Computer Vision: Preliminary Proposal

Fall 25

Neural Game Engine: Real-Time Synthesis of Pong/Tetris Through Diffusion Models

Team information (name, members, size)

Claire Chen, Naria Rush, Edward Zhou

Type of project: Application/Demo, Research, To be Decided, or Other Abstract (Brief Description of the project)

In October of 2024, Decart and Etched released an AI fast transformer inference model named OASIS which simulated Minecraft gameplay without running any type of game loop. With this model, players send keyboard and mouse inputs and are met with computer generated frames of real time updates like: running, jumping, looking around a scene, placing blocks, and interacting with mobs which are conditioned on the different inputs given by the user. Their architecture consisted of a two part system, a spatial autoencoder based on ViT and a latent diffusion backbone based on DiT, and was trained on millions of hours of minecraft gameplay. For this project, we would model a similar architecture but for a simpler game like Pong or Tetris. By the end of the semester, our end goal with this project would be to create a playable demo with higher resolution or frame rate that runs a similar version of pong or tetris without running any external game engine or software loop.

Motivation/Problem Statement (What problem or question does your project address?)

Unlike in classic image generation or new text to image, text to video, image to video methods, generating images based on conditioned inputs in real time is still a novel and relatively unexplored area. Based on the information we found online, most existing works produce results with low resolution and low frame rates, which prevents this technology from being applied to many modern use cases where high visual quality is required. Instead, it can only be used in games that do not emphasize graphics or responsiveness. Additionally, OASIS requires millions of hours of gameplay data for training, which represents a massive investment of both resources and time.

In this project, we aim to overcome these challenges by improving both resolution and frame rate, as well as exploring whether training time can be reduced while still achieving comparable performance. Ultimately, our goal is to make this technology applicable to all types of games, thereby enabling the realization of truly real-time AI-generated interactive worlds. Creating a flashy high resolution low complexity game could also inspire future researchers to join in on the challenge and recreate as many games as possible.

Methodology (Key methods you propose to use, models, techniques, datasets, evaluation metrics, tools)

To build our data, we will run actual rounds of pong or tetris within our machines as we test our model. One way to do this could be through Gymmasium: https://ale.farama.org/environments/pong/, which provides a headless pong environment specifically for automated learning. This same site also features simulations for Tetris. We will configure our input into two parts: the ground truth frame which we begin at in runtime, and the game action associated with that point in time. Rather than generate frames based on actual human actions and expected human behavior, we will choose a comprehensive approach where we can parallelize different runtimes with each possible action at that frame. For instance, in pong, a player can always attempt to move their paddle up and down. As such, we will generate two gameplay paths and conditions on each separately.

After generating our frames and their actions, we would follow the current SOTA for generating conditioned frames based off of real time input. This is likely to be a two part system using a ViT and DiT. We will process our input frames using the ViT and then combine them with the current action to feed into the DiT. Afterwards, we will decode to get our next frames and either continue training based off of the highest probability frame or continue our branching method as before. We will continue this process and employ different segmentation techniques or noise functions to see if this increases picture resolution and environment accuracy.

In order to test our project, we would first generate varying lengths of clips of our game at various starting points. From a human standpoint, we can test how effective our simulation tricks the human mind by comparing a small group of human testers and their perception of whether or not they are looking at the real game to evaluate by random guessing. From a technical standpoint, because we plan on further developing the SOTA two part model system consisting of a ViT and DiT, we will continue to evaluate on PSNR to determine how closely our generated frames model what that action would render in the actual game engine loop. We will also use LPIPS to evaluate similarity from a human perspective. Finally, because our generated gameplay footage is the same as generating videos in real time based on keyboard actions, we will use FVD in order to determine how similar our overall videos are to actual gameplay recordings.

Datasets (public datasets, collected, or synthetic, that you propose to use)

https://github.com/etched-ai/open-oasis/tree/master/sample_data

https://www.kaggle.com/datasets/shankar3234/ping-pong-game-playing-dataset

https://www.kaggle.com/datasets/pratikdate123/dataset-for-playing-ping-pong-game

https://www.kaggle.com/datasets/conlan/tetris-training-set-9262023

Project Plan (Proposal Milestones & Timeline)

Week 5 (10/2~10/9): Look up for the datasets

Week 6~8 (10/10~10/23): Finish the code

Week 9~12 (10/24~11/20): Start training and fix any bugs

Week 13~14 (11/21~12/4): Finish the report and prepare for the presentation

Expected Outcomes (e.g. system, demo, benchmark results)

By the end of this project, we aim to develop a playable demo of Pong or Tetris that operates entirely through neural inference without any traditional game engine. Our primary technical goal is to achieve better FPS, better resolution, and less hallucinations. The ideal situation would be indistinguishable from the actual game. We will set up an interactive demonstration station where classmates can directly play our game on our computer using standard keyboard or controller inputs, providing an immediate, hands-on experience of the technology.

Optional questions/file submission

Pong Simulator

Links: OASIS summary, OASIS Github, Similar Method, Another Example (World Models)