Accelerating Multimodal RAG with Intel Xeon and OpenVINO: When Vision Meets Language

Planning / Implementation

As Retrieval-Augmented Generation (RAG) evolves, the integration of multimodal inputs such as images, text, tables and even voice mark a critical shift in real-world AI applications. Traditional RAG systems focus primarily on text-based retrieval and generation. In contrast, Multimodal RAG incorporates visual and textual understanding into a unified pipeline, enabling more contextually rich and semantically complete reasoning.

This study explores the performance of Multimodal RAG across different hardware and optimization configurations, emphasizing how Intel Xeon 6 processors paired with OpenVINO optimizations can achieve GPU-like performance in visual-language tasks. We evaluate the performance of two representative models, Phi-3.5-vision-instruct (4B) and Mistral-7B-Instruct-v0.1, across several backend configurations on Intel Xeon 6787P, including FP16 precision, OpenVINO FP16 precision, and OpenVINO INT8 precision.

This workflow below illustrates how OpenVINO enables an end-to-end path from model development to optimized deployment across heterogeneous Intel architectures. By supporting direct model conversion, low-precision optimization, and a unified runtime, OpenVINO allows vision and language models to be efficiently deployed on Xeon CPUs and other Intel platforms without changing application logic.

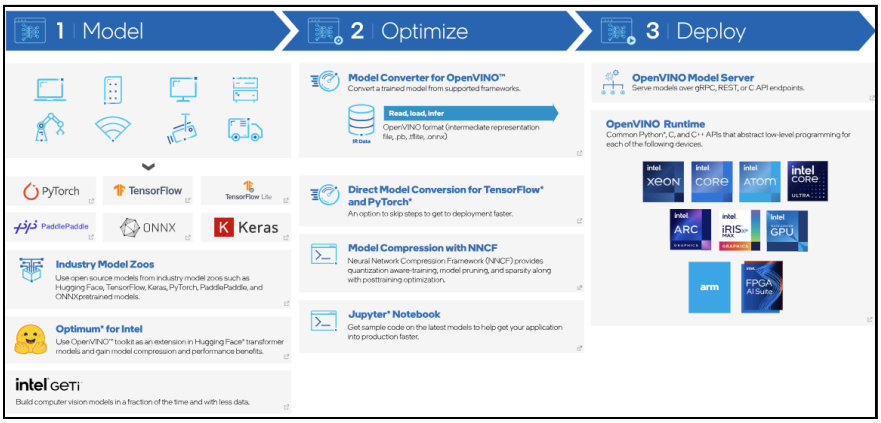


Figure 1. OpenVINO Summary Information

The figure below presents the multimodal RAG pipeline employed in this study. In contrast to traditional text-only RAG, multimodal RAG incorporates both textual and visual inputs. Images are transformed into textual representations via metadata extraction, and captioning. And then embedded alongside native text into a unified vector space. User queries are encoded in the same space, enabling cross-modal retrieval. Retrieved content is then routed to either a standard LLM or a vision-language model, depending on its modality, to generate the final response.

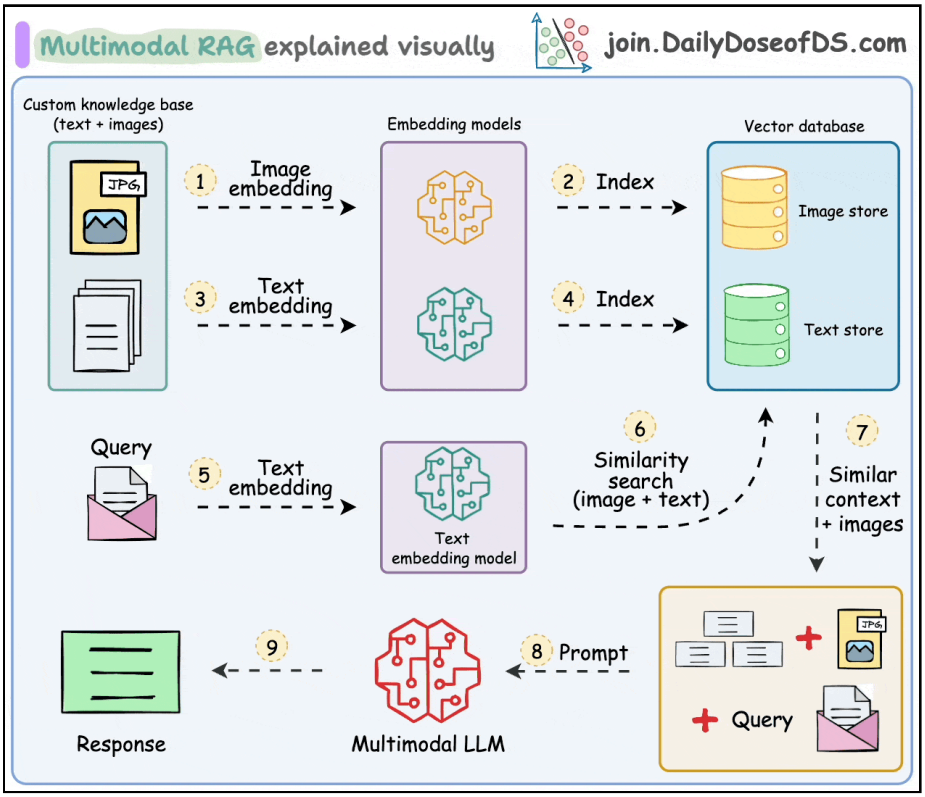


Figure 2. Diagram of Multimodal RAG (from DailyDoseofDS.com; used with permission)

Evaluation Metrics & Model Selection

This section evaluates multimodal RAG performance using latency-focused metrics that capture end-user responsiveness. We will cover Time-to-First-Token (TTFT) and Time-per-Output-Token (TPOT) which serve as the primary measures of inference efficiency. For model selection, we pair Phi-3.5-vision-instruct (4B) for visual understanding with Mistral-7B-Instruct-v0.1 for text generation, representing a typical multimodal RAG workflow. These choices enable controlled and comparable assessment across CPU, OpenVINO-optimized, and GPU baseline configurations.

We are focusing on the performance metrics that matter for production RAG:

- **TTFT (Time to First Token)** (s): interactive latency; lower is better. It captures the delay before the system acknowledges a user request and begins producing the first output.
- **TPOT(Time Per Output Token)** (ms): average time to generate each individual output token after first token, also known as inter-token latency; lower is better. It dominates the total response time for long-form outputs, where hundreds of tokens may be generated. Therefore, inter-token latency becomes the key.

We selected the following models:

- **Vision-Language Model:** Phi-3.5-vision-instruct (4B parameters), capable of visual grounding and multimodal reasoning.
- **Language Model (LLM):** Mistral-7B-instruct-v0.1, optimized for high-quality text generation and instruction following.

The models are described in the following links:

- Microsoft/Phi-3.5-vision-instruct: https://huggingface.co/microsoft/Phi-3.5-vision-instruct
- Mistralai/Mistral-7B-Instruct-v0.1: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

All experiments were performed on a controlled evaluation environment comparing native FP16, OpenVINO FP16, and OpenVINO INT8 inference on Intel Xeon 6787P. This setup isolates the impact of backend optimization and precision reduction on multimodal RAG latency.

The following table summarizes the hardware platforms, inference backends, and precision modes evaluated in this study, serving as the basis for all performance comparisons reported in the following sections.

Table 1. Hardware and Backend Configurations Evaluated in This Study

Setup ID	Platform	Backend	Precision	Notes
1	Xeon 6787P	Huggingface	FP16	Baseline CPU inference
2	Xeon 6787P	OpenVINO	FP16	Optimized via OpenVINO runtime
3	Xeon 6787P	OpenVINO	INT8	Quantized inference for latency reduction

Performance Results & Analysis

This section presents the measured latency characteristics of the multimodal RAG pipeline, focusing on Time-to-First-Token (TTFT) and Time-per-Output-Token (TPOT) across all tested backends. Results are organized by workload and platform, highlighting how OpenVINO optimizations on 6th-Gen Intel Xeon (6787P) substantially reduce inference latency and enable competitive performance relative to GPU baselines. These findings provide practical guidance for meeting SLA targets and planning multimodal RAG deployment capacity.

Time to First Token (TTFT)

The chart below shows that OpenVINO effectively minimizes startup latency through optimized precision compression and system runtime.

- OpenVINO dramatically reduces TTFT on Xeon 6787P.
  - Native FP16 shows higher startup latency (4.9 s for Phi-3.5-vision-instruct, 3.2 s for Mistral-7B).
  - With OpenVINO FP16 and INT8, TTFT drops to about 0.5 s, an near 10× reduction, largely due to graph compilation, operator fusion, and optimized memory scheduling.
- OpenVINO brings CPU TTFT close to GPU-class behavior.
  - While the H100 FP16 baseline achieves ~0.08 s, the 0.5 s TTFT on Xeon with OpenVINO is within practical bounds for interactive multimodal use and represents a major narrowing of the CPU - GPU gap.

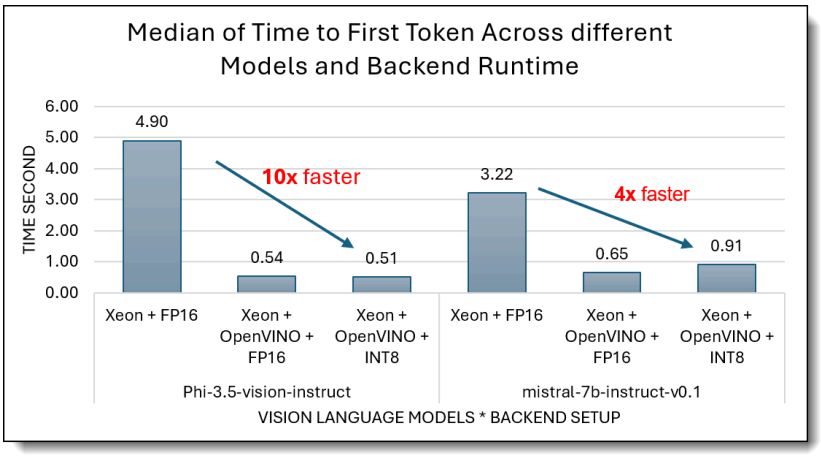


Figure 3. Time to First Token (lower is better)

Time per Output Token (TPOT)

The chart below shows how Intel Xeon 6787P CPUs and OpenVINO successfully reduce the Time-per-Output-Token (TPOT) to a performance level that meets the SLA standards.

- For Phi-3.5-vision-instruct, native FP16 produces slow generation (about b560 ms/token), while OpenVINO FP16 (31 ms/token) and OpenVINO INT8 (18 ms/token) reduce latency by 18× and 30×, respectively. These improvements stem from optimized kernel execution, reduced precision compute, and efficient operator fusion.
- For Mistral-7B-Instruct, OpenVINO INT8 reaches about 26 ms/token, comfortably exceeding real-time interaction thresholds; for reference, human reading speed is 90 ms/token (≈6 words/sec), meaning the model streams tokens faster than users can consume them.

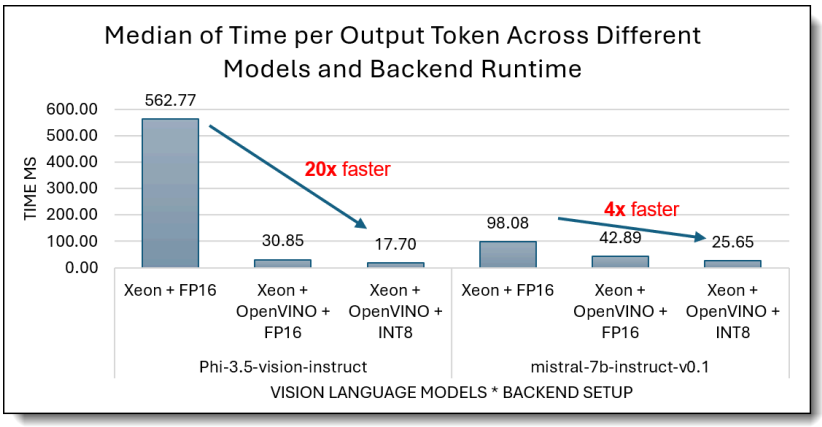


Figure 4. Time per Output Token (lower is better)

Generation Latency for Language and Vision Language Model

This section examines generation latency for vision-language workloads as output length increases, focusing on how backend optimization and precision affect end-to-end response time. By evaluating multiple output lengths, we capture the cumulative impact of per-token latency on long-form multimodal generation, which is critical for document understanding, visual reasoning, and agent-driven workflows.

The data and figures below show that generation latency grows linearly with output length for both Phi-3.5-vision-4B and Mistral-7B models, making per-token efficiency the dominant factor for long responses. Across all output sizes, OpenVINO significantly reduces total generation time compared to native FP16 execution, with INT8 consistently delivering the lowest latency. For Phi-3.5-vision-4B, OpenVINO INT8 achieves up to a 97% reduction in total generation latency at long output lengths, while Mistral-7B shows a 74% reduction, demonstrating that low-precision inference is especially effective in controlling latency growth as output length scales.

Table 2. Generation latency for Phi-3.5-vision-v0.1-4B based various output length

Backend	Time to First Token (s)	Time per Output Token (ms)	Latency @50 Output (s)	Latency @100 Output (s)	Latency @200 Output (s)	Latency @500 Output (s)
Xeon + FP16	4.9	562.8	33.0	61.2	117.5	286.3
Xeon + OpenVINO + FP16	0.5	30.9	2.1	3.6	6.7	16.0
Xeon + OpenVINO + INT8	0.5	17.7	1.4	2.3	4.0	9.4

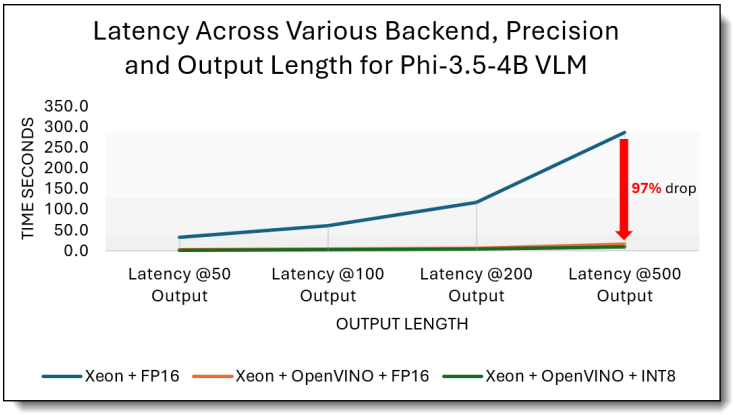


Figure 5. Generation Latency per output length - Phi-3.5-vision-instruct (lower is better)

Table 3. Generation latency for Mistral-v0.1-7B based various output length

Backend	Time to First Token (s)	Time per Output Token (ms)	Latency @50 Output (s)	Latency @100 Output (s)	Latency @200 Output (s)	Latency @500 Output (s)
Xeon + FP16	3.2	98.1	8.1	13.0	22.8	52.3
Xeon + OpenVINO + FP16	0.6	42.9	2.8	4.9	9.2	22.1
Xeon + OpenVINO + INT8	0.9	25.7	2.2	3.5	6.0	13.7

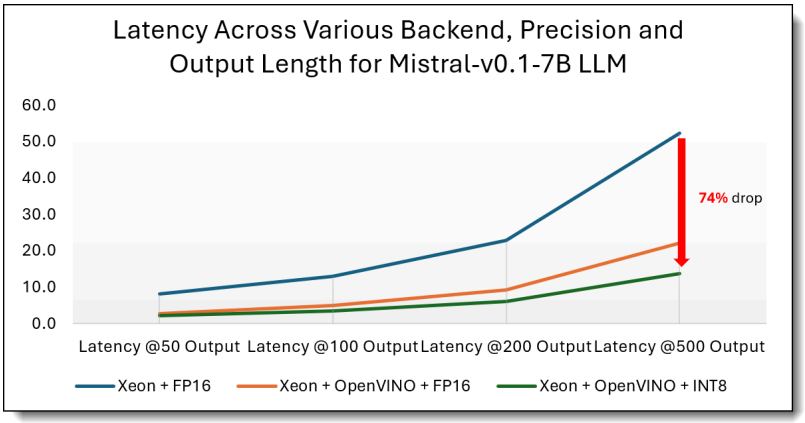


Figure 6. Generation Latency per output length - Mistral-7B-instruct (lower is better)

Conclusion and Business Impact

This study demonstrates that OpenVINO substantially accelerates both latency and generation speed for multimodal RAG workloads on Intel Xeon 6787P. Across vision-language and text-only models, OpenVINO optimizations reduce Time-to-First-Token (TTFT) and Time-per-Output-Token (TPOT) by an order of magnitude compared to native CPU inference. These gains transform Xeon from a baseline CPU platform into a production-capable engine for interactive and long-form multimodal generation.

This section synthesizes the performance comparison and findings along with a visualization of multimodal RAG demo in real-world deployment.

Business Impact based on Real-Time SLA Targets

To assess production readiness, we evaluate OpenVINO-enabled serving against practical real-time SLA thresholds. We conducted benchmark on vLLM and with openVINO as backend and continuous batching for Language Model, and OpenVINO Model Server with continuous batching for Vision Language Models

The following summarizes the maximum sustainable concurrency on a single Intel Xeon 6787P system with OpenVINO while meeting real-time responsiveness targets:

- For the LLM experiments, we used vLLM with continuous batching and issued 200 prompts drawn from the ShareGPT_V3_unfiltered_cleaned_split.json dataset
- For the VLM experiments, we used OpenVINO Model Server (OVMS) with continuous batching and evaluated 200 prompts from lmarena-ai/vision-arena-bench-v0.1.

For each model size and precision mode (FP16, INT8), we increased concurrency until the P90 latency exceeded the SLA thresholds, and report the highest concurrency that still satisfies P90 TTFT ≤ 10 seconds and P90 TPOT ≤ 130 ms/token. Overall, the table provides a practical capacity-planning view of how precision and model size translate into supported concurrent users for both text-only and multimodal serving on Xeon.

Table 4. Maximum Concurrent Users under Real-Time SLA (TTFT ≤ 10s, TPOT ≤ 130ms) on Xeon 6787P with OpenVINO

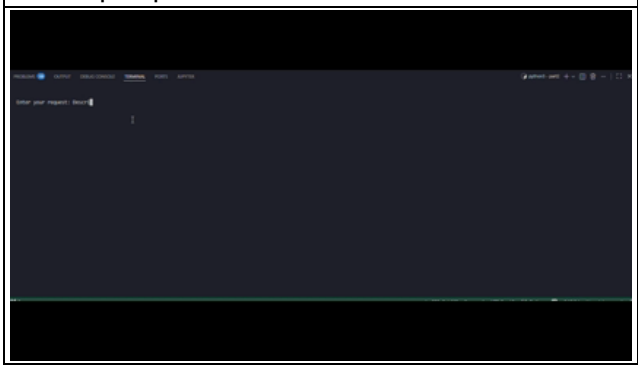
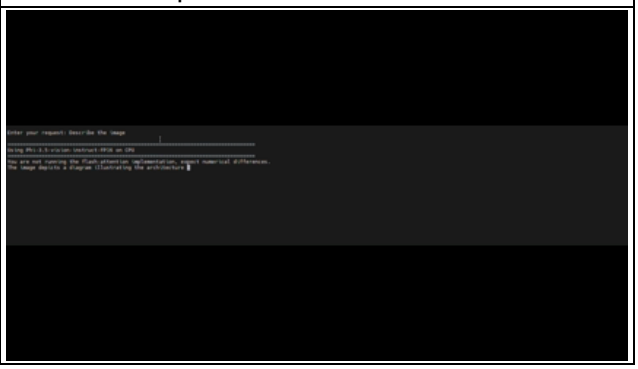
Model Size	LLM (FP16)	LLM (INT8)	VLM (FP16)	VLM (INT8)
3B	40 users	44 users	8 users	10 users
8B	30 users	32 users	4 users	5 users
14B	1 users	2 users	1 users	2 users
20B	15 users	16 users	-	-

Under practical real-time SLA targets (TTFT ≤ 10s, TPOT ≤ 130ms), OpenVINO with INT8 precision enables a single Xeon 6787P system to support dozens of simultaneous users for small-to-mid-sized language models and multiple concurrent multimodal sessions, without GPU acceleration. In concrete terms, one server can serve up to 44 concurrent users for 3B LLMs, 32 users for 8B LLMs, and 10 users for 3B vision-language models, translating into higher infrastructure utilization, lower per-user cost, and greater deployment flexibility for enterprise RAG and copilot workloads.

Multimodal Latency Demonstration

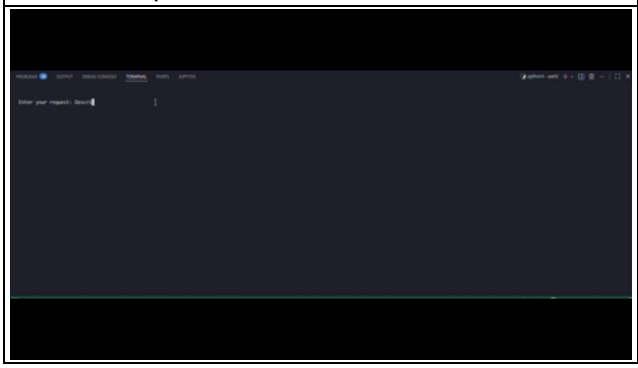
To complement quantitative benchmarks, we present a side-by-side latency demonstration comparing Intel Xeon 6787P with OpenVINO against plain CPU for multimodal generation. The OpenVINO-optimized Xeon system can deliver smooth, real-time token streaming with latency well below human consumption rates.

Figure 7. Side by side model generation on Xeon CPU with OpenVINO (left) and without OpenVINO (right) on Vision Language Model

Xeon CPU plus OpenVINO Generation for Phi-3.5 VLM	Xeon CPU without OpenVINO Generation for Phi-3.5 VLM
	

The following figure shows a side-by-side demonstration of multimodal response generation using OpenVINO INT8 (left) and without OpenVINO optimization (right).

Figure 8. Side by side model generation on Xeon CPU with and without OpenVINO on Language Model

Xeon CPU with OpenVINO Generation for Mistral-v0.1 LLM	Xeon CPU without OpenVINO Generation for Mistral-v0.1 LLM
	

The demonstration clearly shows that OpenVINO is the decisive factor in enabling real-time multimodal generation on Xeon 6787P:

- Without OpenVINO (animations on the right), native FP16 execution exhibits noticeably slower token emission and higher end-to-end latency, resulting in delayed and uneven response streaming.
- In contrast, OpenVINO-optimized FP16 and INT8 inference (animations on the left) delivers immediate token generation and a steady, high-rate output stream, closely matching the quantitative improvements observed in TTFT and TPOT benchmarks.

These results reinforce that hardware alone is insufficient—it is the combination of Xeon architecture with OpenVINO's graph optimization, kernel fusion, and low-precision execution that makes multimodal RAG practical and production-ready on CPU platforms.

Future Work

Agentic Long-Context RAG, which is Part 3 of this series of papers, will introduce planning and tool use with validation loops (self-check, re-query, re-rank), add long-context memory, and evaluate end-to-end task success, stability across multi-step trajectories, and cost–latency trade-offs.

System Configuration

The configuration of the server used in our testing is as follows:

- **CPU**: Intel Xeon 6787P processor, 86 cores / 172 threads @ 3.8 GHz
- **GPU**: None
- **Memory**: 16x 64 GB DDR5 @ 6400 GHz
- **Cache**: 336 MB
- **OS**: Ubuntu 22.04.5 LTS (Linux kernel 6.8.0-59-generic)
- **Python**: 3.12.3
- **OpenVINO**: 2025.4.0-20398
- **Transformer**: 4.45.0

Resources

This paper is Part 2 of 3 of a series of papers on Retrieval Augmented Generation.

Part 1: Standard Retrieval Augmented Generation on Intel: From Search to Answers
https://lenovopress.lenovo.com/lp2322-standard-retrieval-augmented-generation-on-intel-from-search-to-answers

For more information, see these resources:

- Lewis et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP.” arXiv:2005.11401
  https://arxiv.org/abs/2005.11401
- MuRAG: Multimodal Retrieval-Augmented Generator for Open Question Answering over Images and Text
  https://arxiv.org/abs/2210.02928
- OpenVINO Toolkit:
  https://www.intel.com/content/www/us/en/developer/tools/openvino-toolkit/overview.html
- OpenVINO GenAI Model:
  https://openvinotoolkit.github.io/openvino.genai/docs/use-cases/image-processing/
- Huggingface OpenVINO Toolkit:
  https://huggingface.co/OpenVINO/collections
- Intel Advanced Matrix Extensions (Intel AMX):
  https://www.intel.com/content/www/us/en/products/docs/accelerator-engines/what-is-intel-amx.html
- Intel Xeon 6787P (Intel Xeon 6 / Granite Rapids) product page:
  https://www.intel.com/content/www/us/en/products/sku/241844/intel-xeon-6787p-processor-336m-cache-2-00-ghz/specifications.html
- OpenVINO Model Server
  https://docs.openvino.ai/2025/model-server/ovms_what_is_openvino_model_server.html
- UX response-time thresholds (≈0.1 s instant / ≈1 s flow / ≈10 s attention): Nielsen Norman Group
  https://www.nngroup.com/articles/response-times-3-important-limits/
- Average adult silent reading speed (200–300 wpm):
  https://en.wikipedia.org/wiki/Words_per_minute

Model cards:

- OpenVINO/Phi-3.5-vision-instruct-int8-ov
  https://huggingface.co/OpenVINO/Phi-3.5-vision-instruct-int8-ov
- OpenVINO/Phi-3.5-vision-instruct-fp16-ov
  https://huggingface.co/OpenVINO/Phi-3.5-vision-instruct-fp16-ov
- microsoft/Phi-3.5-vision-instruct
  https://huggingface.co/microsoft/Phi-3.5-vision-instruct
- OpenVINO/mistral-7b-instruct-v0.1-int8-ov
  https://huggingface.co/OpenVINO/mistral-7b-instruct-v0.1-int8-ov
- OpenVINO/mistral-7b-instruct-v0.1-fp16-ov
  https://huggingface.co/OpenVINO/mistral-7b-instruct-v0.1-fp16-ov
- mistralai/Mistral-7B-Instruct-v0.1
  https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

Author

Kelvin He is an AI Data Scientist at Lenovo. He is a seasoned AI and data science professional specializing in building machine learning frameworks and AI-driven solutions. Kelvin is experienced in leading end-to-end model development, with a focus on turning business challenges into data-driven strategies. He is passionate about AI benchmarks, optimization techniques, and LLM applications, enabling businesses to make informed technology decisions.

Related product families

Product families related to this document are the following:

- Artificial Intelligence
- Intel Alliance

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service. Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION ”AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk. Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document, LP2350, was created or updated on January 12, 2026.

Send us your comments in one of the following ways:

- Use the online Contact us review form found at:
  https://lenovopress.lenovo.com/LP2350
- Send your comments in an e-mail to:
  comments@lenovopress.com

This document is available online at https://lenovopress.lenovo.com/LP2350.

Trademarks

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:
Lenovo®

The following terms are trademarks of other companies:

Intel®, the Intel logo, OpenVINO®, and Xeon® are trademarks of Intel Corporation or its subsidiaries.

Linux® is the trademark of Linus Torvalds in the U.S. and other countries.

Microsoft® is a trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.