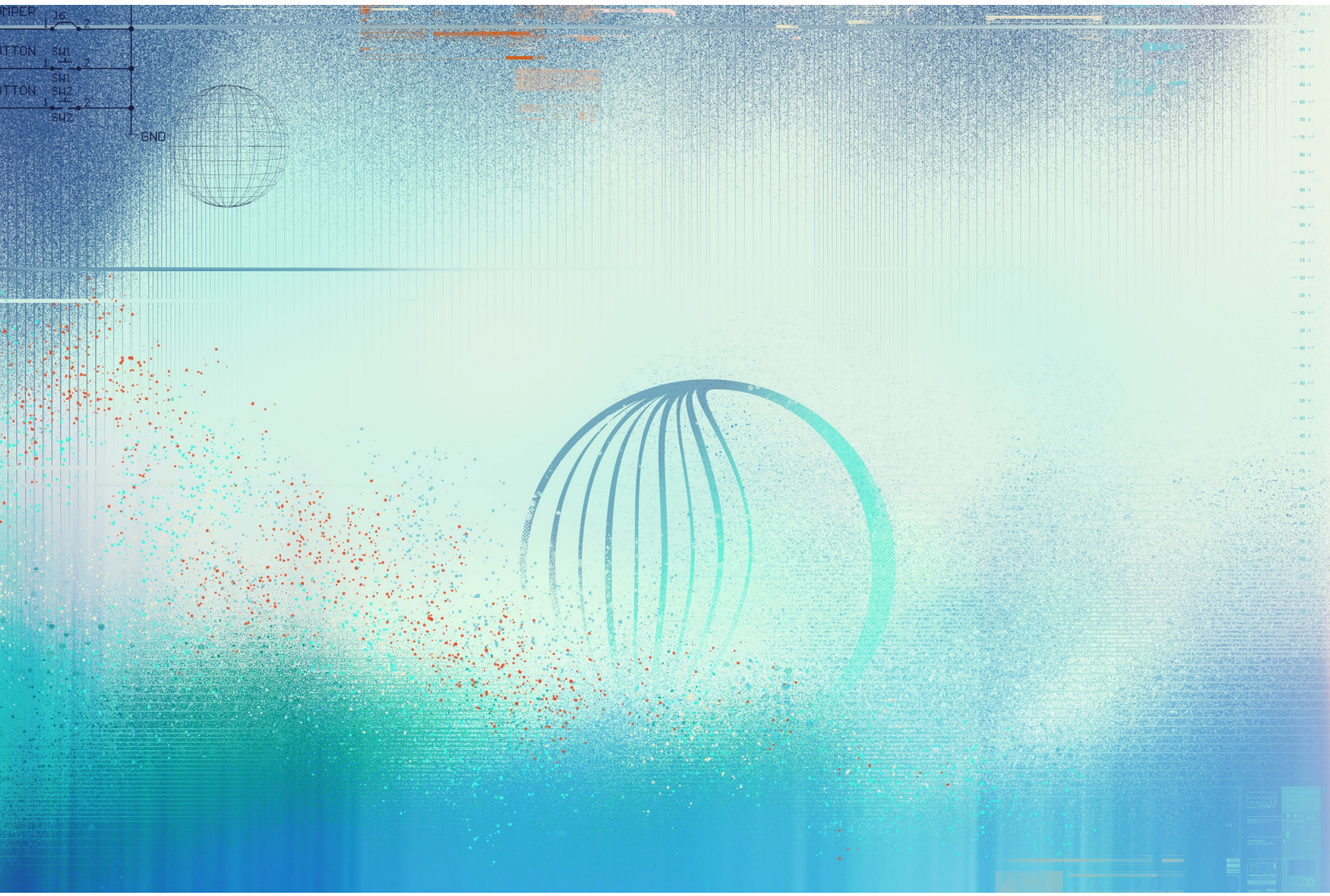
signal65



AGENTIC AI CAPABILITIES TESTING Q3 2025

Measured Leadership with Agentic AI on Open Models

Evaluating Agentic AI Capabilities with the KAMI v0.1 Benchmark

AUTHOR

Mitch Lewis
Performance Analyst | Signal65

NOVEMBER 2025

Executive Summary

Over the past few years, AI has evolved from a speculative technology to a key priority for enterprise organizations. Rapid model development has led to larger, more complex, and more reliable LLM models. For enterprise use, however, it is agentic applications that offer real value – enabling AI to solve challenges and complete valuable business tasks with as little human intervention as possible.

While agentic AI is the focus of these enterprise efforts, evaluating the usefulness of LLMs to complete agentic tasks has proven to be a challenge. Existing AI benchmarks primarily measure a model's reasoning ability, rather than its ability to successfully complete enterprise-related tasks. In addition, static AI benchmarks often become incorporated into a model's training data, reducing the benchmark to a test of memorization.

To overcome these challenges, Signal65 and Kamiwaza have collaborated to establish a new AI benchmark which measures model performance for enterprise-focused agentic tasks. This paper presents the first iteration of the Kamiwaza Agentic Merit Index (KAMI).

Key Highlights

- Qwen3-235B-A22B Instruct-2507-FP8 is the top agentic AI performer at **88.8% mean accuracy score**
- FP8 quantization impacts accuracy by **~3% for agentic workloads**
- Thinking models **up to 25% more accurate** for agentic AI workloads

Key findings include the following:

- **Top Performer:** Qwen3-235B-A22B-Instruct, both the FP8 quantized and full weight version, achieved the highest scores among models tested, indicating it is a top open source AI model to be considered for agentic AI deployments.
- **Model Size:** In general, accuracy was seen to improve with model size, with the highest scores attributed to very large models with over 100B parameters. Small models (<10B parameters) showed a clear deficiency across most agentic tasks. Some models in the 30 to 100B parameter range, however, such as Llama-3.1-70B-Instruct and Qwen3-30B-A3B (thinking mode) outperformed much larger models, demonstrating compelling options for organizations with limited infrastructure.
- **Quantization:** FP8 quantization does not appear to have adverse effects on agentic capabilities. Across FP8 quantized and full weight model pairs tested, the FP8 variations consistently achieved similar or even slightly greater accuracy.
- **Thinking:** Models with thinking capabilities were generally found to be more accurate in achieving agentic tasks than similar non-thinking models. Non-thinking models, however, became highly competitive when provided basic hints and context clues, offering a possible alternative to the high token usage and cost associated with thinking models.
- **Agentic Benchmarking Disconnect:** Several models which achieved high scores across other common AI benchmarks scored disproportionately low in the KAMI v0.1 benchmark, indicating a disconnect between traditional AI benchmarking and real world application. Additionally, some older generation models across both Llama and Qwen model families outperformed their newer generation counterparts that are typically considered to be more advanced according to traditional benchmark results.

An overview of the top 10 performing models tested in the KAMI v0.1 Benchmark can be seen below, with deeper details on the process and results following:

Rank	Model	Mean Accuracy Score
1	Qwen3-235B-A22B-Instruct-2507-FP8	88.8 %
2	Qwen3-235B-A22B-Instruct-2507	88.4 %
3	Claude-3.5-Haiku-20241022	75.9 %
4	Llama-4-Maverick-17B-128E-Instruct	74.6 %
5	Llama-3.3-70B-Instruct-FP8-KV	74.5 %
6	Llama-3.1-70B-Instruct	73.4 %
7	Llama-4-Maverick-17B-128E-Instruct-FP8	73.1 %
8	Qwen3-30B-A3B (thinking mode)	72.7 %
9	Llama-3.3-70B-Instruct	71.6 %
10	Qwen2.5-72B-Instruct	71.1 %

Figure 1: *KAMI v0.1 Benchmark Top 10 Results*

Challenges with Agentic Benchmarking

Agentic AI has quickly become the focus of enterprise AI adoption. By leveraging agents, AI can perform useful enterprise tasks, from simple routine tasks, to complex multi-step operations. As enterprises begin building such agentic systems, however, evaluation of LLMs becomes a crucial component. The chosen LLM will have a direct impact on the accuracy, effectiveness, and efficiency of the agent.

While there are several benchmarks currently available to evaluate AI models, the existing approaches utilize flawed methods for evaluating models for true agentic use cases.

First, the majority of AI benchmarks are static, leading to data contamination and memorization issues. With static benchmarks, the benchmark itself can easily be introduced into a model's training data, whether intentionally or unintentionally. This invalidates the challenge of the benchmark, instead resulting in a test of memorization. When evaluating static benchmarks, it can't be known if a model is performing well due to its own merit or due to previous exposure to the benchmark. While the creation of new benchmarks temporarily solves this problem, it is not a scalable approach to accommodate ongoing model development.

The second key challenge for enterprises is that **existing benchmarks are not accurately representative of agentic use cases**. Most benchmarks only evaluate single-shot question and answer responses. Agentic workflows, on the other hand, often involve multi-step inference and tool calling to complete specific enterprise related tasks, such as querying a database, evaluating the data, and formatting a result.

These limitations can result in misleading or uninformative results, challenging enterprises to select the appropriate models to support their applications. Poor model selection can lead to inaccurate results and incomplete tasks. In the enterprise, these critical mistakes can cause costly disruptions and potentially negate the advantages of leveraging agentic AI. For complex, multi-agent applications, the importance of model selection is further heightened, as inaccuracy in a single agent can compound throughout the application, impacting all other agents and the final quality and reliability.

Introducing KAMI

To overcome the limitations of existing AI benchmarks and establish a realistic method of measuring model performance in agentic scenarios, Kamiwaza and Signal65 have developed the Kamiwaza Agentic Merit Index (KAMI).

KAMI differentiates itself from other AI benchmarks by utilizing a dynamic test suite that measures the completion of real agentic tasks. Unlike traditional, static AI benchmarks, KAMI randomizes each question and generates a unique ground truth answer key at runtime, preventing models from simply memorizing the tests during training. In addition to preventing memorization, KAMI also goes beyond measuring simple reasoning capabilities, with tests designed to evaluate a model's proficiency in completing actual enterprise tasks. KAMI requires models to reason through enterprise-oriented tasks, such as answering business questions by extracting information from CSV files or databases. These tasks accurately represent common, real-world agentic workflows that involve loops of LLM inference and tool calling.

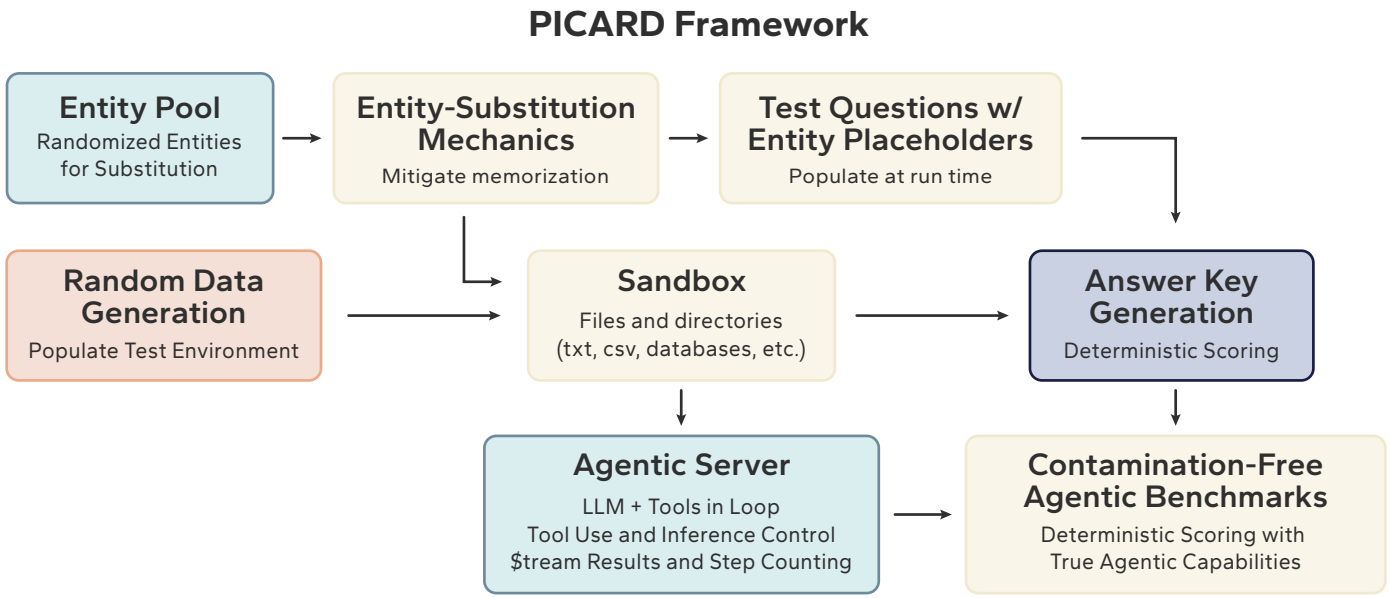


Figure 2: *PICARD Framework Architecture*

KAMI, based on the PICARD framework, creates a dynamic, randomized test suite targeted at specific enterprise workloads. Key components of KAMI include:

- **Sandbox Environment: T**o enable real, agentic tasks, KAMI creates a sandbox environment for LLMs to do work as needed, including writing files, or connecting to databases.
- **Multi-layer Randomization:** KAMI creates unique, dynamic questions by deploying two layers of randomization.
  - Randomized Entity Substitution randomizes the relevant entities in each benchmark question from a pool of possible entities. Examples of entities include file names, directories, and database tables.
  - Randomized Data Generation randomizes the data, such as directories, files, and databases, that are available within the test environment.
- **Answer Key Generation:** All responses are graded against a ground truth answer key. In order to generate ground truth answers while utilizing randomized question generation, the unique answer key for each randomized question set is generated at run time.
- **Agentic Server:** To evaluate real-world agentic tasks, an agentic server is deployed to enable LLMs with tool calls in a loop. By utilizing an agentic server, LLMs can iteratively select tools, execute tools, and evaluate results in a loop to achieve complex tasks.

An example of a question in the KAMI benchmark can be seen in Figure 3 below:

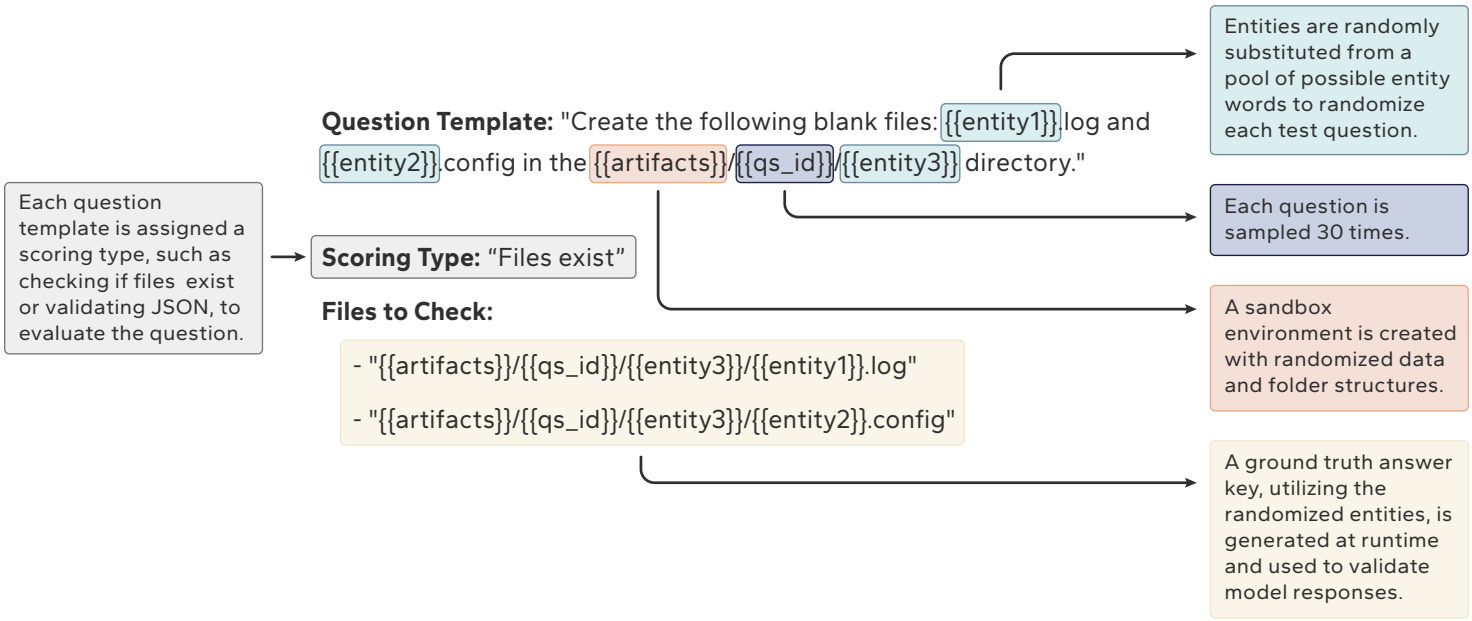


Figure 3: *Question Template Overview*

Figure 4 demonstrates this same question populated with randomized data, along with an example of the correct answer and a possible incorrect answer.

Example:

"Create the following blank files: crimson.log and whisper.config in the test_artifacts/q201_s20/ancient directory."

Correct Answer:

- "test_artifacts/q201_s20/ancient/crimson.log"
- "test_artifacts/q201_s20/ancient/whisper.config"

The correct answer created the properly named files in the correct directories.

Incorrect:

- "test_artifacts/q201_s20/ancient/crimson/ancient.log"
- "test_artifacts/q201_s20/ancient/whisper"

Any answer that doesn't match the expected answer key is incorrect. In this example, the first file is incorrectly named and placed in the wrong folder. The second file is in the correct directory and matching the entity name, but missing the proper file extension.

Figure 4: *Randomly Generated Question Example*

This example demonstrates a fairly simple question within the Kami v0.1 benchmark, which tests if an agent is capable of creating basic files and placing them in the correct directory. The full benchmark contains many more complex questions which involve gathering randomized information from databases, CSV files, and text files. An example of a database question can be seen in Figure 5 and Figure 6. Across the various test cases, models were found to generate incorrect responses for wide range of reasons. Common challenges included incorrect or random tool usage, semantic confusion, and handling numerical values.

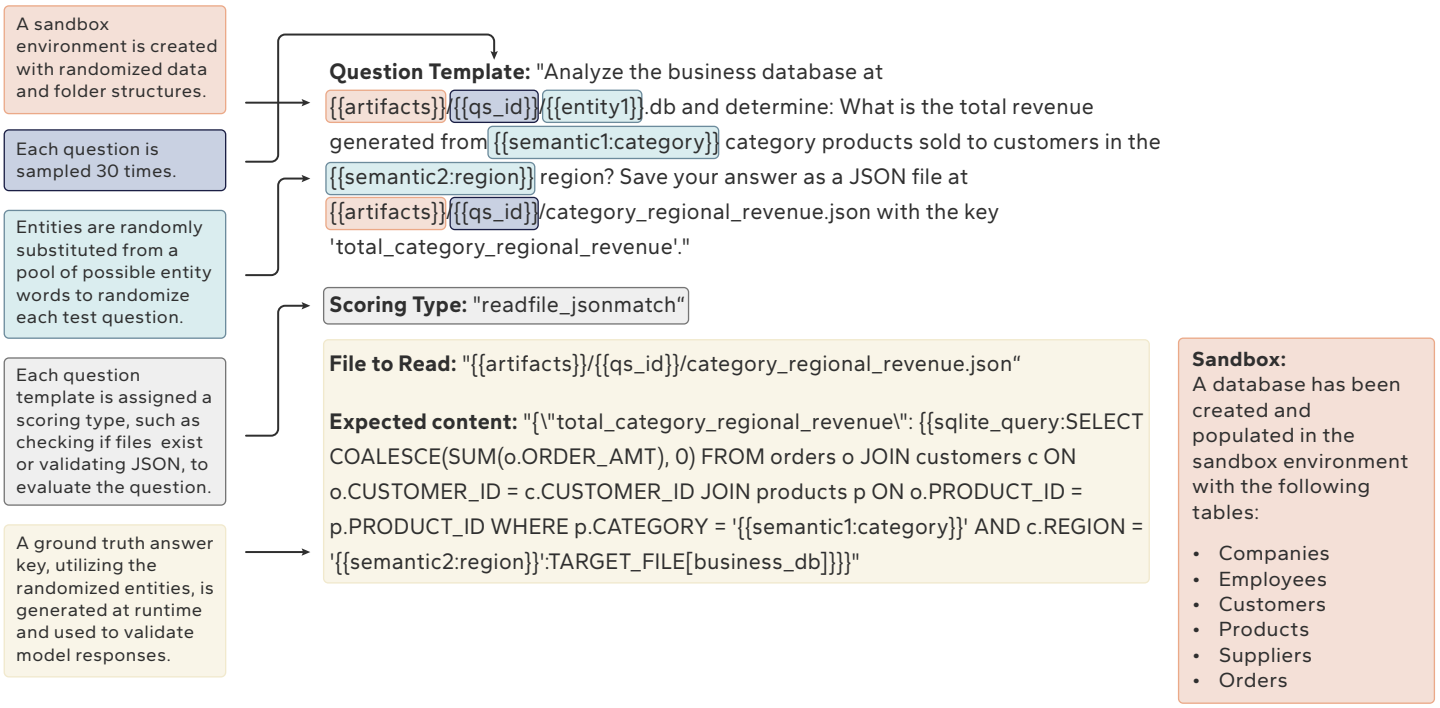


Figure 5: *Database Question Template Overview*

Example:

"Analyze the business database at test_artifacts/q503_s11/harbor.db and determine: What is the total revenue generated from technology category products sold to customers in the west region? Save your answer as a JSON file at test_artifacts/q503_s11/category_regional_revenue.json with the key 'total_category_regional_revenue'."

Correct Answer:

File:
test_artifacts/q503_s11/category_regional_revenue.json

Contents:
{"total_category_regional_revenue": 10000}

The correct answer correctly retrieved the answer to the query, formatted it as JSON, and created the output file in the correct directory.

Incorrect:

File:
test_artifacts/q503_s11/category_regional_revenue.json

Contents:
{"total_category_regional_revenue": 500}

Any answer that doesn't match the expected answer key is incorrect. In this example, the agent correctly created a JSON file with correct formatting, but incorrectly queried the database to calculate the value for 'total_category_regional_value'.

Figure 6: *Database Question Example*

Additional technical details and design principles of the KAMI benchmark are further outlined by Kamiwaza and can be found here.

The v0.1 benchmark presented in this paper represents the first iteration of the KAMI benchmark. Signal65 and Kamiwaza plan for continued development of the KAMI benchmark, with future versions expanding to include more models and expanded testing abilities to further evaluate the agentic abilities and enterprise readiness of LLMs.

The KAMI benchmark introduces a unique new capability within Signal65's AI benchmarking and analysis portfolio. Designed to go beyond traditional test suites, KAMI provides a structured yet flexible framework for evaluating AI models, systems, and applications under realistic enterprise conditions. Signal65 will use KAMI as a foundation for ongoing model validation and end-to-end testing within its AI Lab. Through this approach, Signal65 will generate meaningful AI performance insights and advance industry understanding of how enterprise AI should be measured and compared.

Testing Overview

The KAMI v0.1 Benchmark contains 19 distinct question templates, grouped into 7 specific categories. All questions were sampled 30 times for each run of the KAMI test suite to accommodate the variance of the randomized questions. In addition, for each model tested, the entire test suite was run multiple times and models were scored using their mean accuracy over all runs. An overview of the test questions can be seen in Figure 7 below.

Category	Performance
Basic Reasoning	Respond only with a specific word.
	Respond with multiple specified words in a specified order.
File System Operations	Create specific files in a specified directory.
	Create specific directory structures and include various files.
Text Search and Extraction	Find two specific lines from a file.
	Find several specific lines from an extended file.
	Retrieve two specific words from a text file.
	Retrieve several specific words from an extended text file.
CSV Processing	Create JSON summary of a CSV file.
	Analyze business data across multiple CSV files. Answer 6 specific questions.
	Analyze business data across multiple CSV Files. Single question.
Database Processing	Query business database to fine number of orders over a specified value within a specified region.
	Analyze business database and create a comprehensive report. 6 specific questions.
	Analyze business database to find total revenue from a specified product in a specified region.
Database Processing (Guided)	Repeat simple database task with a hint given.
	Repeat complex database task with a hint given.
Response Format Instruction Following	Output answer to txt file.
	Output answer in JSON format.
	Output number only.

Figure 7: *KAMI Question Overview*

The KAMI v0.1 benchmark was tested on 31 total models, resulting in over 170,000 total test conversations and over 5.5 billion tokens processed. Models were chosen to represent popular LLMs often considered in enterprise AI deployments. Models of various sizes and versions were additionally included to gain insight into their agentic capabilities. This iteration of the KAMI benchmark was primarily focused on open source models, due to their ease of access and the wide range of models available. All open source models were run on hardware in the Signal65 AI Lab with either AMD MI300X or Intel Gaudi 3 devices, while proprietary models were run using their native API endpoints.

Signal65 Comment *– While tests were run across different hardware platforms in the Signal65 AI Lab, it should be noted that this testing was solely focused on model behavior and not as a measurement of system performance. Several tests were repeated across both AMD MI300X and Intel Gaudi 3 devices to ensure consistency in the testing process. No statistically significant differences were found in the results.*

Results

The full results of the KAMI v0.1 benchmark can be seen in Figure 8 below.

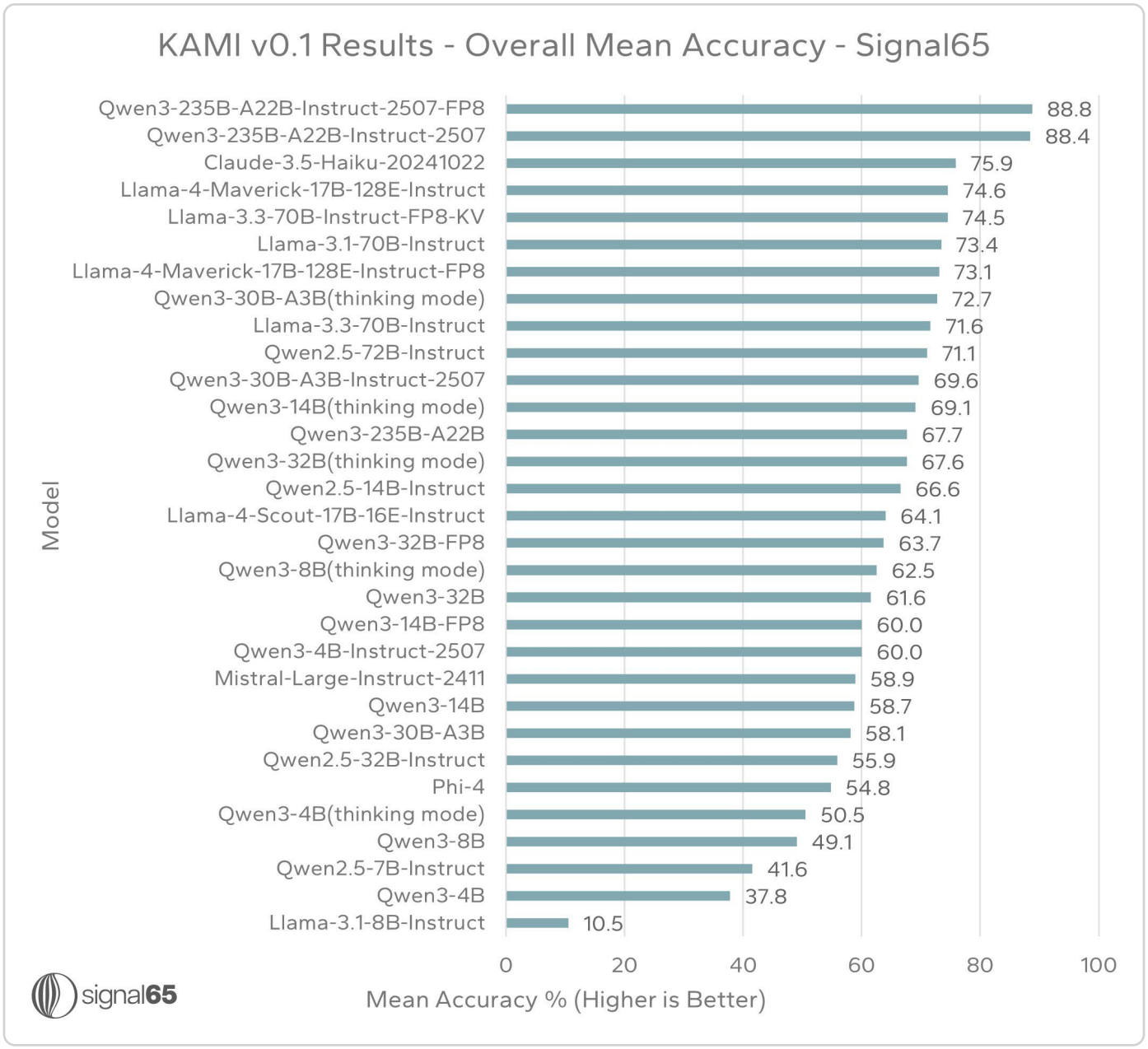


Figure 8: *KAMI v0.1 Results Overview*

Testing found Qwen3-235B-A22B-Instruct-2507-FP8 and Qwen3-235B-A22B-Instruct-2507 to achieve the highest overall scores with mean accuracies of 88.8% and 88.4% respectively. Notably, these two models were the only models to record an average accuracy over 80%.

The remainder of the top five performing models, Claude-3.5-Haiku-20241022, Llama-4-Maverick-17B-128E-Instruct, and Llama-3.3-70B-Instruct-FP8-KV, all obtained relatively competitive scores, ranging from 74.5% to 75.9% accuracy.

At the low end, Llama-3.1-8B scored the lowest overall accuracy at 10.5%. In addition, three other models also scored below 50% accuracy, including Qwen3-4B, Qwen2.5-7B-Instruct, and Qwen3-8B.

A greater understand of each model's agentic performance can be gained from evaluating the results for each test category.

Basic Reasoning

The basic reasoning tasks in the KAMI v0.1 benchmark exist primarily as baseline measurement, to ensure that the models can perform simple, non-tool calling tasks without issue. There are two distinct questions in this category, one that asks the model to respond only with a specific word, and a slightly more challenging question that asks for a response of multiple words in a specified order. Neither question should require tool calls to successfully complete.

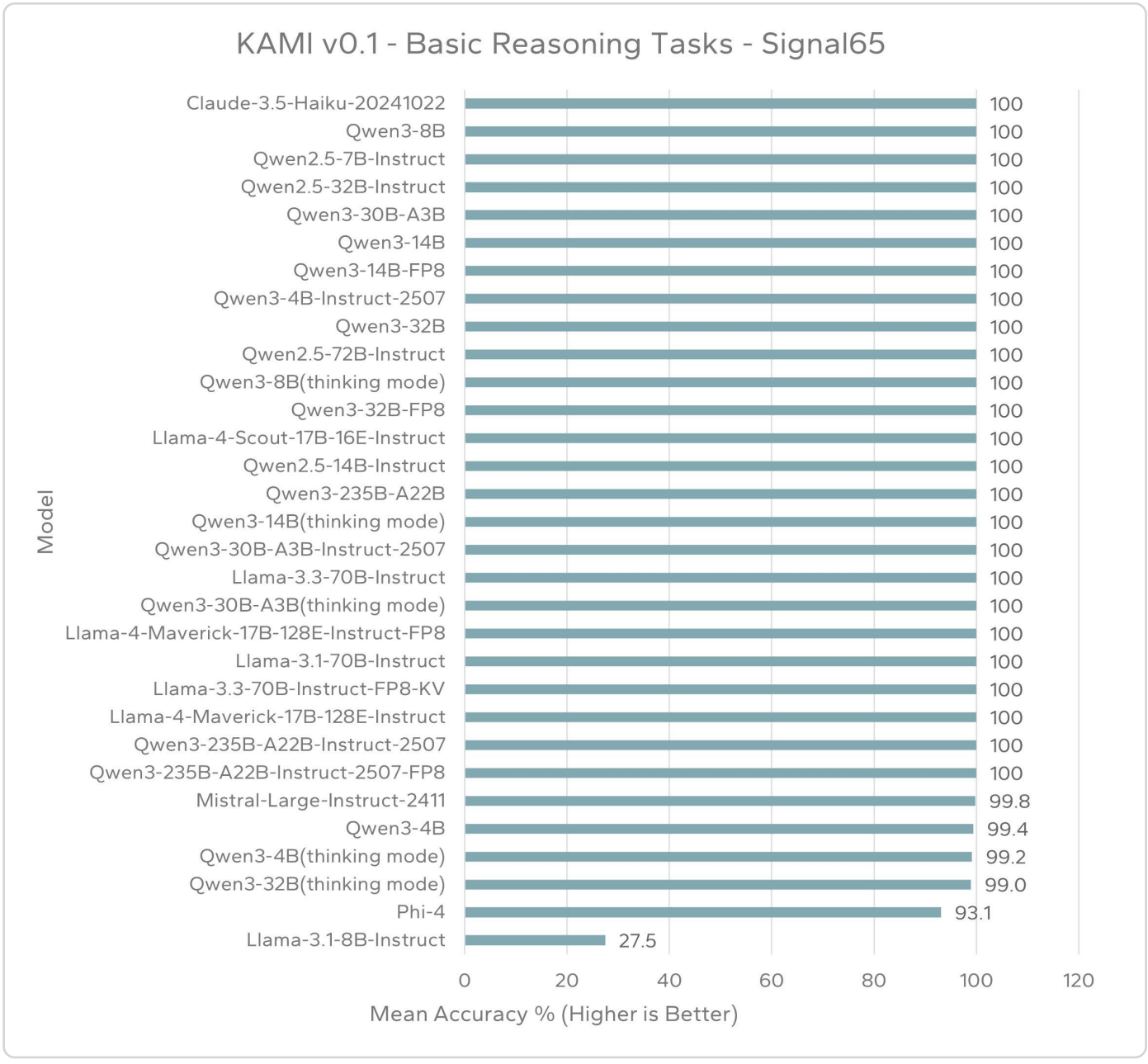


Figure 9: *KAMI v0.1 Basic Reasoning Tasks*

As can be seen in Figure 9, the majority of models achieved 100% or nearly 100% accuracy for these tasks. Llama-3.1-8B-Instruct, however, was found to be highly inaccurate for these basic reasoning tasks, only achieving 27.5% accuracy. The most common issue for models struggling with these questions was attempting to utilize tool calls unnecessarily. This type of error signals that such a model may not be well suited to tool access, and therefore not well suited to agentic use cases.

Filesystem Operations

The filesystem operations category measures a model's ability to complete basic filesystem operations, such as creating files and directories. In the KAMI v0.1 benchmark, this category includes two questions, the first instructing models to create files in a specific folder, and the second, a slightly more complex question instructing the model to create a full directory structure. These tasks reflect common agentic use cases, in which AI agents must navigate and modify filesystems.

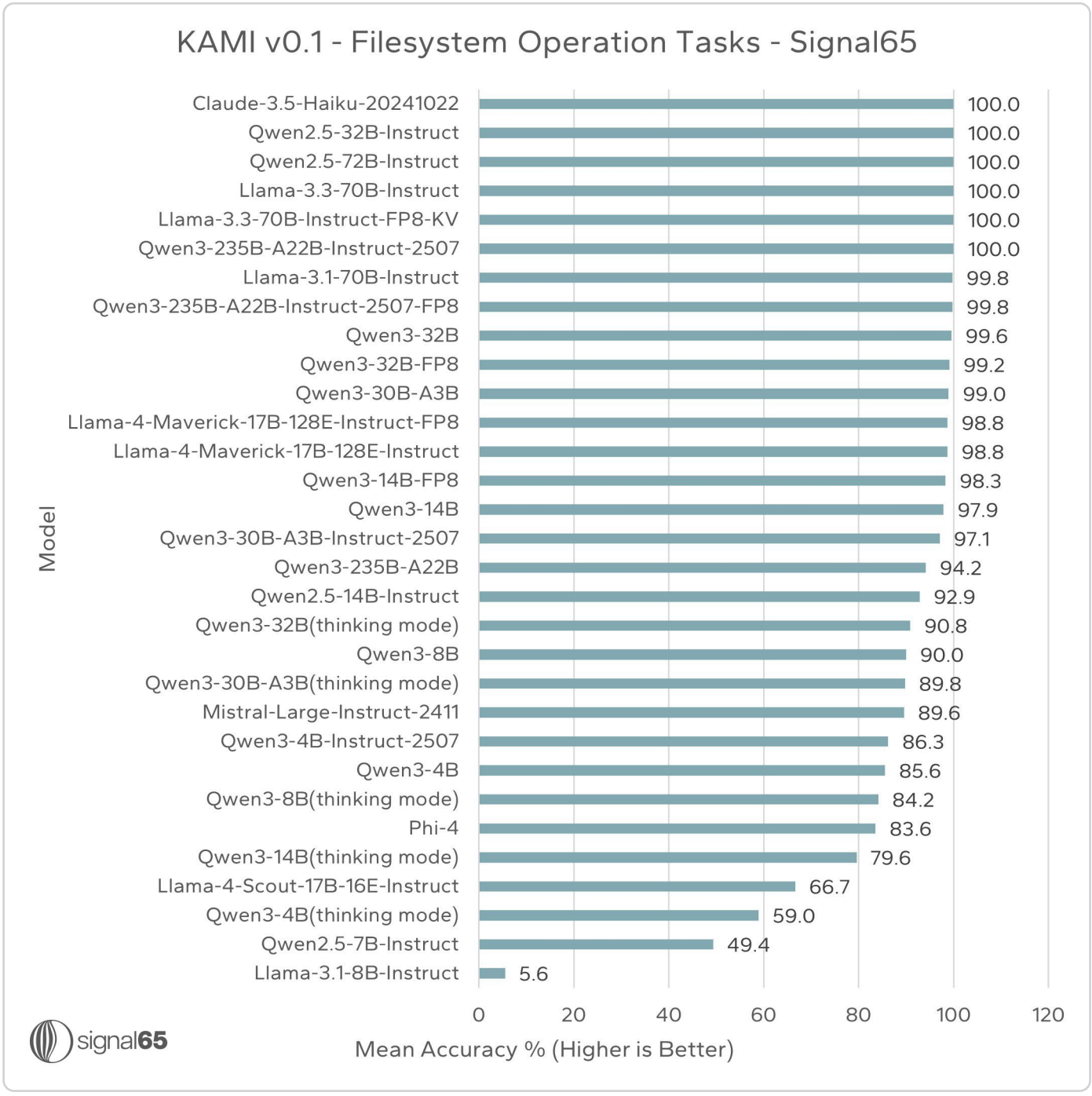


Figure 10: *KAMI v0.1 Filesystem Operation Tasks*

For the filesystem tests, six models achieved 100% accuracy: Qwen3-235B-A22B-Instruct-2507, Llama-3.3-70B-Instruct-FP8-KV, Llama-3.3-70B-Instruct, Qwen2.5-72B-Instruct, Qwen2.5-32B-Instruct, Claude-3.5-Haiku-20241022. Several other models achieved near-perfect accuracy, including the overall benchmark leader Qwen3-235B-A22B-Instruct-2507-FP8 at 99.8%.

As could reasonably be expected, the Llama-3.1-8B-Instruct model that did not perform well during the basic reasoning tasks was also highly inaccurate for both filesystem tasks, achieving an accuracy of only 5.6%. The only other model to score below 50% for these tasks, Qwen2.5-7B-Instruct, also struggled nearly equally with both tasks, with a 51.25% accuracy for the first question and 47.5% accuracy for the second. For many other models that performed poorly, however, the second, more complicated filesystem task was found to be much more challenging. Notably, this trend was consistently seen amongst Qwen3 thinking models. Examples of this can be seen in Figure 11 below:

Model	Filesystem Question #1	Filesystem Question #2	Overall Filesystem Accuracy
Qwen3-4B (thinking mode)	87.1%	30.8%	59%
Qwen3-14B (thinking mode)	99.2%	60%	79.6%
Qwen3-8B (thinking mode)	97.9%	70.4%	84.2%
Qwen3-30B-A3B (thinking mode)	97.9%	81.7%	89.8%

Figure 11: *KAMI v0.1 File System Operation Tasks (Qwen3 Thinking)*

Text Search and Extraction

The third test category involved tasks in which AI agents must find information from within text files. This was tested in two distinct ways – first by asking models to retrieve lines of text given specific line numbers, and second to retrieve specific words given specific word counts. Two versions of each task were asked: a short version, which asked to retrieve two lines or words, and a more complex version, which asked to retrieve several lines or words from a longer text file.

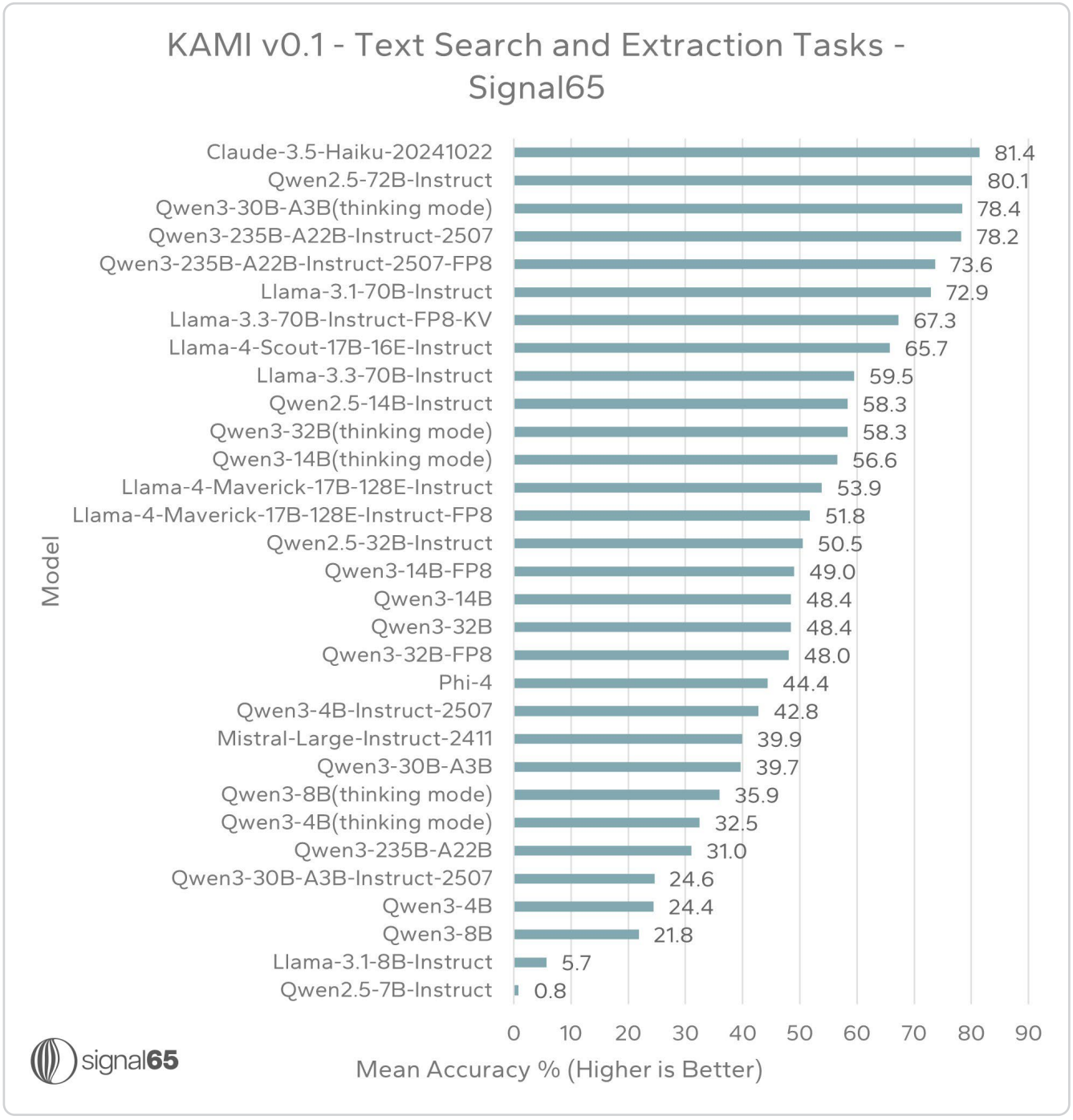


Figure 12: *KAMI v0.1 Text Search and Extraction Tasks*

These text search and extraction tasks proved to be much more difficult than the previous filesystem tasks, with no models scoring 100% accuracy. Most models, with the exception of the few lowest scoring models, completed the two line retrieval tasks with a high accuracy, however, even the most successful models were far less accurate for the two word retrieval tasks. Since line numbers are recorded in text files, and specific word counts are not, retrieving specific words is considered a much more complicated task. Successful models were found to write python code to find specific word counts, while other models often resorted to a less successful approach of attempting to manually count each word – a task not well suited to LLMs.

A breakdown of the individual scores for the top five performing models can be seen below:

Model	Text Search Question #1	Text Search Question #2	Text Search Question #3	Text Search Question #4	Average Accuracy
Claude-3.5-Haiku-20241022	97.8%	70.0%	68.9%	88.9%	81.4%
Qwen2.5-72B-Instruct	99.0%	91.4%	60.5%	69.5%	80.1%
Qwen3-30B-A3B (thinking mode)	97.1%	82.9%	46.3%	87.5%	78.4%
Qwen3-235B-A22B-Instruct-2507	100.0%	100.0%	57.9%	55.0%	78.2%
Qwen3-235B-A22B-Instruct-2507-FP8	99.6%	100.0%	40.0%	55.0%	73.6%

Figure 13: *KAMI v0.1 Text search and Extraction Tasks Top 5*

The impact of the challenging word retrieval tasks is apparent even within the top five performing models. The two overall benchmark leaders Qwen3-235B-A22B-Instruct-2507-FP8 and Qwen3-235B-A22B-Instruct-2507 both scored 100%, or near 100% for both line retrieval tasks, but were far less accurate for the word retrieval tasks. Alternatively, Claude-3.5-Haiku-2024022, Qwen2.5-72B-Instruct, and Qwen3-30B-A3B (thinking mode) were slightly less accurate during the two line retrieval tasks, however, the greater word retrieval accuracy displayed by these models resulted in a higher overall ranking within this category.

In total, eight of the models tested scored 0% for both word retrieval tasks, including several that scored 90% or above for the two line retrieval tasks. This demonstrates an ability for some models to achieve relatively simple tasks, in which information – such as line numbers – is easily accessible, but break down when more complex logic is required.

CSV Processing Tasks

The CSV processing tasks included in the KAMI v0.1 benchmark instructed models to analyze business data across one or more CSV files to answer specific questions. The KAMI v0.1 benchmark included three distinct CSV processing tasks, with varying levels of complexity. These tasks reflect a highly desirable enterprise AI use case of analyzing CSV data to answer valuable business questions.

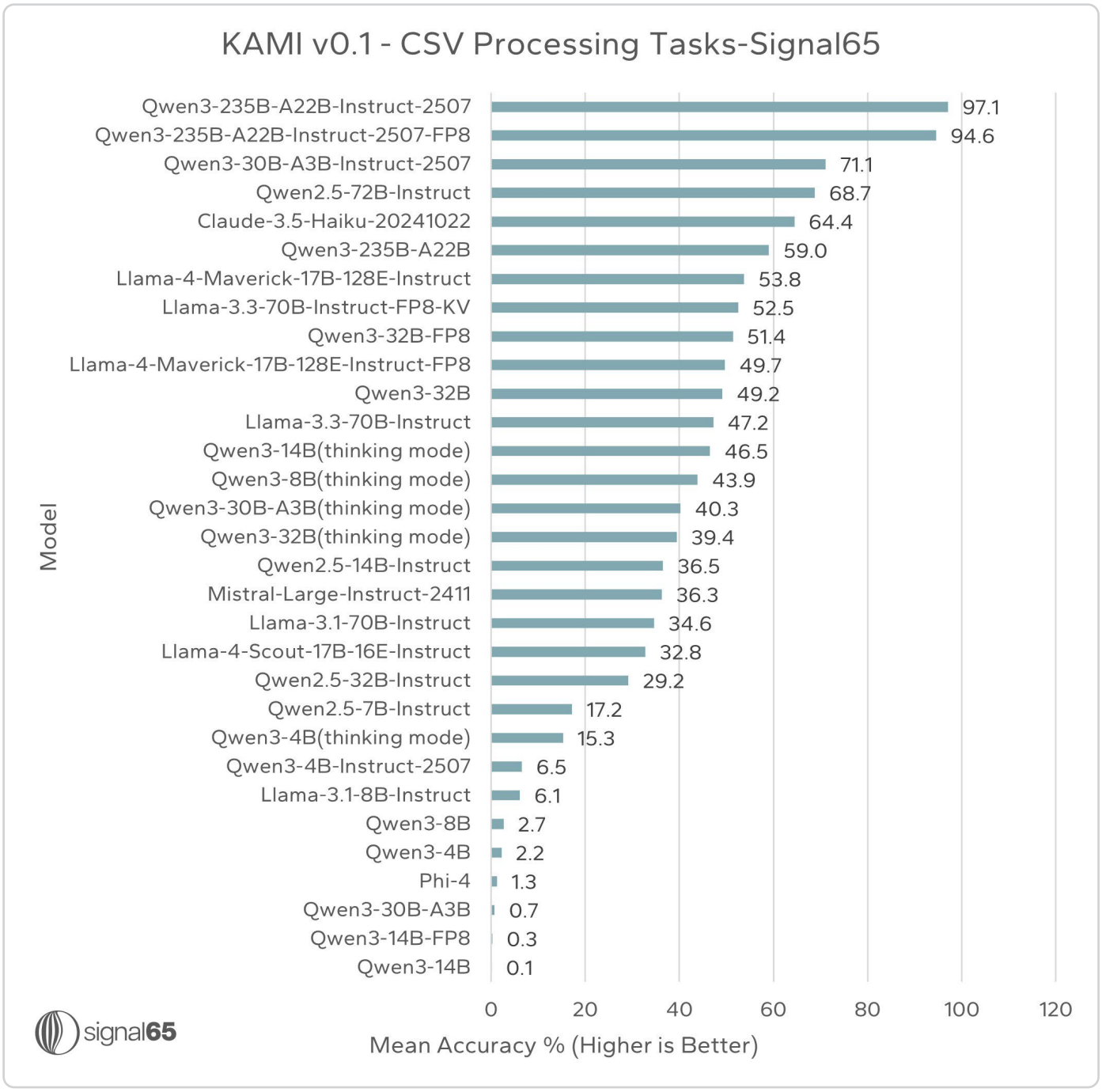


Figure 14: *KAMI v0.1 CSV Processing Tasks*

The CSV processing tasks proved to be challenging for most models tested, with only nine models scoring above 50% average accuracy for the three tasks. On average, the second question – which required models to analyze multiple CSV files and answer six specific questions – was found to be the most challenging. On average across all models, this question was answered with 22.2% accuracy. Comparatively, the first and third question had average accuracies of 51.4% and 37.7%, respectively.

An overview of the top five most accurate models can be seen below:

Model	CSV Question #1	CSV Question #2	CSV Question #3	Average Accuracy
Qwen3-235B-A22B-Instruct-2507	98.8%	95.4%	97.1%	97.1%
Qwen3-235B-A22B-Instruct-2507-FP8	94.2%	94.2%	95.4%	94.6%
Qwen3-30B-A3B-Instruct-2507	88.3%	70.4%	54.6%	71.1%
Qwen2.5-72B-Instruct	83.8%	41.0%	81.4%	68.7%
Claude-3.5-Haiku-20241022	100.0%	4.4%	88.9%	64.4%

Figure 15: *KAMI v0.1 CSV Processing Top 5*

Qwen3-235B-A22B-Instruct-2507 and Qwen3-235B-A22B-Instruct-2507-FP8 were the only models to achieve consistently high accuracy across all three CSV processing questions. While the other models in the top 5 were capable of answering the first CSV processing question with a high level of accuracy, they were all far less accurate for either one or both of the two more complicated CSV questions. One of the most notable examples of this trend is Claude-3.5-Haiku-20241022, which scored 100% on question #1 and 88.9% on question #3, yet achieved only 4.4% accuracy on question #2. This, again, presents a clear case in which a model can successfully achieve a useful agentic task – such as retrieving information from a CSV file – yet struggle once the same task becomes lengthy or complicated.

Database Processing (Standard)

For AI agents in enterprise settings, retrieving and analyzing data stored in SQL databases will be a core task. The KAMI v0.1 benchmark includes three questions to evaluate SQL database processing capabilities, similar to the previous CSV processing tasks. All three questions require the model to query a database in order to find specific business information, and save the results to a JSON file. The second question involves additional complexity by including six unique questions.

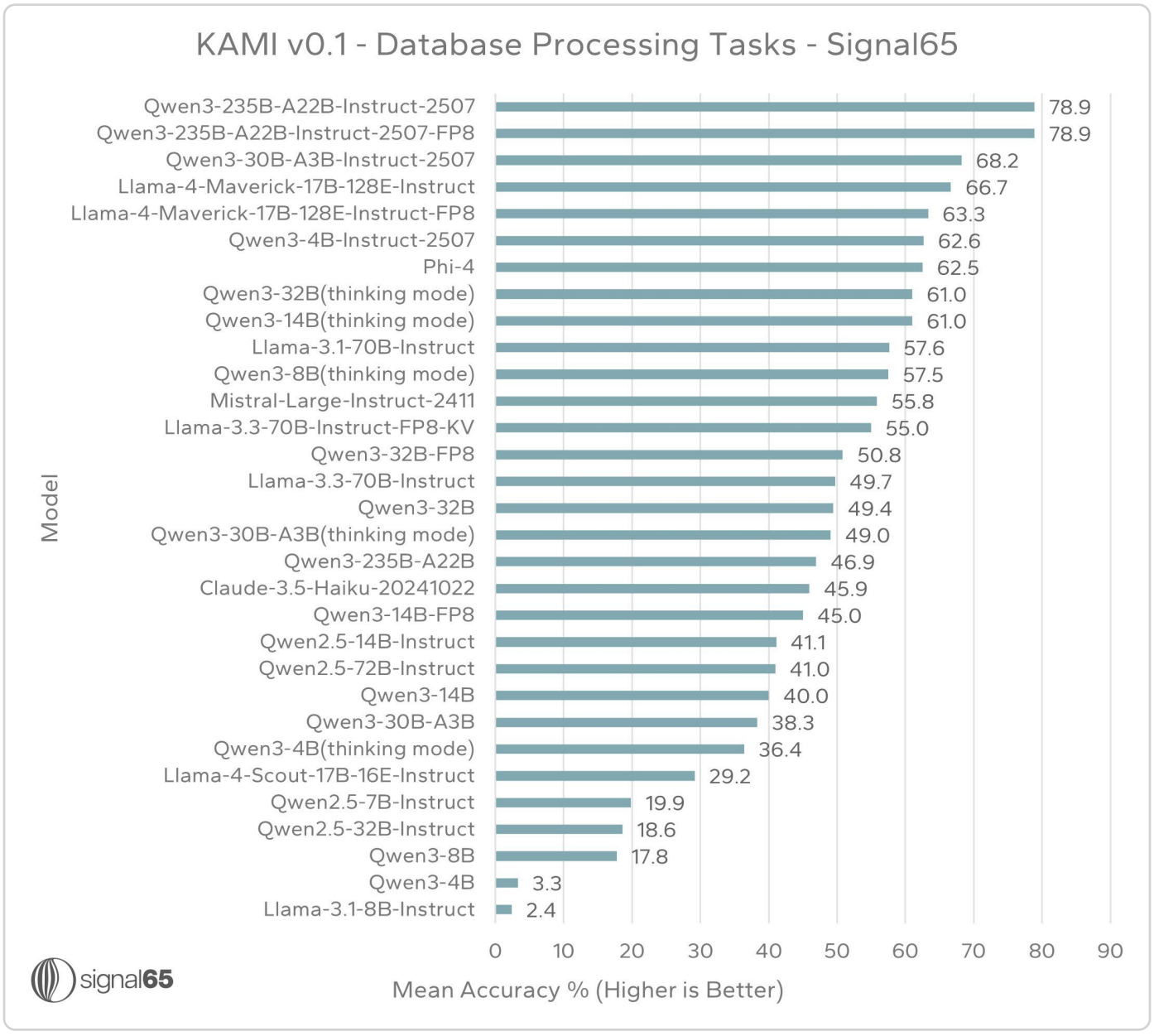


Figure 16: *KAMI v0.1 Database Processing Tasks*

On average, models were more successful in completing the database tasks than they were for the very similar CSV processing tasks. While the specific business questions asked were mirrored across the two test categories, the average accuracy for the database tasks rose to 46.9% from 37.1% for the CSV processing tasks. This greater overall success in answering database questions is likely attributed to an ability to query data using a well-defined language in SQL, compared to the more open-ended task of retrieving information from a CSV file.

In general, the models tested achieved high accuracy for both the first and third database processing question, but struggled significantly with the longer, more complex second question. On average across all models, question #1 was answered with 61.5% accuracy and question #3 was answered with 62.6% accuracy. Question #2, however, was only answered with 16.6% accuracy on average. This pattern can be seen clearly, even amongst the top performing models, shown in Figure 17.

Model	DB Question #1	DB Question #2	DB Question #3	Average Accuracy
Qwen3-235B-A22B-Instruct-2507	76.3%	71.7%	88.8%	78.9%
Qwen3-235B-A22B-Instruct-2507-FP8	70.4%	75.4%	90.8%	78.9%
Qwen3-30B-A3B-Instruct-2507	84.6%	23.8%	96.3%	68.2%
Llama-4-Maverick-17B-128E-Instruct	97.1%	48.3%	54.6%	66.7%
Llama-4-Maverick-17B-128E-Instruct-FP8	95.4%	42.5%	52.1%	63.3%

Figure 17: *KAMI v0.1 Database Processing Tasks Top 5*

While the two leading models, Qwen3-235B-A22B-Instruct-2507 and Qwen3-235B-A22B-Instruct-2507-FP8, maintain a relatively high scores across all three questions, the other top models all experience a notable drop-off in accuracy for question #2, with some additionally challenged by question #3. Some models outside the top 5 were even more challenged by the complexity of question #2, such as Qwen3-32B (thinking mode), which scored 87.9% for question #1 and 90.4% on question #3, yet only achieved 4.6% accuracy on question #2.

Database Processing (Guided)

In addition to the standard database processing tasks, the KAMI v0.1 benchmark includes two additional database questions that provide a hint to the model in the prompt. These questions closely mirror questions #1 and #2 from the standard database processing category, however, they include additional instruction to the models: “Begin by examining the schema to find relevant columns, and then do your analysis.” This hint assists models overcome a common problem found in AI database processing, in which they often attempt to query a database while guessing the schema. The second, and more complex question, includes an additional hint to assist with handling null values: “Note that if the requested company data is not present in the database, then assume the answer is 0 for the relevant question.”

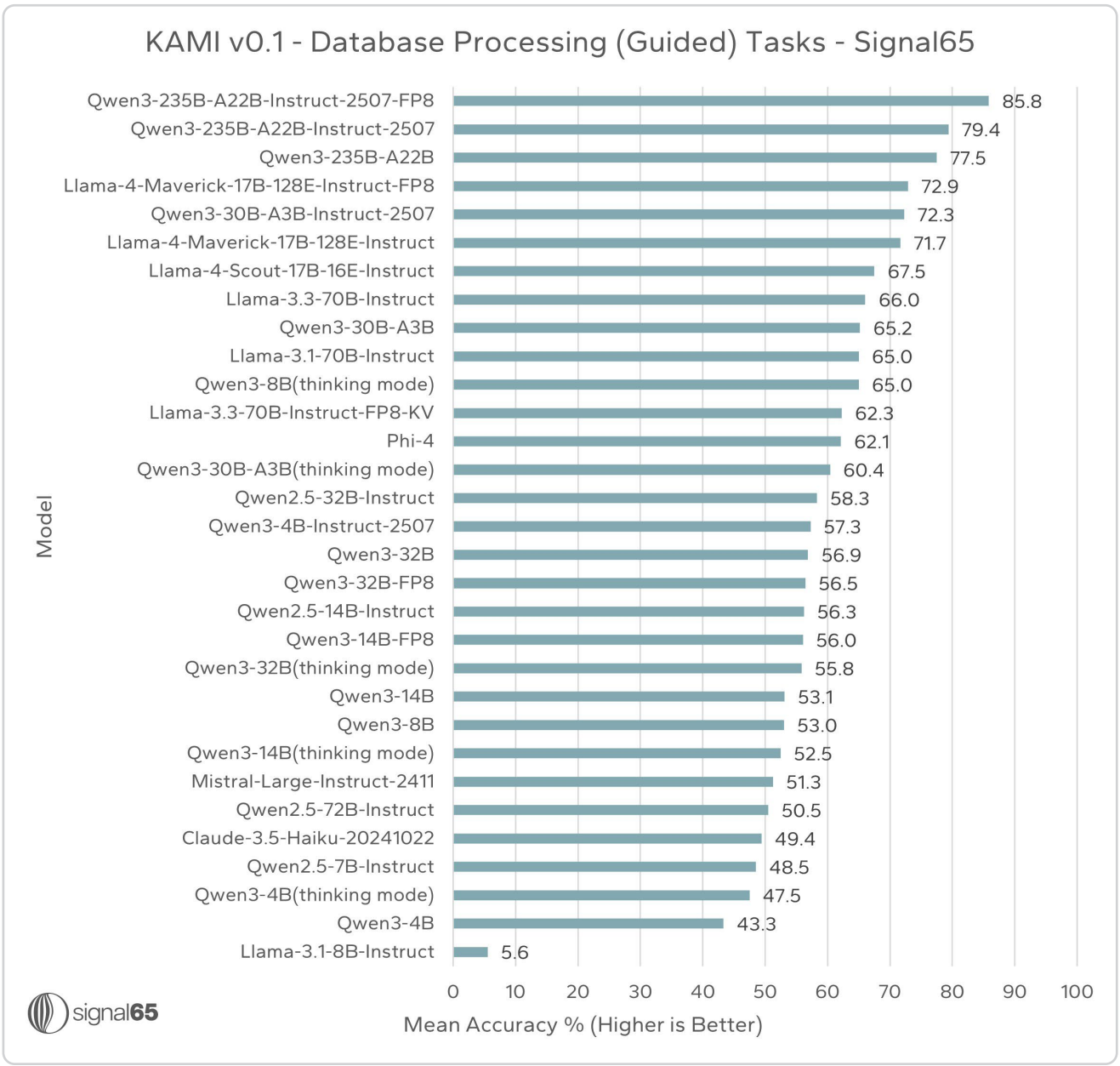


Figure 18: *KAMI v0.1 Database Processing with Hints Tasks*

With the addition of the hint, most models achieved a high level of success answering question #1. Twenty-two distinct models completed question #1 with 100% accuracy, with an additional seven achieving over 90% accuracy. The average accuracy for question #1 across all models tested was 96.2%, a notable improvement from the 62.5% accuracy achieved during the non-hinted database task. Llama-3.1-8B0-Instruct, which struggled with all tests in the KAMI benchmark, was the only model tested to achieve less than 80% accuracy for this question.

The hints, however, were less effective in improving model performance for the second, more complex question. The average accuracy across all models for question #2 was 21.6%, only a slight improvement from the 16.6% accuracy achieved without hints. This indicates that the challenges with answering question #2 stem more directly from the overall length and complexity of the task, than from simple errors that can be quickly improved with small hints.

An overview of the top 5 models for the hinted database questions can be seen in Figure 19.

Model	DB (Guided) Question #1	DB (Guided) Question #2	Average Accuracy
Qwen3-235B-A22B-Instruct-2507-FP8	100.0%	71.7%	85.8%
Qwen3-235B-A22B-Instruct-2507	100.0%	58.8%	79.4%
Qwen3-235B-A22B	100.0%	55.0%	77.5%
Llama-4-Maverick-17B-128E-Instruct-FP8	100.0%	45.8%	72.9%
Qwen3-30B-A3B-Instruct-2507	100.0%	44.6%	72.3%

Figure 19: *KAMI v0.1 Database Processing (Guided) Top 5*

The impact of the hints is immediately noticeable in the success of the top five models, with all models achieving 100% accuracy. The impact of the hints in question #2 is much more varied, with some models actually performing significantly worse when given the hints. Qwen3-235B-A22B-Instruct-2507, for example, achieved 71.7% accuracy without hints, but only achieved 58.8% accuracy when given hints. Another notable change within the top 5 models is the addition of Qwen3-235B-A22B as the 3rd most accurate model, which ranked 19th during the standard database tasks without hints.

Response Format Instruction Following

The final category tested in the KAMI v0.1 benchmark again closely follows the database tasks from the previous categories, but focuses on specific instructions for outputting the results. This category includes three questions, all of which ask the AI agent to query the same information as in the first hinted database question, but with different response formats. Since most models were previously found to be highly successful at achieving this task, lower scores for these tasks can be attributed to challenges in output formatting and instruction following, rather than database processing errors. The first task requires the agent to save the response as a text file with a numerical value, the second asks the model to respond with only a JSON formatted answer, and the third requires a numerical output only.

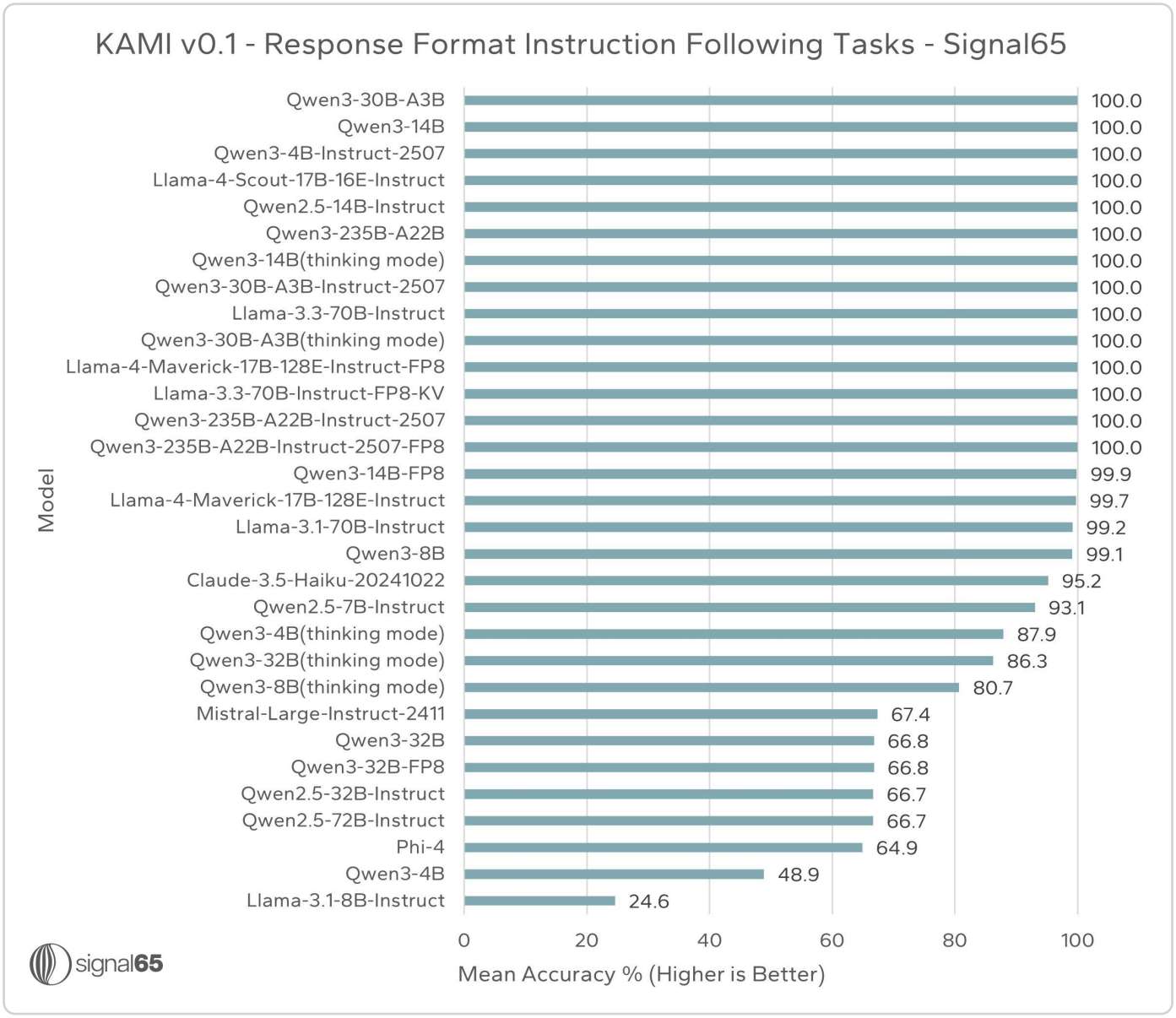


Figure 20: *KAMI v0.1 Response Format Instruction Following Tasks*

In general, most models performed these instruction following tasks successfully. Fourteen of the models tested scored 100% accuracy across all three tasks, while an additional six scored above 90%.

Notably, most models were found to be highly accurate for the first and the third tasks, with only two models – Qwen3-4B and Llama-3.1-8B-Instruct – scoring below 80% for either task. All other models with lower overall scores primarily struggled to follow the JSON formatting instructions of the second task. In many of these cases, the model would correctly query the information, and include a correctly formatted JSON response, but include additional erroneous text such as "Here is the text in JSON Format:". While such answers are nearly correct, they ignore the explicit instructions to output JSON only with no other text. These answers showcase that some models can correctly retrieve data and answer business questions, but lack explicit instruction following capability, which can become a problem for enterprise tasks requiring specific data formatting.

Model Size

A significant consideration for enterprises selecting LLMs is model size. In general, new model development has resulted in increasingly large models, with the general viewpoint being that larger models achieve better results. On the other hand, hardware limitations, and the associated cost considerations, may sway organizations to instead select smaller models. Additionally, ongoing model development and new architectures have led to the emergence of some small models that are thought to be highly competitive even with much larger LLMs.

By running the KAMI v0.1 benchmark across a wide range of LLMs, the results can be used to evaluate how models of various sizes achieve agentic tasks. To create such an evaluation, models have been grouped into four distinct groups:

- Small Models: < 10B Parameters
- Medium Models: 10B – 50B Parameters
- Large Models: 50B – 100B Parameters
- Very Large Models: > 100B Parameters

While there is still significant variation between models in each group, this rough grouping allows an overview of model performance based on size. It should be noted, that for this exercise, Mixture of Experts models were grouped by their total number of parameters rather than their active parameter counts. Additionally, Claude-3.5-Haiku-20241022 has been omitted from this analysis, as it does not have an officially documented parameter size.

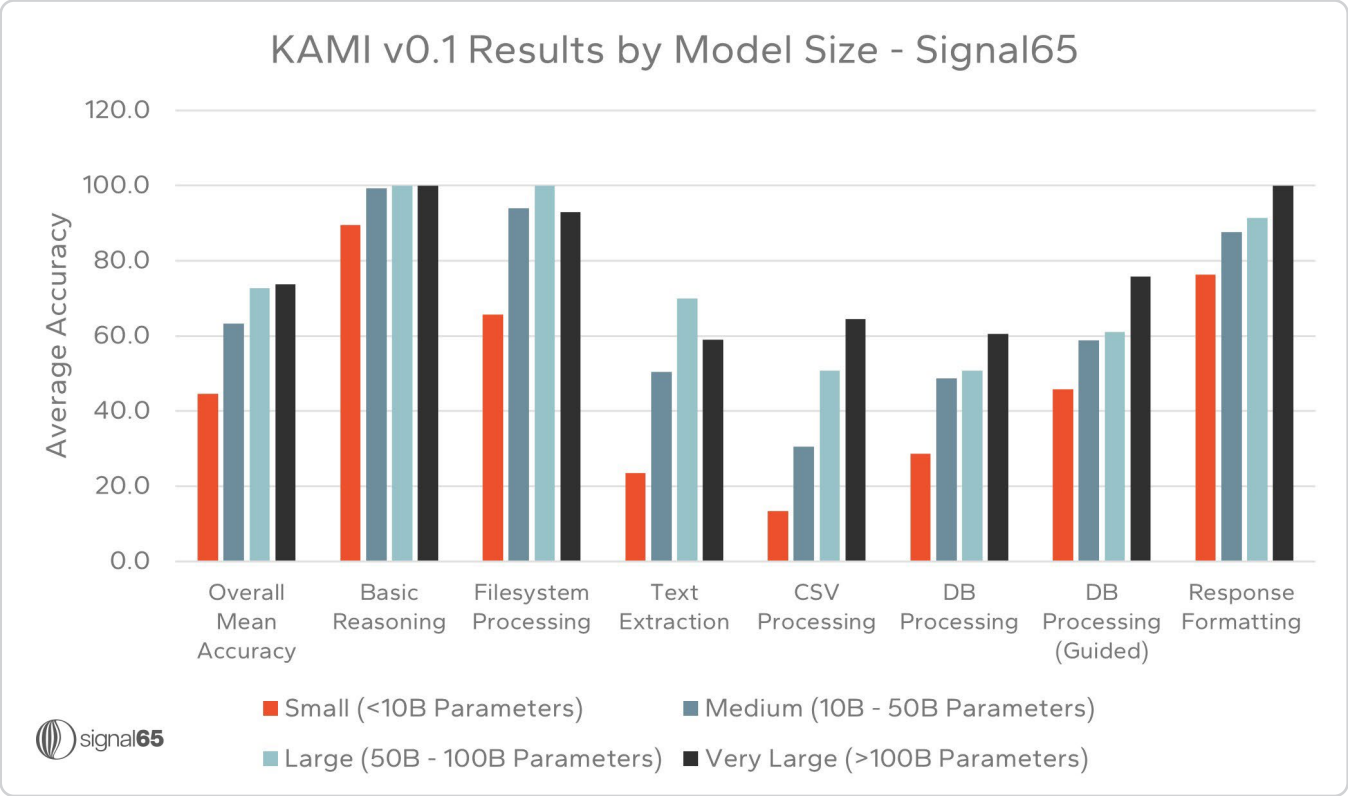


Figure 21: *KAMI v0.1 Results by Model Size*

The overall mean accuracy shows a clear trend that follows conventional thinking – on average, larger models perform better, with the large and very large model groupings far outperforming smaller models. The very large models (>100B parameters) show clear advantages across the CSV processing, database processing, and instruction following tasks, while the large (50B-100B parameters) group achieved the highest average scores for filesystem processing and text extraction. The medium models (10B-50B parameters), however, were found to remain competitive with the very large models for filesystem tasks and competitive with the large models for database processing and response formatting. The small models were found to be at a clear disadvantage across all test categories.

These results indicate that, in general, very small models may not be suitable choices for agentic AI applications. While they also indicate that the very large models will offer the greatest overall performance, the results for the medium and large models demonstrate that they may provide a reasonable choice for organizations seeking to balance hardware requirements, depending on the specific use case. Notably, the highest performing model in the medium category: Qwen3-30B-A3B (thinking mode), achieved an overall accuracy score 72.7%, which was highly competitive with the top models in the large grouping, as well as some of the lower performing models in the very large grouping.

While the very large model grouping is bolstered by the top performing Qwen3-235B-A22B-Instruct-2507-FP8 and Qwen3-235B-A22B-Instruct-2507-FP8 models, the remaining models were mostly matched, and in some cases outperformed, by the large model grouping. A notable example of this can be seen within the Llama model family, with 70B parameter models of the Llama-3.3 and Llama-3.1 generations directly competing with the newer and much larger Llama-4 models.

Model	Size	Overall Mean Accuracy
Llama-4-Maverick-17B-128E-Instruct	Very Large – 400B Parameters	74.6%
Llama-3.3-70B-Instruct-FP8-KV	Large – 70B Parameters	74.5%
Llama-3.1-70B-Instruct	Large – 70B Parameters	73.4%
Llama-4-Maverick-17B-128E-Instruct-FP8	Very Large – 400B Parameters	73.1%
Llama-3.3-70B-Instruct	Large -70B Parameters	71.6%
Llama-4-Scout-17B-16E-Instruct	Very Large – 109B Parameters	64.1%

Figure 22: *Llama Model Size Comparison*

Analyzing the KAMI v0.1 results by model size demonstrates that on average, agentic workloads favor larger model sizes, however, some models, such as Qwen3-30B-A3B (thinking mode) or Llama-3.1-70B-Instruct can achieve similar results with far fewer parameters. Finding such models that outperform their size range may enable enterprises to deploy effective agentic applications without stretching their overall infrastructure capabilities.

Quantization

Another aspect to evaluate within the KAMI v0.1 benchmark results is the impact of quantization on model performance. Quantization is typically utilized for efficiency; however, it comes with the risk of losing accuracy. Within the models selected for the KAMI v0.1 benchmark, several models were tested with both their full weights as well as a quantized fp8 version, enabling an evaluation of the quantization on agentic task performance.

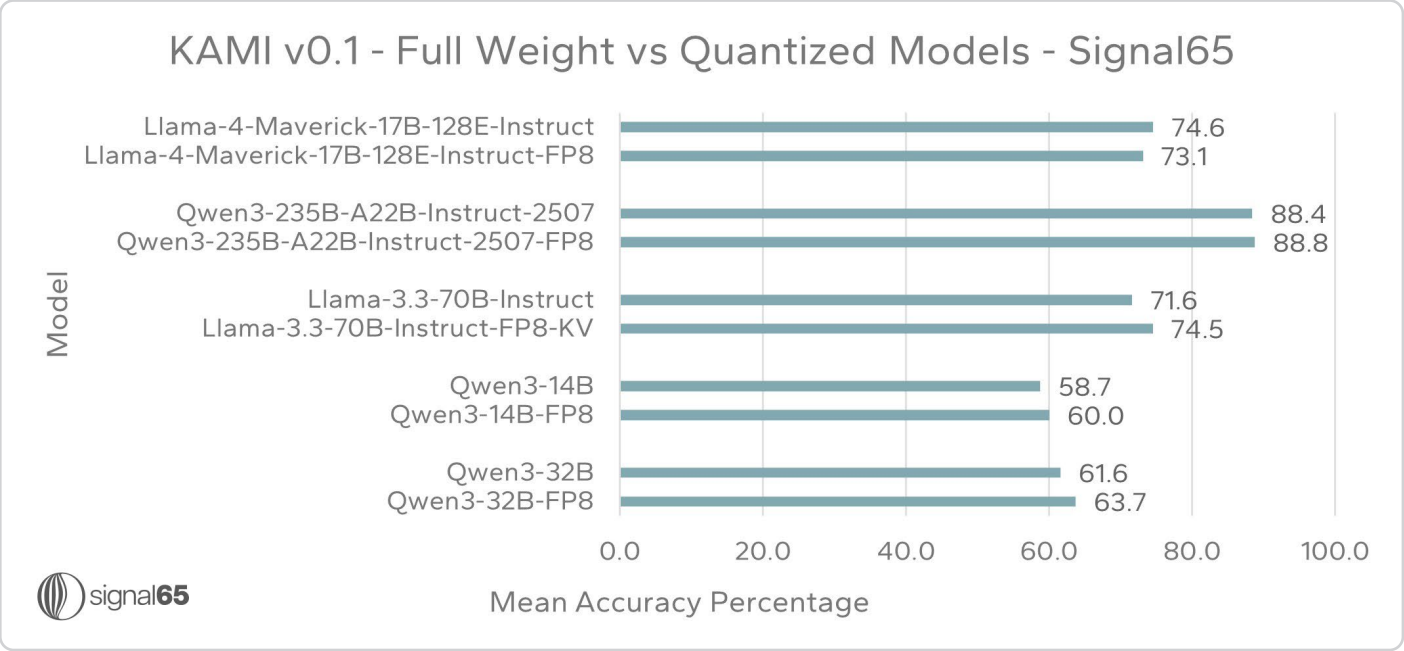


Figure 23: *KAMI v0.1 Full Weight vs Quantized Models*

Interestingly, in four of the five model pairs, the quantized version actually outperformed its full weight counterpart. In all cases, both models achieved very similar accuracies, and the differences may be a result of variation between test runs. While this testing does not provide enough to conclude that quantization actually improves model performance, it indicates that the loss of accuracy associated with fp8 quantization has minimal negative impact on completing agentic tasks.

Thinking vs Non-Thinking Models

In addition to including both quantized and non-quantized models for comparison, the KAMI v0.1 benchmark included several Qwen3 models that were run in both thinking and non-thinking modes, which enables a comparison of the impact of thinking on otherwise identical models.

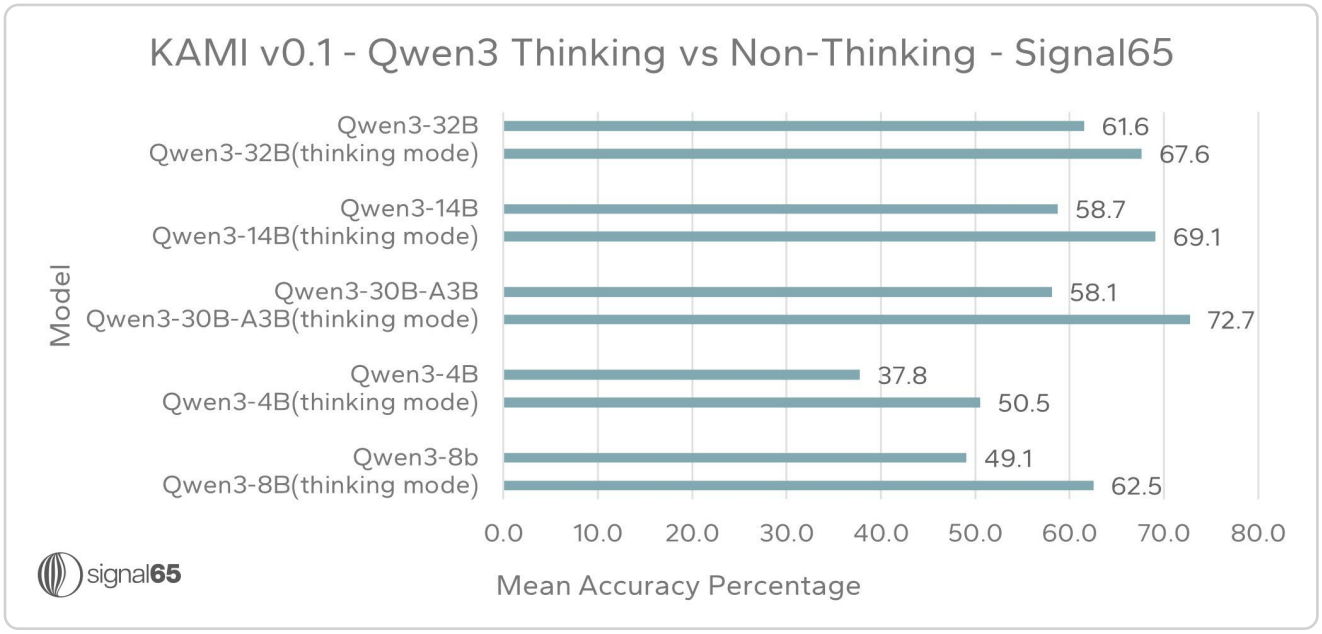


Figure 24: *Thinking vs Non-thinking Models Overview*

On average, all of the thinking-enabled Qwen3 variations outperformed their non-thinking counterparts. The thinking variations achieved notable advantages across the Information Finding, CSV Processing, and non-hinted database tasks. Interestingly, the inclusion of hints enabled much more competitive performance by the non-thinking models. The non-thinking models were additionally found to outperform the thinking versions during the filesystem tasks.

Model	Mean Accuracy	Basic Reasoning	Filesystem	Text Extraction	CSV Processing	DB	DB(Guided)	Instruction Following
Qwen3 Thinking models	64.5%	99.6%	80.7%	52.4%	37.1%	53.0%	56.3%	91.0%
Qwen3 Non-Thinking models	53.1%	99.9%	94.4%	36.6%	11.0%	29.8%	54.3%	83.0%

Figure 25: *Thinking vs Non-thinking Models by Category*

The higher accuracy of the thinking models logically makes sense, as they can utilize step-by-step thinking capabilities to reason through all the actions needed to complete the goal. The non-thinking versions, on the other hand, benefit significantly from additional prompting to supplement some of this thinking ability, which can be seen in the dramatic improvement in database processing when given explicit hints. It was noted, however, that in some of the longer and more complex tasks, thinking models experienced a significant loss of accuracy, as was observed amongst Qwen3 thinking models during the complex filesystem task. While thinking appears to offer significant benefits for many agentic tasks, the additional thinking steps also bring the potential to introduce additional errors. The challenges of thinking models during the longer tasks may be attributed to the additional thinking steps propagating unnecessary errors throughout the completion of multi-step tasks.

Comparison to other Benchmarks

The KAMI v0.1 benchmark introduces a new tool into the quickly developing space of LLM benchmarking. While other AI benchmarks can be useful indicators of AI reasoning and other specific capabilities, there is typically a disconnect between benchmark scores and real world agentic capability. This can be seen when comparing KAMI v0.1 scores with other benchmark results. Figure 26 displays the KAMI v0.1 scores for five of the models in the Qwen model family, along with additional AI benchmark scores that were released at the launch of Qwen3[1].

Benchmark	Qwen3-235B-A22B	Qwen3-30B-A3B	Qwen3-32B	Qwen3-4B	Qwen2.5-72B-Instruct
KAMI v0.1	67.7	58.1	61.6	37.8	71.1
ArenaHard	95.6	91.0	89.5	76.6	81.2
AIME'24	85.7	80.4	79.5	73.8	18.9
AIME'25	81.5	70.9	69.5	65.6	15.0
LiveCodeBenchv5	70.7	62.6	62.7	54.2	30.7
CodeForces	2056	1974	1982	1671	859
LiveBench(2024-11-25)	77.1	74.3	72.0	63.6	51.4
BFCLv3	70.8	69.1	66.4	65.9	63.4
MultiIF	71.9	72.2	68.3	66.3	65.3

Figure 26: *Benchmark Comparison*

[1]Qwen3: Think Deeper, Act Faster | Qwen

Notably, the benchmarks consistently show the Qwen3 models outperforming Qwen-2.5-72B-Instruct, even at much smaller sizes. The KAMI results, however, show the older generation Qwen-2.5-72B-Instruct outperforming each of the four other models. This comparison highlights the disconnect between common benchmarks and a model's ability to actually perform common agentic tasks. A clear discrepancy can be seen with Qwen3-4B, which shows an impressive ability to compete with and even outperform its larger counterparts across many of the benchmarks. In the KAMI v0.1 tests, however, Qwen3-4B achieved only 37.8% accuracy – the second lowest of all models tested.

Of particular note, are the results of the BFCLv3 benchmark. This benchmark focuses on multi-step tool calling, making it highly relevant to evaluating agentic capabilities, however there is a clear discrepancy with the KAMI results. In the BFCLv3 results, Qwen3-235B-A22B achieves the highest score, with the other three models achieving fairly competitive results, ranging from 69.1 to 63.4. As with many of the other benchmarks, the BFCLv3 scores show all of the Qwen3 models outperforming Qwen2.5-72B-Instruct. In the KAMI scores, however, not only is Qwen2.5-72B-Instruct the highest performing model, but there is a much larger variance between the remaining models.

The discrepancies between these benchmarks underline the importance of evaluating models based on intended use cases, and utilizing the correct tools to do so. In general, evaluating models based on a broad set of benchmarks is useful for organizations to gain a comprehensive understanding of a model's overall strengths and weaknesses. From there, use case specific benchmarks and tests can be utilized to evaluate how a model may perform for specific applications. When evaluating models for agentic use cases, however, the vast differences between KAMI and other popular benchmarks highlight the difficulty enterprises face during model evaluation. While models may perform well on various benchmarks, they may not actually be able to complete routine enterprise tasks, as tested in the KAMI benchmark, and therefore result in unsuccessful agentic deployments.

Final Thoughts: Utilizing KAMI for Enterprise AI Evaluation

The KAMI v0.1 Benchmark, and our Signal65 analysis of these first results, serve as tools to evaluate how LLMs perform in realistic agentic scenarios. KAMI differentiates itself from other popular LLM benchmarking approaches by utilizing dynamic question generation and a focus on real-world agentic tasks. For enterprise organizations building agentic AI applications, this approach offers valuable insight into model performance and can assist organizations in selecting the right models to build successful AI agents.

The KAMI v0.1 Benchmark enables organizations to quickly evaluate LLMs based on an ability to complete actual agentic tasks, rather than rely solely on logic-based question answering or memorization. In addition to the overall KAMI rankings, by evaluating models based on their performance in specific use case categories, organizations can more carefully select models that excel for their intended use cases. Further exploration into specific results and failure scenarios observed during testing may additionally provide guidance on model selection, as well as approaches to effectively build agents and prompt models.

Key Highlights

- Qwen3-235B-A22B Instruct-2507-FP8 is the top agentic AI performer at **88.8% mean accuracy score**
- FP8 quantization impacts accuracy by **~3% for agentic workloads**
- Thinking models **up to 25% more accurate** for agentic AI workloads

Understanding the agentic capability of models becomes increasingly important for enterprise organizations when considering multi-agent applications, in which multiple different models may be required to maximize the potential of each agent. Selecting accurate models for each agent role becomes additionally important as a single inaccurate agent can impact the entire application, however, it also further complicates the model selection processes. By utilizing the KAMI benchmark, Signal65 can provide context and data to help educate organizations and gain a deeper understanding into the strengths and weaknesses of various LLMs in real agentic scenarios.

The KAMI v0.1 benchmark provides illuminating results around the agentic capability of several popular models, with Qwen3-235B-A22B-Instruct-2507 showcasing itself as a top model to be considered for agentic workloads. The results additionally provide insight into how variables such as model size, quantization, and thinking ability can impact agentic performance. The results also notably diverge from other common AI benchmarks, highlighting the disconnect between typical AI benchmarking and agentic AI and challenging the consensus around model performance.

Future iterations of KAMI intend to expand the current v0.1 benchmark to evaluate additional agentic AI use cases and further expand the total number of models tested. While the v0.1 benchmark focuses on broad use case categories, such as general database processing, future versions intend to include more granular, vendor-specific evaluations, such as evaluating proficiency in Oracle, PostgreSQL, and SQLServer. These iterations will also likely include greater evaluation of popular proprietary models. With an enhanced test suite, future releases of the KAMI benchmark can enable greater model evaluation and help Signal65 develop test plans and comparisons to guide organizations in overcoming key challenges of building AI agents.

Appendix

Full Results

Model	Overall Mean Accuracy
Qwen3-235B-A22B-Instruct-2507-FP8	88.8%
Qwen3-235B-A22B-Instruct-2507	88.4%
Claude-3.5-Haiku-20241022	75.9%
Llama-4-Maverick-17B-128E-Instruct	74.6%
Llama-3.3-70B-Instruct-FP8-KV	74.5%
Llama-3.1-70B-Instruct	73.4%
Llama-4-Maverick-17B-128E-Instruct-FP8	73.1%
Qwen3-30B-A3B (thinking mode)	72.7%
Llama-3.3-70B-Instruct	71.6%
Qwen2.5-72B-Instruct	71.1%
Qwen3-30B-A3B-Instruct-2507	69.6%
Qwen3-14B (thinking mode)	69.1%
Qwen3-235B-A22B	67.7%
Qwen3-32B (thinking mode)	67.6%
Qwen2.5-14B-Instruct	66.6%
Llama-4-Scout-17B-16E-Instruct	64.1%
Qwen3-32B-FP8	63.7%
Qwen3-8B (thinking mode)	62.5%
Qwen3-32B	61.6%
Qwen3-4B-Instruct-2507	60.0%
Qwen3-14B-FP8	60.0%
Mistral-Large-Instruct-2411	58.9%
Qwen3-14B	58.7%

Qwen3-30B-A3B	58.1%
Qwen2.5-32B-Instruct	55.9%
Phi-4	54.8%
Qwen3-4B (thinking mode)	50.5%
Qwen3-8B	49.1%
Qwen2.5-7B-Instruct	41.6%
Qwen3-4B	37.8%
Llama-3.1-8B-Instruct	10.5%

Model Size Groupings

Small (<10B Parameters)	
Model	**Overall Mean Accuracy**
Qwen3-8B (thinking mode)	62.5%
Qwen3-4B-Instruct-2507	60.0%
Qwen3-4B (thinking mode)	50.5%
Qwen3-8B	49.1%
Qwen2.5-7B-Instruct	41.6%
Qwen3-4B	37.8%
Llama-3.1-8B-Instruct	10.5%
Average	**44.6%**

Medium (10B - 50B Parameters)	
Model	**Overall Mean Accuracy**
Qwen3-30B-A3B (thinking mode)	72.7%
Qwen3-30B-A3B-Instruct-2507	69.6%
Qwen3-14B (thinking mode)	69.1%
Qwen3-32B (thinking mode)	67.6%
Qwen2.5-14B-Instruct	66.6%
Qwen3-32B-FP8	63.7%
Qwen3-32B	61.6%
Qwen3-14B-FP8	60.0%
Qwen3-14B	58.7%
Qwen3-30B-A3B	58.1%
Qwen2.5-32B-Instruct	55.9%
Phi-4	54.8%
Average	**63.2%**

Large (50B - 100B Parameters)	
Model	**Overall Mean Accuracy**
Llama-3.3-70B-Instruct-FP8-KV	74.5%
Llama-3.1-70B-Instruct	73.4%
Llama-3.3-70B-Instruct	71.6%
Qwen2.5-72B-Instruct	71.1%
Average	**72.7%**

Very Large	
Model	**Overall Mean Accuracy**
Qwen3-235B-A22B-Instruct-2507-FP8	88.8%
Qwen3-235B-A22B-Instruct-2507	88.4%
Llama-4-Maverick-17B-128E-Instruct	74.6%
Llama-4-Maverick-17B-128E-Instruct-FP8	73.1%
Qwen3-235B-A22B	67.7%
Llama-4-Scout-17B-16E-Instruct	64.1%
Mistral-Large-Instruct-2411	58.9%
Average	**73.6%**

Important Information About this Report

CONTRIBUTORS
Mitch Lewis
Performance Analyst | Signal65

PUBLISHER
Ryan Shrout
President and GM | Signal65

INQUIRIES
Contact us if you would like to discuss this report and Signal65 will respond promptly.

CITATIONS
This paper can be cited by accredited press and analysts, but must be cited in-context, displaying author's name, author's title, and "Signal65." Non-press and non-analysts must receive prior written permission by Signal65 for any citations.

DISCLOSURES
Signal65 provides research, analysis, advising, and consulting to many high-tech companies, including those mentioned in this paper. No employees at the firm hold any equity positions with any companies cited in this document.

ABOUT SIGNAL65
Signal65 is an independent research, analysis, and advisory firm, focused on digital innovation and market-disrupting technologies and trends. Every day our analysts, researchers, and advisors help business leaders from around the world anticipate tectonic shifts in their industries and leverage disruptive innovation to either gain or maintain a competitive advantage in their markets.

CONTACT INFORMATION
Signal65 I signal65.com