ECE 405: Introduction to Large-Scale Artificial Intelligence Systems

Course Syllabus — Spring 2027

Course Information

Meeting Days/Time	Monday, Wednesday & Friday, 10:30 – 11:20 AM (CRN 57997, 3 credits)
Location	Holmes Hall 211
Semester Dates	January 11, 2027 – May 5, 2027 (last day of instruction)
Instructor	Haoyue Tang — Department of Electrical & Computer Engineering
Email	haoyue@hawaii.edu
Office Hours	TBA, POST 205E (and by appointment)
Course Website	`https://loveisbasa.github.io/thy.github.io/ECE405.html`

Course Description

Language models are the core of modern AI systems. This course gives students the technical background to understand, build, and improve them by building one *from scratch*. Following the structure of Stanford CS336 (*Language Modeling from Scratch*) and the previous offering of ECE 405 by Prof. Igor Molybog, students go through the full pipeline of a language model: tokenization, Transformer architecture and training, GPU systems and kernels, distributed training, scaling laws, data collection and filtering, inference, evaluation, and post-training (supervised fine-tuning, preference alignment, and reinforcement learning for reasoning). **Aim:** for students to gain the engineering ability to reproduce results from AI papers. The course is implementation-heavy: four large coding assignments run on GPU clusters.

Learning Objectives

By the end of the course, students will be able to:

- Implement a byte-pair-encoding tokenizer, a Transformer language model, and an optimizer from scratch in PyTorch, and train the model end-to-end.
- Account for the compute (FLOPs) and memory of training and inference, and profile and benchmark model components on GPUs.
- Write a fused GPU kernel in Triton and build data-parallel and memory-efficient (sharded) distributed training.
- Fit and use scaling laws to choose model size and data size under a compute budget.
- Build a pretraining data pipeline from raw web crawl (extraction, quality and safety filtering, deduplication).
- Evaluate language models and apply supervised fine-tuning, preference optimization (DPO), and reinforcement learning with verifiable rewards (e.g., GRPO) to improve them.

Prerequisites

Linear algebra, calculus, probability & statistics, and basic machine learning. A **solid coding background** is required: Python, PyTorch, bash, and git; familiarity with basic systems concepts (e.g., the

memory hierarchy) is helpful. This is a demanding, implementation-heavy course; the assignments involve substantially more code than a typical course.

Course Materials

No required textbook. Lectures and assignments draw on publicly available resources:

- Stanford CS336: Language Modeling from Scratch (Hashimoto & Liang) — `https://cs336.stanford.edu/`
- ECE 405, Spring 2026 (Igor Molybog) — `https://igormolybog.github.io/ece405_sprg2026.html`
- Goodfellow, Bengio & Courville, *Deep Learning* — `https://www.deeplearningbook.org`; Zhang et al., *Dive into Deep Learning* — `https://d2l.ai`

Slides, readings, recordings, and assignment handouts will be posted on the course website and Lamaku (`https://lamaku.hawaii.edu`). Questions and announcements go through the course discussion forum (link on Lamaku).

Computing Resources

Assignments require GPUs. Onboarding instructions will be given in Week 1–2. Debug on CPU or a small GPU first, and use the cluster for training runs and benchmarking. Clusters are shared resources: start early, because queues are long right before deadlines.

Assignments

There are four coding assignments, adapted from the Stanford CS336 handouts. Each consists of code plus a short written report with plots and answers to the handout questions.

A1. Basics. Implement a BPE tokenizer, a Transformer language model, the cross-entropy loss and the AdamW optimizer; train a small language model on TinyStories / OpenWebText and run hyperparameter and architecture ablations.

A2. Systems & Scaling. Profile and benchmark the A1 model; implement FlashAttention-2 in Triton; build distributed data-parallel training with optimizer-state sharding; fit a compute-optimal scaling law.

A3. Data. Turn raw Common Crawl dumps into pretraining data: text extraction, language identification, quality and harmful-content filtering, PII masking, and exact and fuzzy (MinHash) deduplication.

A4. Alignment & Reasoning RL. Evaluate a pretrained model on math reasoning; improve it with supervised fine-tuning, expert iteration, and GRPO; (optional) preference alignment with DPO.

Assignments are due on **Fridays at 11:59 PM HST** unless otherwise noted (see the schedule).

Grading

Following the previous offering of ECE 405, the course grade is based entirely on the four assignments, weighted equally:

$$\textbf{Course grade} = 0.25\,\mathrm{A1} + 0.25\,\mathrm{A2} + 0.25\,\mathrm{A3} + 0.25\,\mathrm{A4}$$

There are no exams. Letter grades follow the UH Mānoa plus/minus grading system:

A+	≥ 97	B+	87–89.99	C+	77–79.99	D+	67–69.99		
A	93–96.99	B	83–86.99	C	73–76.99	D	63–66.99		
A−	90–92.99	B−	80–82.99	C−	70–72.99	D−	60–62.99	F	< 60

These cutoffs are guaranteed; they may be lowered (never raised) at the end of the semester. An Incomplete (I) may be given, per UH Mānoa policy, only when a small part of the course remains unfinished for reasons beyond the student's control.

Course Policies

Late Days Each student has **5 late days** for the semester. Each late day extends a deadline by 24 hours, and at most **3 late days** may be used on any one assignment. Late days are applied automatically, so you do not need to request them. Save them for later in the semester. Once your late days are used up, or beyond 3 days on one assignment, late work is not accepted except for documented emergencies (illness, family emergency, or other cases recognized by UH Mānoa policy). Contact the instructor as early as possible.

Submission Submit all work on Lamaku as specified in each handout (code archive plus PDF write-up). Do not email submissions. You may resubmit until the deadline, and only the last submission is graded. Partial work is better than no submission.

Regrades Regrade requests must be submitted within **3 days** after grades are released, and only for objective grading errors. The whole assignment may be re-graded.

Collaboration Study groups are allowed, but each student must understand and write their own code and write-up, and submit their own assignment. List your study-group members at the top of your submission.

Use of AI Tools Large language models (e.g., ChatGPT, Claude, Gemini) may be used for low-level programming questions (e.g., "how do I call this PyTorch function?") and high-level conceptual questions, but **not to solve the assignment problems directly**. We strongly encourage turning off AI autocomplete (e.g., Copilot, Cursor Tab) in your IDE when working on the assignments. Non-AI autocomplete, such as completing function names, is fine. The point of the course is to build these systems yourself.

Existing Code Do not consult or copy third-party implementations (e.g., GitHub solutions to CS336, Hugging Face model code) unless a handout explicitly allows it. The handouts are designed to be self-contained.

Academic Integrity Students are expected to follow the UH Mānoa Student Conduct Code (`https://manoa.hawaii.edu/studentsuccess/conduct-code/`). Cheating, plagiarism (including submitting code or text produced by others or by AI tools contrary to the policy above), and unauthorized collaboration are academic dishonesty. Violations will result in a zero on the assignment at minimum and may be referred to the Office of Student Conduct.

Accessibility (KOKUA) Students needing accommodations should contact the KOKUA Program (UH Mānoa Office for Students with Disabilities): `https://www.hawaii.edu/kokua/`. Please share your accommodation letter with the instructor early in the semester.

Title IX UH Mānoa is committed to a learning environment free of sex discrimination, harassment, and violence. Information on resources and reporting: `https://manoa.hawaii.edu/titleix/`. Instructors

are generally required to report incidents they learn about.

Well-being If your performance is being affected by circumstances outside class, please talk to the instructor. Free, confidential support is available from the UH Mānoa Counseling and Student Development Center: https://manoa.hawaii.edu/counseling/.

Feedback An anonymous feedback form will remain open throughout the semester (link on Lamaku). Every submission is read. There will also be a formal course evaluation at the end of the semester.

Weekly Schedule

Week	Dates	Topics	Coursework
Part I — Basics			
Week 1	Jan 11 – 15	Course overview & logistics; language modeling; tokenization (byte-pair encoding)	**A1 (Basics) released**; Koa onboarding
Week 2	Jan 18 – 22	PyTorch tensors, `einops`; resource accounting (FLOPs, memory). *Jan 18: MLK Day, no class*	
Week 3	Jan 25 – 29	Transformer architecture & hyperparameters (normalization, positional encodings, activations)	
Week 4	Feb 1 – 5	Attention alternatives; mixture of experts	*Feb 3: last day to drop without W*
Part II — Systems & Scaling			
Week 5	Feb 8 – 12	GPUs & TPUs; memory hierarchy; benchmarking & profiling	**A1 due Fri Feb 12**; **A2 (Systems & Scaling) released**
Week 6	Feb 15 – 19	Kernels & Triton; FlashAttention. *Feb 15: Presidents' Day, no class*	
Week 7	Feb 22 – 26	Parallelism I: data parallelism, ZeRO / FSDP	
Week 8	Mar 1 – 5	Parallelism II: tensor, pipeline, sequence / context parallelism	
Week 9	Mar 8 – 12	Scaling laws (Chinchilla, compute-optimal training, μP)	**A2 due Fri Mar 12**; **A3 (Data) released**
—	*Mar 15 – 19*	***Spring Recess (No Class)***	
Part III — Data			
Week 10	Mar 22 – 26	Data I: sources, Common Crawl, text extraction. *Mar 26: Kūhiō Day / Good Friday, no class*	
Week 11	Mar 29 – Apr 2	Data II: quality filtering, deduplication, data mixing, synthetic data	*Mar 29: last day to withdraw (W)*
Part IV — Post-Training, Inference & Evaluation			
Week 12	Apr 5 – 9	Post-training I: supervised fine-tuning; RLHF; preference optimization (PPO, DPO)	**A3 due Fri Apr 9**; **A4 (Alignment & Reasoning RL) released**
Week 13	Apr 12 – 16	Post-training II: RL with verifiable rewards (GRPO); reasoning models	
Week 14	Apr 19 – 23	Inference: KV caching, quantization, speculative decoding, serving (vLLM)	
Week 15	Apr 26 – 30	Evaluation: benchmarks & metrics; agents and tool use	
Week 16	May 3 – 5	Alignment & multimodality; course wrap-up	**A4 due Wed May 5**

Note: This schedule is a draft and may be adjusted as the semester progresses. There is no final exam; the final examination period (May 10–14) is not used.