

# Week 5 Assignment

Triton: Modern GPU Programming for Researchers

GPU Programming using CUDA and Triton (WiDS'25)

Abhineet Agarwal

Dept. of Electrical Engineering, IIT Bombay

## Overview

In Weeks 2–4, you implemented GPU kernels directly in CUDA, managing threads, memory, and synchronization explicitly.

In this assignment, you will use **Triton**, a Python-first GPU programming language, to reimplement a real kernel you previously wrote in CUDA. The goal is to understand the **trade-off between developer productivity and low-level control**, while still reasoning carefully about performance.

This assignment is intentionally comparative: CUDA vs Triton.

## Submission Instructions

Submit a single PDF summarizing your work, along with source files. Your submission folder should look like:

```
week5/  
  assignment.pdf  
  triton_kernel.py  
  correctness_check.py
```

## 1 Task 1: Reimplement a CUDA Kernel in Triton

Choose **one** kernel you implemented in Week 4:

- Softmax (numerically stable)
- Dense matrix multiplication (GEMM)
- A custom PyTorch operation

Reimplement the kernel using Triton.

### Requirements

- The Triton kernel must be functionally equivalent to the CUDA version
- The implementation must use Triton's programming model
- Correctness must be verified against a CPU or PyTorch reference

Place your implementation in `triton_kernel.py`.

**Note:** You are not required to match CUDA performance exactly.

## 2 Task 2: Benchmarking and Performance Comparison

Benchmark the following implementations:

- Optimized CUDA kernel from Week 3/4
- Triton kernel

### Benchmarking Requirements

- Use the same input sizes for both implementations
- Run multiple iterations and report average runtime
- Clearly state GPU model and environment

### Analysis

In your PDF, include:

- Runtime table or plot
- Speedup or slowdown relative to CUDA
- Brief explanation of observed performance differences

### 3 Task 3: CUDA vs Triton — Design Reflection

Answer the following questions concisely (2–4 sentences each):

- Which parts of the kernel were easier to express in Triton?
- What low-level control did you lose compared to CUDA?
- For your chosen kernel, would you prefer CUDA or Triton in a research setting? Why?

## Analysis Guidelines

Your PDF should prioritize:

- Correctness verification
- Clear benchmarking methodology
- Honest discussion of trade-offs
- Correct technical terminology

## Notes

- You may use Google Colab or a local GPU
- You may reference Triton documentation and examples
- You may reference your own Week 3/4 CUDA code
- All submitted code must be your own

## End of Week 5 Assignment