# AC 215 Statement of Work

## Title & Authors

**Title**: Fine-tuning LLM for Animation Engine
**Authors**: Yan Kaled, Tom Zhang, Tadhg Looram, Mina Lee, Jason Xiang, Nishtha Sardana & Kareema Batool

## Background & Motivation

The team lead is currently spearheading the development of an ed-tech startup and would like to integrate an LLM into a web application. Our team is composed of dedicated data scientists with a profound interest in NLP and LLM technologies. The primary objective of this project is to enhance a specific feature within our ed-tech application, HiSolver. Further insights and motivations for this endeavor are delineated below.

3Blue1Brown is a YouTube channel held in high regard within the mathematics community for its exceptional animations that depict mathematical models. The success of this channel can be largely attributed to these animations, which elucidate complex concepts with intuitive visuals. The creator of the channel pioneered an animation engine that, over time, has seen significant improvements thanks to contributions from the community. The engine is accessible via the Python package Manim. However, interfacing with this engine remains a challenge for most users due to its requirement for specialized expertise.

Given the animation engine's potential juxtaposed with its complexity, our project's inspiration is clear. We aim to develop an AI system that provides a seamless interface to this engine. This AI tool will eliminate the need for coding, enabling users to engage solely through natural language, facilitated by the LLM.

## Problem Statement

The goal is to develop an AI application powered by an LLM that can receive user input in the form of raw text consisting of mathematical problems at the SAT level. The application should provide step-by-step solutions or hints for the student - as well as generate Python code for the animation engine. One major challenge would be to construct and fine-tune the said LLM.

## Minimum Components for a Good Project

- **Large Data**: Our data consists of Python code involving the Manim package. We will perform collection and preprocessing of open-source codebases containing Manim code. Data cleaning would also be a major part of this step: the goal is to produce semi-automated production of cohesive Manim code starting from collected codebases (this allows us to capture nuances in different animations originating from the same base case).
- **Scalability**: Serving multiple user queries to the text-to-manim AI and multiple concurrent animation rendering.
- **Complex Models**: fine-tuning large language models to perform text-to-manim conversion such that the resulting code is cohesive and accurate.
- **Computationally Expensive Inference**: optimization of context identification, code generation, and animation rendering.

## Objectives

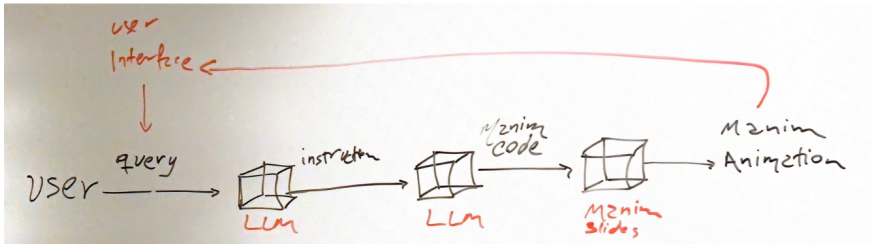
1. Collect and preprocess a large dataset of open-source Manim code relevant to the scope of this project (SAT level math questions).
    a. Preprocessing might require updating the code to the latest versions of Manim.
    b. The data for training consists of natural language instructions as well as the resulting Manim code.
    c. Between the user input and the LLM that will generate the Manim code there will be an intermediary LLM that will transform the user input into proper instructions. The UI will also enforce a formatting of the input, but the intermediate LLM will facilitate the code generation step.
2. Fine-tune an LLM that generates Manim code given an instruction input.
3. Deploy the LLM as a REST API.
4. Implement an interface on the current platform (HiSolver) that receives user input and presents output in an intuitive manner.
5. Integrate LLM with the current platform (HiSolver)

## Learning Emphasis

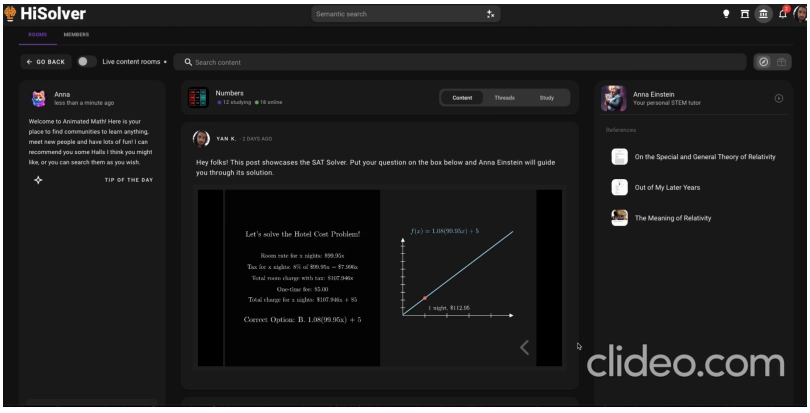
The project will center its efforts on the refinement of a Language Model (LLM) through a fine-tuning process and subsequently integrating this enhanced LLM into a web application. The initial component of this endeavor leverages our existing proficiency, while the latter facet is slated for comprehensive coverage within the context of our ongoing course.

## Application Mock Design

This will be the application interface structure from user input to response:




Prototype of the user interface (presentation of the animation as interactive slides):



## Data

- Data source: Github public repositories and YouTube videos transcriptions. We will supplement the data by generating variations of animations.
- Data attributes: Our data will consist of user prompts, video narrations and Manim code snippets.
- Description: the dataset for training will consist of natural language instructions and Manim code (the pre-processed data will consist of user queries and video narrations).
- Key Attributes: the dataset should include diverse animations (within the boundaries of the complexity level of this project, i.e. SAT questions) that cover specific plotting, algebraic and geometrical manipulations, as well as longer animations (though still short) that illustrate composition of shorter animations. We foresee an upper limit of about 1 minute for any animation generated.
- Data relevance: each component of the data will be vital for fine-tuning the LLM. User queries will be simulated with automatically generated questions using both code and SAT questions, and video narrations (which usually accompany most Manim videos) will be further used to steer the model towards cohesive animation code.
- Data quality: We anticipate our data to have a diverse set of quality issues, ranging from poor code organization to code using outdated versions of Manim (which has poor backwards compatibility).

## Research and Development

We will review literature and open-source projects related to fine-tuning LLM's, as well as documentation of LLM providers (especially OpenAI) and Manim.

## Fun Factor

As data nerds, we love anything that has to do with LLM's, especially when it generates math animations. Plus, the work is for a really good cause.

## Limitations and Risks

- Possible challenges in obtaining a large and diverse enough dataset.
- Computational limitations when deploying complex models.

## Milestones

- Data collection and preprocessing: Sep 26
- LLM Development and fine tuning: Oct 5
- Backend API development and integration: Oct 19
- Frontend integration with HiSolver: Oct 26
- System Testing and Optimization: Nov 12
- Final Deployment and Launch: Nov 30