# Proposal

## Parallelizing a Simplified Transformer Forward Pass in C++ using OpenMP

### 1. Motivation

Transformer architectures lie at the heart of modern large language models (LLMs) like GPT, BERT, and T5. While these models are often implemented using high-level deep learning frameworks (e.g., PyTorch, TensorFlow), their core components—especially multi-head self-attention and feed-forward layers—are ultimately composed of dense linear algebra operations that are highly parallelizable.

This project aims to **re-implement the core forward pass of a simplified Transformer encoder in C++**, and apply **OpenMP-based parallelization** to accelerate critical components such as matrix multiplication, softmax computation, and multi-head attention.

This hands-on approach will not only strengthen my understanding of how parallelism supports modern deep learning, but also provide a practical application of course material in a real-world, high-impact setting.

### 2. Objectives

- Implement a simplified Transformer Encoder forward pass in C++, including:
    - Multi-head self-attention layer
    - Feed-forward network (FFN)
    - Residual connections
    - Simplified normalization
- Compare the performance of:
    - Baseline **serial** implementation
    - Optimized **OpenMP-parallelized** version
- Measure and analyze:

- Execution time
- Speedup ratio
- Thread scaling behavior
- Numerical correctness and consistency

---

## 3. Tools and Environment

- **Language**: C++17
- **Parallelization**: OpenMP 4.5+
- **Compiler**: g++ with `fopenmp` flag
- **Platform**: Single-node CPU (8+ logical cores)
- **Data**: Randomly generated matrices (simulating embeddings)
- **Visualization**: Python (for plotting results from CSV logs)

---

## 4. Evaluation Metrics

- **Execution Time** (ms per forward pass)
- **Speedup**: Serial vs. parallel (varied thread count)
- **Scalability**: Varying sequence lengths and embedding dimensions
- **Correctness**: Output deviation within numerical tolerance

---

## 5. Timeline

| Week | Task |
| --- | --- |
| Week 1 | Implement matrix operations (matmul, softmax) serial version |
| Week 2 | Build full forward pass for attention and FFN |
| Week 3 | Introduce OpenMP to key modules, begin benchmarking |
| Week 4 | Finalize experiments, conduct analysis |
| Week 5 | Report writing and submission |