10_LLM Compression Trade-off Evaluator-Robins AFB

CS4850, 03, Fall, 2025, 8/28/2025, Aldi Susanto

Project Team

Roles	Name	Role	Contact (Cell Phone)
Project owner	Ryan Tran	Developer/Documentation	678-670-9868
Team leader	Aldi Susanto	Developer/Documentation	404-834-9304
Team members	Pranav Kartha	Developer	943 268 4909
	Thomas Graddy	Documentation	229-661-5041
Advisor / Instructor	Sharon Perry	Facilitate project progress; advise on project planning and management.	770-329-3895

Ryan Tran

Aldi Susanto

Pranav Kartha

Thomas Graddy

1.0 Project Overview / Abstract

This project studies how model compression affects the speed–accuracy trade-off for large language models (LLMs) on code-generation and debugging tasks, and delivers a prototype that dynamically routes queries between a full model and one or more compressed variants. We will establish a baseline inference environment and implement multiple compression methods (quantization, pruning, and knowledge distillation). Using a suite of code-centric prompts, we will measure task quality with BLEU/Code-BLEU alongside latency, throughput, and memory/cost metrics. Building on these results, we will design routing logic that estimates query complexity and selects the smallest model that preserves target quality, thereby improving efficiency while constraining degradation on difficult inputs. Experiments will compare static single-model deployments to our dynamic allocation approach, highlighting Pareto frontiers across quality and performance. A reporting dashboard will visualize trends in accuracy, latency, resource usage, and cost savings for different compression/routing configurations. The project culminates in end-to-end documentation, including data analysis scripts, evaluation harnesses, and system architecture, to support reproducibility and cloud deployment. Sponsor-provided technical guidance will inform benchmarks, compression settings, and routing criteria, with the overarching goal of practical, scalable LLM inference for code-related workloads.

Deliverables

· LLM compression benchmarking scripts.

· Dynamic query routing proof-of-concept application.

· Code-BLEU metric evaluation reports.

· Interactive dashboard with performance visualizations.

· Final project report with deployment recommendations

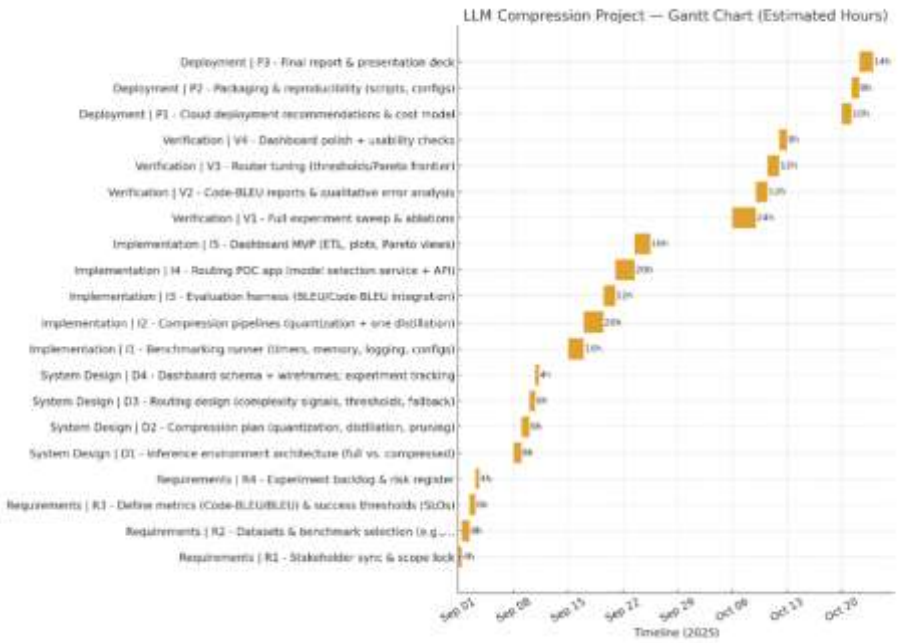
Group Meeting Schedule Date/Time

Tuesdays: 6:00 – 6:15

Collaboration and Communication Plan

Discord/In person Tuesdays and Thursdays 3:30-4:30

Project Schedule and Task Planning



Version Control Plan

We'll use a trunk-based, monorepo workflow for llm-compression, keeping main deployable and doing short-lived branches (feat/…, fix/…, exp/…, docs/…) that are rebased on main and merged via squash. Commits follow Conventional Commits and reference experiment IDs when configs affect results; software is tagged with SemVer (e.g., v0.2.0) while experiment bundles may use CalVer (e.g., 2025.10-exp42). Every release tag carries artifacts (bench CSV/JSON, Code-BLEU summaries, Docker digests) and a generated CHANGELOG; hotfixes branch from the tagged release and merge back to main. CODEOWNERS gate critical paths and PRs require at least one approval plus green checks: lint/format, types, tests, security scan, a DVC pull check for data/model pointers, a tiny bench/Code-BLEU smoke run, and Docker builds for the router/dashboard. Data and model artifacts are tracked with DVC (remote storage); experiments are logged to MLflow/W&B with commit SHA, config hash, dataset/model versions, and metrics (Code-BLEU, latency, memory). Notebooks are kept reproducible via Jupytext and nbstripout; each component ships a Dockerfile with locked deps and simple make/just targets. Secrets never live in Git, use environment/secret managers and .env.example for local. Branch protection on main forbids force-pushes and requires reviews and status checks; releases are cut biweekly (weekly during verification) with rollback to prior tags if needed. Contributor hygiene lives in README, CONTRIBUTING, PR/issue templates, and lightweight ADRs so cross-team changes remain auditable and fast.