CAP6614 -- Efficient AI: Course Project Announcement

University of Central Florida -- College of Engineering and Computer Science Department of Computer Science
Course: CAP6614 -- Efficient AI (Graduate Level)
Semester: Spring 2026
Instructor: Dr. Yuzhang Shang

---

1. Overview

You will work in teams to engage deeply with a recent research paper from the efficient AI literature. Each team will select one paper from a curated list, then **reproduce its key results and/or apply the proposed technique** in a new setting. By the end of the project, you should have a strong working understanding of at least one modern efficiency technique, practical experience with the tools and frameworks used in the field, and a conference-style report suitable for your professional portfolio.

---

2. Team Formation and Paper Selection

Team Size

- **Team size:** 2 to 4 students per team. A team of **3 is recommended** as it provides a good balance between workload distribution and coordination overhead.
- With 107 students enrolled, we expect approximately **40 teams** depending on team sizes.
- Teams may be formed freely. You are encouraged to find teammates with complementary skills (e.g., someone strong in systems and someone strong in ML theory).

Paper Selection

- Each paper may be selected by **at most 2 teams**. This ensures diversity of presentations and avoids excessive overlap.
- If your first-choice paper is already taken by two teams, you will need to select an alternative.
- If you would like to work on a paper **not on the list**, you may propose it to the instructor for approval. The paper must fall within the scope of efficient AI (compression, quantization, efficient inference, efficient training, or on-device deployment). Submit your proposal via email with the paper title, venue, and a brief justification. Approval is not guaranteed, so please have a backup choice from the provided list.

Registration

1. Form your team and agree on your paper selection.
2. Register your team and paper choice on the **shared Google Sheet**:

   > **Team & Paper Registration Sheet**

3. Registration is **first-come, first-served** for paper selection.

Deadlines

Item	**Deadline**
Team formation	**February 22**
Paper selection	**March 1**

Students who have not joined a team by February 22 will be assigned to teams by the instructor.

3. What to Do

Each team selects one paper from the provided list and works on it for the semester. Your project should include some combination of the following:

- **Reproduce** the key results reported in the paper (or a subset, if full reproduction is infeasible with your compute budget)
- **Apply** the proposed technique to a different model, dataset, or task not covered in the original paper
- **Analyze** the method through ablation studies, hyperparameter sensitivity analysis, or comparisons with related methods
- **Benchmark** the method's efficiency metrics (latency, memory, throughput) on your available hardware

You do not need to do all of the above -- choose a scope that is realistic for your team size and compute resources. The key is **depth over breadth**: a thorough study of one technique is far more valuable than a superficial study of many.

For each paper in the companion document, a suggested project angle is provided to help you get started. You are welcome to pursue a different angle if you have a compelling idea.

4. Paper List

The project paper list is provided in a separate companion document.

The list contains **54 papers** organized into **6 categories**:

Category	**Number of Papers**
1. LLM Pruning and Knowledge Distillation	8
2. Quantization	9
3. Efficient LLM Inference and Serving	10
4. Efficient Training and Fine-Tuning	7
5. Efficient Visual Generation	15
6. On-Device and Edge AI	5

Category	Number of Papers
Total	**54**

Each entry includes the paper title, authors, venue, a brief description, and a suggested project angle.

5. Timeline and Deliverables

Timeline

Deadline	Milestone
Feb 22	**Team formation** -- Form your team and register on the Google Sheet.
Mar 1	**Paper selection** -- Finalize your paper choice on the Google Sheet.
March	**Project implementation** -- Implement your chosen method, run experiments, and iterate.
Early April	**Presentations begin** -- 13-minute presentation + 2-minute Q&A per team, in class. The exact schedule will be announced based on the final number of teams.
May 7	**Final report and code due** -- Submit your report and code repository.

Deliverables

1. Final Presentation (Starting Early April)

- **Duration:** 10 minutes + 3 minutes Q&A
- **Format:** Slide presentation, delivered in class
- All team members must participate in the presentation
- A live demo is encouraged but not required
- **Content should include:**
  - Motivation and background
  - Method overview
  - Experimental setup and results
  - Key findings and insights
  - Limitations and potential future work

2. Final Report (Due: May 7)

- **Length:** 4--6 pages in NeurIPS format (excluding references and appendix)
- **Structure:** Abstract, Introduction, Related Work, Method, Experiments, Analysis/Discussion, Conclusion
- Must include a **contribution statement** listing each team member's specific contributions
- Supplementary material (additional figures, tables, extended results) may be included in an appendix

3. Code Repository (Due: May 7)

- Hosted on **GitHub** (public or private; if private, add the instructor as a collaborator)
- Must include:

  - A clear **README** with setup instructions, dependencies, and how to reproduce results
  - All source code used for experiments
  - Scripts or notebooks to regenerate key figures and tables in the report
  - An environment file (e.g., `requirements.txt` or `environment.yml`)

---

6. Grading Rubric

The project is worth **100 points total**, split equally between the presentation and the code/report.

Presentation (50 points)

Criterion	Points	Description
Clarity and organization	15	Logical flow, clear slides, appropriate level of detail, effective use of figures and diagrams
Technical depth	15	Demonstrates thorough understanding of the method, correct explanation of key concepts, awareness of related work
Demo and results quality	10	Compelling experimental results, clear visualizations, fair comparisons, quantitative evidence
Q&A handling	10	Thoughtful and accurate responses to audience questions, demonstrates deep understanding beyond the slides

Code and Report (50 points)

Criterion	Points	Description
Technical correctness	15	Correct implementation of methods, valid experimental methodology, no significant errors in analysis
Experimental rigor	10	Proper baselines, fair comparisons, error bars or confidence intervals where appropriate, clearly specified experimental conditions
Analysis and insights	10	Thoughtful interpretation of results, discussion of why methods succeed or fail, ablation studies, practical takeaways
Report writing quality	10	Clear and concise writing, well-structured paper, professional figures and tables, proper citations
Code quality and reproducibility	5	Clean, well-documented code; results can be reproduced by following the README; good use of version control

Grade Interpretation

Score Range	Interpretation
90--100	Excellent work with insightful analysis and strong technical execution
80--89	Solid work with correct implementation and reasonable analysis

Score Range	Interpretation
70--79	Acceptable work with some gaps in implementation or analysis
Below 70	Significant issues in technical correctness, completeness, or presentation

7. Compute Resources

Projects in this course vary in their compute requirements, from approximately 5 GPU-hours (deployment benchmarking) to 100+ GPU-hours (large-scale distillation or training). You should choose a project whose compute needs match the resources available to you, and you must state your compute plan in the proposal.

Available Resources

Resource	Details
UCF ARCC / STOKES cluster	If a class allocation is available, details will be announced separately. Check with the instructor for current availability.
Google Colab Pro / Pro+	Provides access to T4, V100, and A100 GPUs. Colab Pro+ (~$50/month) offers longer runtimes and higher-tier GPU access. Suitable for light to medium projects.
Personal hardware	Consumer GPUs (RTX 3090, 4090, etc.) and Apple Silicon Macs are suitable for many projects, especially those involving smaller models or on-device deployment.

GPU Cloud Providers

If you need dedicated GPU access beyond what Colab or personal hardware can provide, the following cloud GPU providers offer on-demand rentals. Pricing is approximate and may vary; check each provider's website for current rates.

Provider	GPU	Approximate Price	Notes
RunPod	A100 80GB	~$1.99/hr	Also offers A100 40GB at lower cost. Community Cloud pricing may be cheaper.
RunPod	H100 80GB	~$2.99/hr	Available in Secure Cloud.
Lambda Labs	A100 80GB	~$2.49/hr	On-demand instances. Also offers 8×A100 nodes.
Lambda Labs	H100 80GB	~$3.29/hr	On-demand instances.
Paperspace	A100 80GB	~$2.24/hr	Gradient platform with Jupyter-style notebooks.

Provider	GPU	Approximate Price	Notes
Together AI	8×A100 80GB	~$2.85/hr per GPU	Minimum 8-GPU cluster rental. Best for large-scale experiments requiring multi-GPU.

> **Tip:** Most providers charge per-second or per-minute, so you can spin up an instance, run your experiment, and shut it down to minimize cost. Budget $50--$150 for a medium-scope project.

Compute Planning Guidance

- **Light projects (5--20 GPU-hours):** On-device deployment, inference benchmarking, quantization of smaller models. Can be done with Colab Pro or a consumer GPU.
- **Medium projects (20--50 GPU-hours):** Most quantization comparisons, LoRA/PEFT studies, diffusion model experiments with pretrained checkpoints. Budget ~$50--$100 on cloud GPUs if needed.
- **Heavy projects (50--100+ GPU-hours):** Large-scale distillation, training diffusion models, experiments requiring multi-GPU setups. Budget ~$100--$300. Requires dedicated GPU access (cluster or cloud).

---

8. Academic Integrity

The University of Central Florida's academic integrity policies apply in full to this project. In addition, the following course-specific guidelines must be observed:

Code and Implementation

- All code must be **original or properly attributed**. Using public implementations as a starting point is acceptable and even encouraged, but you must clearly cite the source in your code comments, README, and report.
- If you build upon an existing codebase (e.g., an author's official repository), explicitly state which parts are original to you and which are from the source.
- You may use standard libraries and frameworks (PyTorch, HuggingFace Transformers, etc.) without special attribution, as they are considered common tools.

Data and Models

- Report all pretrained models, datasets, and external resources used in your project.
- If you use model outputs from commercial APIs (e.g., for generating training data or baselines), disclose this.

Collaboration Between Teams

- Teams must work independently. Sharing code, experimental results, or report text between teams is not permitted.
- General discussions about course concepts, debugging strategies, or resource recommendations are acceptable.

Use of AI Assistants

- The use of AI coding assistants (e.g., GitHub Copilot, ChatGPT) is permitted as a productivity tool, but you must disclose their use in your report.
- You remain fully responsible for understanding and being able to explain every line of code and every claim in your report.

Contribution Statement

- Every report must include a **contribution statement** that lists each team member's specific contributions to the project. Example: "Student A implemented the quantization pipeline and ran experiments. Student B wrote the evaluation scripts and produced figures. Student C drafted the report and conducted the literature review."
- All team members are expected to contribute meaningfully. If issues arise regarding unequal contribution, contact the instructor promptly.

Consequences

Violations of academic integrity, including uncited use of others' code, fabrication of experimental results, or submitting work produced by another team, will be handled in accordance with UCF's academic integrity procedures and may result in a failing grade for the project or the course.