References

These are the primary sources for the technical claims in the book. Where multiple sources existed for the same fact, the most authoritative (vendor docs first, then peer-reviewed papers, then independent measurement) was used. Sources marked † are referenced but were not directly accessed at time of writing — treat their specific details as load-bearing-but-unverified and re-check before depending on them.

Organized by source type.

Intel & OpenVINO Primary Sources

OpenVINO Documentation (docs.openvino.ai). The canonical reference for OpenVINO Runtime, OpenVINO GenAI, NPU plugin, and Optimum-Intel. Pages cited throughout the book:

- `about-openvino/compatibility-and-support/supported-operations.html` — the operator coverage matrix per release. Used in Chapter 1.2.
- `openvino-workflow-generative/inference-with-genai-on-npu.html` — the canonical "GenAI on NPU" guide. Source for the INT4-sym / `--ratio 1.0` / group-size quantization rule, the NF4 Lunar-Lake-only constraint, and the `LLMPipeline` NPU property table in Chapters 1.2 and 2.2.
- `openvino-workflow/running-inference/inference-devices-and-modes/npu-device.html` — the NPU plugin reference. Source for `CACHE_DIR`, `MAX_PROMPT_LEN`, `NPUW_LLM_PREFILL_CHUNK_SIZE`, `PREFILL_HINT`, `GENERATE_HINT`, `NPUW_LLM_ENABLE_PREFIX_CACHING`, and `PERFORMANCE_HINT` properties.

OpenVINO Release Notes. Per-version feature deltas:

- **2025.2** — encoder-side QKV projection and MHA graph-level fusions for transformer encoders.
- **2025.3** — chunked prefill on NPU (`PREFILL_HINT=DYNAMIC`, `NPUW_LLM_PREFILL_CHUNK_SIZE=1024`); NF4 + FP16 KV cache on Lunar Lake.
- **2025.4** — 8K context GA on NPU, prefix caching (`NPUW_LLM_ENABLE_PREFIX_CACHING:YES`), multinomial sampling on NPU; memory-mapped cached models.
- **2026.0** — NPU compiler decoupled from OEM driver; speculative decoding on NPU.
- **2026.1** — `TextEmbeddingPipeline` NPU support; current stable as of May 2026.

OpenVINO Model Hub (`huggingface.co/OpenVINO`). Source of the DeepSeek-R1-Distill-Llama-8B INT4 benchmark at **6.10 tok/s on Intel NUC 14 Pro (Lunar Lake)** used as the ITL anchor in Chapter 1.3. Per-model benchmark pages list TTFT, ITL, and target device.

Intel `intel/linux-npu-driver` (github.com/intel/linux-npu-driver). The in-tree Linux driver for Intel NPU. Apache 2.0. Source for the "OS support spans Windows and Linux" claim in Chapter 1.1.

Intel Lunar Lake Launch (Intel Newsroom, September 3, 2024). "Intel Core Ultra Series 2 Processors Deliver Unmatched Power-Efficient AI Performance and x86 Compatibility." Source for the 48 TOPS NPU 4 figure, the LPDDR5X-8533 / 136.5 GB/s spec, and the single-tile compute architecture.

Intel Panther Lake CES 2026 Announcement. Source for the 50 TOPS NPU 5, native FP8 (E4M3/E5M2), programmable LUT for activations, and Intel 18A process claims. Press materials at intel.com/content/www/us/en/newsroom/news/. † Specific NCE count (3 each ~2× wider) is from secondary press coverage and should be re-verified against Intel's formal whitepapers when available.

Intel `openvino-ai-plugins-gimp` 3.2 Release Notes (github.com/intel/openvino-ai-plugins-gimp/releases). Source for the verbatim "FP8 model installation is now gated to NPU5000 and newer architectures" quote in Chapter 1.2.

Intel Community Forums (community.intel.com). Thread 1735991 (February 2026) on `DetectionOutput` NPU/iGPU compile failures; GitHub openvinotoolkit/openvino issue #13594 on `ScatterNDUpdate` rejection. Background for the operator-coverage landmines in Chapter 1.2.

Microsoft & Windows Copilot+ Sources

"Phi Silica, small but mighty on-device SLM" (Windows Experience Blog, December 2024). The canonical reference for Phi Silica architecture: CPU tokenizer + embedding + LM-head, NPU transformer, CPU decode with N=64 KV sliding window. Source for the verbatim "Context processing involves intense parallel computation..." quote in Chapter 1.1 (the closest Microsoft analog to a "decode is bandwidth-bound" statement).

"DeepSeek-R1-Distill on Phi Silica stack" (Windows Developer Blog, 2026). Microsoft's extension of the Phi Silica architecture to a 1.5B and 14B reasoning model. Source for the 1.5B at ~40 tok/s and 14B at ~8 tok/s figures on Snapdragon X NPU in Chapter 2.3. † Specific numbers vary across blog updates — re-check against the canonical post.

Click to Do documentation (learn.microsoft.com/windows/ai/apis/phi-silica). The Phi Silica frontend's prompt templates and single-turn execution model. Background for the single-shot positioning in Chapter 2.3.

Phi Silica Windows Update KBs. KB5079266, KB5084176, KB5089866 — the cumulative updates that progressively rolled Phi Silica out to Intel Copilot+ hardware. † Specific KB numbers are from third-party Windows news aggregators; verify against the Microsoft Update Catalog before quoting.

Hugging Face & Optimum-Intel

Hugging Face Optimum-Intel Documentation (huggingface.co/docs/optimum/intel). Source for the `optimum-cli export openvino` command syntax, the task-name conventions (including the `text2text-generation-with-past` vs. `--task translation` distinction in Chapter 1.2), and the `OVModelForSeq2SeqLM` / `OVModelForCausalLM` class hierarchy.

M2M-100 Model Cards. `facebook/m2m100_418M`, `facebook/m2m100_1.2B`, `facebook/m2m100-12B-avg-5-ckpt`. Source for: model architectures (24 encoder + 24 decoder layers on 1.2B, 16 heads, 64 head_dim), MIT license, the 128,112 vocabulary size, the `forced_bos_token_id` requirement, and the `decoder_start_token_id = eos_token_id = 2` convention. Also: `transformers`'s `modeling_m2m_100.py` source for the no-GQA architectural claim.

NLLB-200 Model Card (`facebook/nllb-200-distilled-600M`, etc.). Source for the CC-BY-NC 4.0 license and the shared `M2M100ForConditionalGeneration` class implementation.

Phi-3-mini-3.8B Model Card (`microsoft/Phi-3-mini-4k-instruct`). Source for the 32 layers / 32 heads / 8 KV heads / GQA architecture used in the KV-cache comparison in Chapter 2.1.

DeepSeek-R1-Distill-Llama-8B Model Card (`deepseek-ai/DeepSeek-R1-Distill-Llama-8B`, and its OpenVINO Model Hub quantized variant). The 8B parameter count and Llama architecture lineage are referenced in Chapters 1.3 and 2.3.

Hugging Face × Intel "Build an Agent with Qwen3-8B on Intel iGPU" Blog (huggingface.co/blog). Closest existing analog to a published multi-step agent on Intel hardware. **Runs on iGPU, not NPU** — used in Chapter 2.3 as the "negative result" reference for the absence of NPU-targeted agent guidance.

Independent Benchmarks & Analysis

MLPerf Client v0.6 (mlcommons.org/benchmarks/client). The industry-standard client-side ML benchmark suite. Intel's Core Ultra Series 1 Meteor Lake numbers (Llama 2 7B: TTFT **1.09 s**, **18.55 tok/s** sustained) used as the TTFT/ITL anchor in Chapter 1.3 come from MLPerf Client v0.6 submitter data.

Chips and Cheese, "Intel Meteor Lake's NPU" (chipsandcheese.com/p/intel-meteor-lakes-npu). The independent measurement of **9.5 TOPS at 1.16 GHz** for NPU 3720, against Intel's marketing claim of ~11.5 TOPS. Source for the "TOPS is the marketing number" framing in Chapter 1.1.

IPEX-LLM Quickstart Documentation. Source for the "30 s to several minutes cold start for 3B–8B LLM INT4 on NPU" anchor in Chapter 1.3.

Markaicode and Audacity OpenVINO Documentation. Source for warm-start LLM load times (<3 s) and the 10–30 s cold / 1–3 s warm range for Whisper/MusicGen/Demucs. † These are secondary developer-blog sources; treat the specific numbers as illustrative ranges, not validated SLAs.

Foundational Papers

Fan, A. et al. (2020). "Beyond English-Centric Multilingual Machine Translation." arXiv:2010.11125. The M2M-100 paper. Background for the architecture and training data choices.

Yao, S. et al. (2022). "ReAct: Synergizing Reasoning and Acting in Language Models." arXiv:2210.03629. The foundational ReAct paper. Background for the reasoning-architecture discussion in Chapter 2.3.

Ainslie, J. et al. (2023). "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints." arXiv:2305.13245. The GQA paper. Background for the MHA-vs-GQA KV-cache comparison in Chapter 2.1.

Williams, S., Waterman, A., Patterson, D. (2009). "Roofline: An Insightful Visual Performance Model for Multicore Architectures." Communications of the ACM 52(4). The original roofline-model paper. Background for the bandwidth-vs-compute analysis in Chapter 1.3.

NLLB Team (2022). "No Language Left Behind: Scaling Human-Centered Machine Translation." arXiv:2207.04672. The NLLB-200 paper. Background for M2M-100's relationship to its successor and the architectural-class continuity.

Standards Bodies & Competitor Vendor Pages

Apple Neural Engine — Apple Developer documentation (developer.apple.com/machine-learning). Source for the M4 family 16-core ANE / 38 TOPS figure. Core ML is the only access path. Background in Chapter 1.1.

Qualcomm Snapdragon X Elite Product Page (qualcomm.com). Source for the 45 TOPS Hexagon NPU figure. Background for the Phi Silica-on-Snapdragon-X numbers (TTFT 230 ms, 20 tok/s) referenced throughout. † Phi Silica's published numbers are on Snapdragon X, not Intel — this is called out explicitly in Chapters 1.3 and 2.3 to prevent cross-platform extrapolation errors.

AMD Ryzen AI 300 / XDNA 2 Documentation (amd.com). Source for XDNA 2's 50 TOPS INT8 + 50 TOPS Block FP16 specs and the separate-IP-block integration model contrast.

Google Edge TPU (coral.ai). Background for the systolic-array architecture comparison in Chapter 1.1.

On Verification and Recency

This book was written in May 2026. NPU silicon, OpenVINO releases, and Phi Silica documentation are all moving targets — features cited as "2025.3+" or "2026.0" will be displaced by newer releases within months of publication. When in doubt, re-check the canonical Intel and Microsoft sources for the current state of:

- The `LLMPipeline` NPU property table (configuration knobs change frequently)
- The validated NPU model list (grows with each release)
- The precision matrix by generation (NF4, FP8 support evolves)
- Phi Silica's deployment surface on Intel hardware (specific KB numbers and rollout coverage)
- Lunar Lake and Panther Lake performance numbers (Intel publishes refreshed datapoints regularly)

The technical reasoning in the book — bandwidth ceilings, encoder/decoder partition, single-shot-vs-ReAct tradeoff — outlasts any specific version. The numbers don't.

Previous: *Glossary*

Revision #1
Created 2026-05-12 19:23:03 UTC by Admin
Updated 2026-05-12 19:23:03 UTC by Admin