Computer Vision

Peking University, 2025 Fall

🦾 Final Project

Important Dates

	Weight	Due Date
Project Milestone	10 %	2026.01.03
Final Report	90 %	2026.01.18

Overview

The Course Project is an opportunity for you to apply what you have learned in class to a problem of your interest. **Potential projects usually fall into these two tracks:**

- **Reconstructing semantics from images.** Image classification, object detection, semantic segmentation, image generation, etc.
- **Reconstructing 3D from images.** You can choose to implement traditional algorithms, deep learning algorithms, or a combination of both to achieve the goal, such as NeRF (Neural Radiance Field), SLAM, Stereo, etc.

We provide some topics in in the following text, you can choose one from them. **You can also explore other topics on the above two tracks, instead of choosing those we provide.** We encourage in-group discussion and finishing tasks independently within the group, but if you need **necessary** help, feel free to contact us. Hope you enjoy the final project!

To inspire ideas, you might also look at recent deep learning publications from top-tier conferences, as well as other resources below.

- CVPR: IEEE Conference on Computer Vision and Pattern Recognition
- ICCV: International Conference on Computer Vision
- ECCV: European Conference on Computer Vision
- NIPS: Neural Information Processing Systems
- ICLR: International Conference on Learning Representations
- Kaggle challenges: An online machine learning competition website.

You might also gain inspiration by taking a look at some past Projects of Stanford cs231n & cs229.

- Stanford University CS231n
- Stanford University CS229

Selected Project

DragGAN

Field: ***Image Editing*** **Difficulty:** ⭐ ⭐ ⭐ **Codebase:** 😊 **GPU:** ⭐

Interactive Image Manifold Manipulation via DragGAN

Overview

In the realm of Deep Generative Models, Generative Adversarial Networks (GANs) have achieved unprecedented success in synthesizing photorealistic images. However, precise controllability over the synthesized content remains a significant challenge. While text-guided synthesis is popular, it lacks spatial precision.

This project focuses on **DragGAN(SIGGRAPH 2023)**, a method that allows users to interactively "drag" specific points in an image to target locations. You will explore how to manipulate the generative image manifold by optimizing latent codes and tracking feature points without relying on external 3D priors or manual annotations.

Motivation

The primary goal of this project is to understand the mechanics of **Latent Space Manipulation**.

- **The Problem:** Traditional image editing (like warping) often distorts object structure. Existing GAN control methods usually rely on rigid 3D models or limited supervised learning.
- **The Solution:** DragGAN proposes that the feature space of a pre-trained GAN is sufficiently discriminative to enable both motion supervision and precise point tracking. By solving this, we achieve flexible, precise, and generic controllability across categories like animals, cars, and humans.

Tasks

Part A: Paper Reproduction (40%)

You are required to reproduce the official implementation of DragGAN or write your own based on the paper.

1. Setup & Execution
   - Clone the official DragGAN repository, configure the environment and successfully run the interactive GUI or inference scripts.
2. Code Understanding:
   - Locate and analyze the specific code segments responsible for the core mechanisms described in the paper:
     - **Motion Supervision:** Identify where the shifted patch loss is calculated and how the stop gradient option is implemented.
     - **Point Tracking:** Identify the function that performs the nearest neighbor search within the feature patch to update handle points.
3. Reproduction
   - Reproduce the Qualitative results in this paper as many as possible.
4. What to report:
   - An analysis report of official implementation codes.
   - An image or video demo.

Part B: Innovation & Extension (60%)

Choose one or more of the following directions to extend the project. You must conduct experiments to validate your extension.

Option 1: Tracking Robustness

- **Context:** The paper notes that handle points in texture-less regions sometimes suffer from drift during tracking.
- Task:

  Improve the tracking mechanism.

  - *Idea:* Replace the simple nearest neighbor search with an optical flow estimator (like RAFT) or a particle video tracker (like PIPs), or propose a hybrid feature fusion strategy.
  - *Comparison:* Compare your improved tracker against the baseline DragGAN tracking in challenging, low-texture scenarios.

Option 2: Masking and Background Preservation

- **Context:** DragGAN allows users to draw a binary mask to keep certain regions fixed.
- Task:

  Analyze and improve the masking performance.

  - *Idea:* The current method uses a reconstruction loss on the unmasked region. Experiment with blending features from the original image into the generated image to better preserve background details, similar to techniques mentioned in the discussion.

Option 3: DragGAN vs. DragDiffusion

- **Context:** While DragGAN achieves precise control, it is limited by the **domain-specificity** of the underlying GAN (e.g., a model trained on faces cannot edit cars). Furthermore, DragGAN can suffer from distortions when editing images towards **out-of-distribution** poses that the GAN never saw during training.
- **The Alternative: DragDiffusion** adapts the interactive point-based editing framework to Diffusion Models (e.g., Stable Diffusion), aiming to solve the generality issue.
- Task:

  1. **Deploy DragDiffusion:** Set up and run the official implementation of *DragDiffusion*.
  2. Comparative Analysis:

     Compare between DragGAN and DragDiffusion on two types of inputs:

     - *In-Domain:* An image that fits DragGAN's training set perfectly (e.g., a human face or a cat). Compare the precision and speed.
     - *Open-Domain:* An arbitrary image (e.g., a sculpture, a sketch, or a fictional creature) that DragGAN was not explicitly trained on.
  3. Problem Solving Analysis:

     Specifically analyze and report on:

     - **Generality:** Does DragDiffusion successfully edit images where DragGAN fails due to lack of category-specific training?
     - **Quality:** Compare the background preservation. DragGAN sometimes struggles with complex backgrounds or texture-less regions. Does the Diffusion model handle these better?

Submission Requirements

1. **Codebase:** A link to your GitHub repository with a `README` explaining how to run your edits.
2. Project Report (PDF):
   - **Introduction:** Brief summary of DragGAN.
   - **Methodology:** Details of your specific implementation and your chosen *Innovation*.
   - **Experiments:** Visual comparisons (Before/After), loss curves, and quantitative metrics (MD/FID).
   - **Discussion:** Analysis of why your extension worked or failed.
3. **Demo Video(Optional):** A short video (<1 min) showing the interactive dragging process.

Reference

- **Paper:** Pan et al., "Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold", SIGGRAPH 2023.
- **Project Page:** https://vcai.mpi-inf.mpg.de/projects/DragGAN/

Collaboration

You can work in teams of up to **3** people. We do expect that projects done with 3 people have more impressive writeups and results than projects done with 2 people or 1 person.

Late Policy

3 points will be deducted for each day of late submission.

Project Milestone

Your project milestone report should be between 1 - 2 pages. The following is a suggested structure for your report:

- Title, Author(s)
- Introduction: This section introduces your problem and the overall plan for approaching your problem.
- Problem statement: Describe your problem precisely, specifying the dataset to be used, expected results, and evaluation.
- Technical Approach: Describe the methods you intend to apply to solve the given problem.
- Intermediate/Preliminary Results: State and evaluate your results up to the milestone.

Submission: Please submit your milestone **in English or Chinese** as a PDF.

Final **Report**

Your final write-up is required to use the provided template with **no more than 6 pages. Chinese or English** are allowed to finish the report. You can apply command “\usepackage{ctex}” and choose the XeLaTeX compiler in the overleaf.com. The following is a suggested structure for your report. You can organize your report using these sections in this order.

- **Title, Author(s):** Please specify your chosen project name in the title, and enumerate all your teammates in the authors.
- **Abstract:** Briefly describe your problem, approach, and key results. It should be no more than 300 words.
- **Introduction (5%)**: Describe the problem you are working on, why it's important, and an overview of your results.
- **Related Work (5%)**: Discuss published work that relates to your project. How is your approach similar or different from others?
- **Data (5%)**: Describe the data you are working with for your project. What type of data is it? Where did it come from? How much data are you working with? Did you have to do any preprocessing, filtering, or other special treatment to use this data in your project?

- **Methods (30%)**: Discuss your approach to solving the problems that you set up in the introduction. Why is your approach the right thing to do? Did you consider alternative approaches? It may be helpful to include figures, diagrams, or tables to describe your method or compare it with other methods. **If you propose some innovations, describe the approaches in detail. (10%)**
- **Experiments (45%)**: Discuss the experiments that you performed to demonstrate that your approach solves the problem. The exact experiments will vary depending on the project. You should include graphs, tables, or other figures to illustrate your experimental results. **If you propose some innovations, you need to prove your improvements by qualitative and quantitative comparisons with existing methods. (20%)**
- **Conclusion (5%):** Summarize your key results - what have you learned? Suggest ideas for future extensions or new applications of your ideas.
- **Writing / Formatting (5%)** Is your paper clearly written and nicely formatted?
- Supplementary Material:

  Your supplementary material might include:

  - **Runnable** source code (obligatory).
  - A file illustrates your environment of code and running guidance, structured like this (obligatory).
  - Cool videos, interactive visualizations, demos, etc (optional).
  - Model checkpoints (optional).

Submission: You will submit your final report **in English** as a **PDF** and your supplementary material in **one** ZIP file.

Additional Submission Requirements: We will also ask you to do the following when you submit your project report:

- Your report PDF should list all authors who have contributed to your work. **All authors should be listed directly underneath the title on your PDF.**
- Any code used as a base for projects must be referenced and cited in the body of the report.