Scaling LLMs: Systems, Optimization, and Emerging Paradigms

A research seminar, Fall 2026, Mondays 4:00 to 6:00 PM (Eastern)
Columbia University, Department of Computer Science

Instructor

Dr. Kaoutar El Maghraoui
Adjunct Professor of Computer Science, Columbia University
Principal Research Scientist, IBM T.J. Watson Research Center, NY
Principal Investigator, IBM Research AI Hardware Center
LinkedIn: linkedin.com/in/kaoutar-el-maghraoui | Web: kaoutarelmaghraoui.com

Prerequisites

High Performance Machine Learning (COMS W4995/6998) or equivalent. HPML covers model compression, quantization, knowledge distillation, NAS, parallelism and distributed training, LoRA/QLoRA, FlashAttention, speculative decoding, and LLM serving (vLLM), in PyTorch and CUDA. Because those fundamentals are assumed, this seminar starts at the frontier. Students without HPML should contact the instructor before enrolling.

What this course is about

Large language models stopped being only about making the network bigger. The interesting questions in 2026 are about systems and economics: how you train a frontier model without burning a fortune, how you serve it fast and cheaply, and how it learns to reason and to act instead of just predicting the next token. This seminar follows that whole arc, from scaling laws to reasoning models to the hardware underneath, and it keeps one thread running through everything: hardware and software are designed together, and that is where most of the real gains come from.

We read the papers that actually moved the field, present them, argue about them honestly, and build a semester-long group project that ends in a survey with a real experiment. The reading list leans recent and industrial on purpose. You should leave able to read a systems paper critically, reason about cost and latency, and speak the language used in frontier labs.

What you will be able to do by the end

- Explain modern scaling laws, including what low precision does to the trade-offs, and reason about compute-optimal versus inference-optimal decisions.
- Compare how frontier models are trained at scale: parallelism strategies, memory sharding, and FP8/FP4 recipes.

- Analyze inference systems end to end: paged KV memory, batching, disaggregated serving, speculative decoding, quantization, KV compression, and mixture-of-experts.
- Evaluate the reasoning and agentic paradigms, from test-time compute and RL from verifiable rewards to tool use, the Model Context Protocol, and agent evaluation.
- Weigh hardware futures, comparing digital, analog in-memory, and neuromorphic-inspired approaches through a co-design lens.
- Produce a focused survey with an experimental evaluation, and present it clearly to peers.

How a typical class runs (two hours)

- **Instructor context and trends (about 15 minutes).** I open each week with a short deck that frames the topic, connects it to what industry is shipping, and sets up the papers.
- **Paper presentations (about 60 minutes).** Two core papers are presented, each by a small student team.
- **Critique and discussion (about 30 minutes).** A discussant team leads a critical conversation on each paper: what holds up, what does not, and what they would test next.
- **Optional lab pointer (a few minutes).** I point you to that week's hands-on notebook, which you can run on your own time.

We keep it to two presented papers per week. That was the right pace last time, and it leaves room for real discussion instead of rushing. A third paper appears only when it shares a theme and the same team presents it.

Reading expectations

- **Core papers: required for everyone.** Each week has two core papers. Read them closely enough to discuss the problem, method, evidence, assumptions, and limitations. Core papers are the only papers covered directly by the paper-reading portion of the weekly quiz.
- **Supplementary papers: recommended, not required for the whole class.** These provide historical context, competing approaches, or a path into the current frontier. They are not covered by the weekly paper quiz.
- **Presenter and discussant deep dive.** Teams assigned that week are expected to consult at least one relevant supplementary paper for each core paper they cover. Use it to place the core paper in context, test its assumptions, and identify what changed before or after it.

Context resources such as an industry specification or system documentation may also appear in the schedule. They are clearly labeled and are not treated as research papers.

Presenting and critiquing with a large class

There are about 54 of us, so roles are organized in small teams and everyone does both jobs once. Over the semester the two core papers each week give 24 presentation slots and 24 discussant slots, or 48 team slots total. Each student joins one presenter team and, in a different week, one

discussant team. Teams are pairs by default; twelve of the 48 team slots take a third person so all 54 students can complete both roles. Nobody presents twice, and nobody sits out.

- **Presenters** give a roughly 20 minute talk: the problem, the method, the results, and an honest read of the limitations, plus a short demo or figure walk-through. Use the presenter template.
- **Discussants** give a roughly 10 minute critique and lead discussion, then submit a short write-up. Use the discussant template.

You sign up on the shared paper-signup sheet, first-come, first-serve. The sheet checks that you have exactly one presenter role and one discussant role.

Grading

- **Weekly quizzes (25%).** Short online quiz on the week's two core papers. Released after class Monday, due Wednesday 11:59 PM ET. One or two questions may ask about concepts demonstrated in that week's hands-on notebook; the notebook itself is not submitted.
- **Paper presentation and discussion (35%).** Each student presents one paper in a team and serves on a discussant (critic) team for a different paper. Multiple students present a paper as a team.
- **Final project (40%).** Group survey paper with a real experimental evaluation, built across the semester. Up to 10 bonus points for publish-ready work or a meaningful open-source contribution.

Weekly quizzes

A short online quiz goes out right after class each Monday and is due Wednesday at 11:59 PM ET. It covers the two core papers, plus at most one or two questions about concepts demonstrated in that week's hands-on notebook. Supplementary papers are not tested. The point is not to trick you; it is to make sure everyone comes to the discussion having read the same two papers closely.

Hands-on labs (not submitted)

Every content week has a short, runnable notebook that makes the week's idea concrete: measure a KV cache, watch speculative decoding accept and reject tokens, quantize a small model and see the accuracy move. Labs are not collected and are not graded separately. A quiz may include one or two questions about concepts demonstrated in the notebook, so you should at least review the notebook even if you do not run every cell. Most labs run on a free Colab GPU or even CPU.

Final project

In groups, you will write a survey on a focused topic from the course and back it with a real experiment, not just a literature summary. Every proposal must state a concrete scaling claim or trade-off that the experiment will test: for example, throughput versus batch size, reasoning accuracy versus token budget, context length versus KV footprint, agent success versus tool-call budget, or accuracy versus precision. Proposals are due in Weeks 3 to 4, the work runs in parallel with the seminar, and final presentations are in Week 14 (Dec 14, the last day of classes). Full

guidelines and the proposal template are on the course site. I am happy to connect strong projects with industrial mentors.

Course modules

Module I: Foundations and Architecture (Weeks 2 to 4)

Scaling laws that account for inference cost, distributed training beyond 10,000 GPUs, and architectures moving beyond the uniform dense Transformer: sparse mixture-of-experts, hybrid attention/recurrent designs, and hardware-aligned sparse attention. Since HPML covers the basics, we read the frontier versions.

Module II: Inference Optimization and Economics (Weeks 5 to 8)

Where most production dollars live. We study disaggregated serving and SLO-aware scheduling, the long-context KV-memory wall, INT4/KV4/FP4 co-design, and the changing generation stack from speculative decoding to parallel and diffusion-based decoding. Every week connects the method to goodput, tail latency, memory traffic, and dollars per token.

Module III: Reasoning and Alignment (Weeks 9 to 10)

How models learn to reason and how reasoning changes the systems stack. We study RL post-training and rollout bottlenecks first, then test-time compute: given a reasoning model, how should an inference system spend a token and latency budget adaptively?

Module IV: Agentic AI and Multimodal (Weeks 11 to 12)

LLMs that act and models that combine text, vision, audio, and other components. We study long-horizon agent evaluation, workflow orchestration and resource allocation, agent-computer interfaces, and modern multimodal serving where a request may traverse a dataflow graph rather than one autoregressive decode loop.

Module V: Hardware Futures (Week 13)

Hardware-software co-design through contrasting architectures: GPU clusters, wafer-scale systems, SRAM-based deterministic pipelines, TPUs, memory-near-compute, and specialized inference accelerators. The goal is to understand which bottleneck each architecture moves, not to crown one universal winner.

Module VI: Final Project (Weeks 3 to 14)

Your group survey plus a real experiment, developed across the whole semester and presented on the last day.

Weekly schedule and reading list

Week 1, Sep 14: Course introduction and the 2026 LLM landscape

Instructor-led session. No core paper.

Week 2, Sep 21: Scaling laws after Chinchilla: training, architecture, and inference cost

Core

- Hoffmann et al., *Training Compute-Optimal Large Language Models (Chinchilla)* (NeurIPS 2022)
- Bian et al., *Scaling Laws Meet Model Architecture: Toward Inference-Efficient LLMs* (ICLR 2026)

Supplementary

- Kaplan et al., *Scaling Laws for Neural Language Models* (2020)
- Sardana & Frankle, *Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws* (ICML 2024)
- Kumar et al., *Scaling Laws for Precision* (ICLR 2025)

Week 3, Sep 28: Training at frontier scale: parallelism, heterogeneity, and communication

Core

- Jiang et al., *MegaScale: Scaling Large Language Model Training to More Than 10,000 GPUs* (NSDI 2024)
- Hu et al., *Tessera: A Holistic Pipeline Parallelism Framework for Trillion-Parameter Heterogeneous MoE Training* (OSDI 2026)

Supplementary

- Zhao et al., *PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel* (VLDB 2023)
- Micikevicius et al., *FP8 Formats for Deep Learning* (2022)
- Zhang et al., *COMET: Fine-grained Computation-communication Overlapping for Mixture-of-Experts* (MLSys 2025)

Week 4, Oct 5: Beyond the dense Transformer: sparse MoE and hybrid sequence models

Core

- DeepSeek-AI, *DeepSeek-V3 Technical Report* (2024/2025)
- Zuo et al., *Falcon-H1: A Family of Hybrid-Head Language Models Redefining Efficiency and Performance* (2025)

Supplementary

- Gu & Dao, *Mamba: Linear-Time Sequence Modeling with Selective State Spaces* (COLM/2024)
- Yuan et al., *Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention* (ACL 2025)
- DeepSeek-AI, *DeepSeek-V3.2 Technical Report* (2025)

Week 5, Oct 12: Production serving: disaggregation, scheduling, and SLO-aware goodput

Core

- Qin et al., *Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving* (FAST 2025)
- Zhang et al., *JITServe: SLO-aware LLM Serving with Imprecise Request Information* (NSDI 2026)

Supplementary

- Kwon et al., *Efficient Memory Management for Large Language Model Serving with PagedAttention (vLLM)* (SOSP 2023)
- Zheng et al., *SGLang: Efficient Execution of Structured Language Model Programs* (NeurIPS 2024)
- Zhong et al., *DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving* (OSDI 2024)

Week 6, Oct 19: Long-context systems: sparse attention and hierarchical KV memory

Core

- Yuan et al., *Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention* (ACL 2025)
- Xie et al., *Strata: Hierarchical Context Caching for Long Context Language Model Serving* (OSDI 2026)

Supplementary

- Zhang et al., *H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models* (NeurIPS 2023)
- Xiao et al., *DuoAttention: Efficient Long-Context LLM Inference with Retrieval and Streaming Heads* (ICLR 2025)
- Gao et al., *Rethinking Key-Value Cache Compression Techniques for Large Language Model Serving* (MLSys 2025)

Week 7, Oct 26: Low precision as a systems problem: INT4, KV4, and FP4

Core

- Lin et al., *QServe: W4A8KV4 Quantization and System Co-design for Efficient LLM Serving* (MLSys 2025)
- Egiazarian et al., *Bridging the Gap Between Promise and Performance for Microscaling FP4 Quantization* (ICLR 2026)

Supplementary

- Xiao et al., *SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models* (ICML 2023)
- Lin et al., *AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration* (MLSys 2024)
- Ashkboos et al., *QuaRot: Outlier-Free 4-Bit Inference in Rotated LLMs* (NeurIPS 2024)

Week 8, Nov 2: Breaking the autoregressive bottleneck: speculation and parallel generation

Core

- Liu et al., *Speculative Decoding: Performance or Illusion?* (MLSys 2026)
- Wang et al., *Diffusion LLMs Can Do Faster-Than-AR Inference via Discrete Diffusion Forcing* (ICLR 2026)

Supplementary

- Leviathan et al., *Fast Inference from Transformers via Speculative Decoding* (ICML 2023)
- Li et al., *EAGLE-3: Scaling up Inference Acceleration of Large Language Models via Training-Time Test* (2025)
- Cai et al., *Medusa: Simple LLM Inference Acceleration Framework with Multiple Decoding Heads* (ICML 2024)
- SPEED-Bench team, *SPEED-Bench: A Unified and Diverse Benchmark for Speculative Decoding* (ICML 2026)

Week 9, Nov 9: RL post-training at scale: reasoning, rollouts, and systems bottlenecks

Core

- DeepSeek-AI, *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning* (Nature 2025 / arXiv)
- Gao et al., *RollPacker: Taming Long-Tail Rollouts for RL Post-Training with Tail Batching* (NSDI 2026)

Supplementary

- Yu et al., *DAPO: An Open-Source LLM Reinforcement Learning System at Scale* (NeurIPS 2025)
- Rafailov et al., *Direct Preference Optimization: Your Language Model is Secretly a Reward Model* (NeurIPS 2023)
- Shao et al., *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models* (2024)

Week 10, Nov 16: Test-time compute: adaptive inference budgets and reasoning efficiency

Core

- Snell et al., *Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Model Parameters* (ICLR 2025)
- Lin et al., *Plan and Budget: Effective and Efficient Test-Time Scaling on Reasoning Large Language Models* (ICLR 2026)

Supplementary

- Muennighoff et al., *s1: Simple Test-Time Scaling* (2025)
- Wang et al., *Self-Consistency Improves Chain of Thought Reasoning in Language Models* (ICLR 2023)

Week 11, Nov 23: Agentic systems at scale: long-horizon evaluation and workflow orchestration

Core

- Li et al., *AgencyBench: Benchmarking the Frontiers of Autonomous Agents in 1M-Token Real-World Contexts* (ACL 2026)
- Chaudhry et al., *Murakkab: Resource-Efficient Agentic Workflow Orchestration in Cloud Platforms* (OSDI 2026)

Supplementary

- Li et al., *The Tool Decathlon: Benchmarking Language Agents for Diverse, Realistic, and Long-Horizon Task Execution* (ICLR 2026)
- Yang et al., *SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering* (NeurIPS 2024)
- Yao et al., *ReAct: Synergizing Reasoning and Acting in Language Models* (ICLR 2023)
- Anthropic / MCP, *Model Context Protocol specification and documentation* (Industry specification) [context resource]

Week 12, Nov 30: Multimodal and composite models: long context, modality pipelines, and serving

Core

- Bai et al., *Qwen3-VL Technical Report* (2025)
- Jha et al., *M*: A Modular, Extensible, Serving System for Multimodal Models* (2026)

Supplementary

- Liu et al., *Visual Instruction Tuning (LLaVA)* (NeurIPS 2023)
- Qwen team, *Qwen2.5-VL Technical Report* (2025)
- Beyer et al., *PaliGemma: A Versatile 3B VLM for Transfer* (2024)

Week 13, Dec 7: Hardware-software co-design: wafer scale, SRAM pipelines, dataflow, and memory

Core

- He et al., *WaferLLM: Large Language Model Inference at Wafer Scale* (OSDI 2025)
- Bitar et al., *SHIP: SRAM-Based Huge Inference Pipelines for Fast LLM Serving* (MLSys 2026)

Supplementary

- Abts et al., *Think Fast: A Tensor Streaming Processor (TSP) for Accelerating Deep Learning Workloads* (ISCA 2020)
- Jouppi et al., *TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings* (ISCA 2023)
- Modha et al., *Neural Inference at the Frontier of Energy, Space, and Time (NorthPole)* (Science 2023)
- Alizadeh et al., *LLM in a Flash: Efficient Large Language Model Inference with Limited Memory* (ACL 2024)

Week 14, Dec 14: Final project presentations

Last day of classes.

Policies

- **Academic integrity.** Columbia's rules apply. Present others' work as theirs, cite honestly, and do not misrepresent results. In a research seminar, intellectual honesty is the whole game.
- **Using AI tools.** You may use LLMs to help you read, code, and draft, and I expect you will. Say where you used them in your project and slides, and never present AI-generated text or results as verified when you have not checked them. Getting fooled by your own tools is a real failure mode we will discuss.
- **Attendance and participation.** This is a discussion course, so showing up and engaging matters. If you must miss a week when you present, arrange a swap with another team ahead of time.
- **Late work.** Quizzes close on time because we discuss answers afterward. Talk to me early if something serious comes up; I would much rather hear from you before a deadline than after.
- **Accommodations.** If you are registered with Disability Services, send me your accommodation letter early and we will make it work.

A note from me

I run this course the way I wish research had been taught to me: read the primary sources, argue about them with people who disagree, and build something real. The two core papers give us a shared base each week; the supplementary list is there when you want to go deeper or need context for a presentation or project. The reading list will keep moving as the field moves, and I welcome you proposing papers you think we should cover. Come curious, come skeptical, and let us have a good semester.