Homework 4

Deep Learning for Computer Vision

NTU, Fall 2025

TA: Ting-Hsun Chi, Kenneth Yang

Outline

- Problems & Grading
- Dataset
- Submission & Rules
- Supplementary

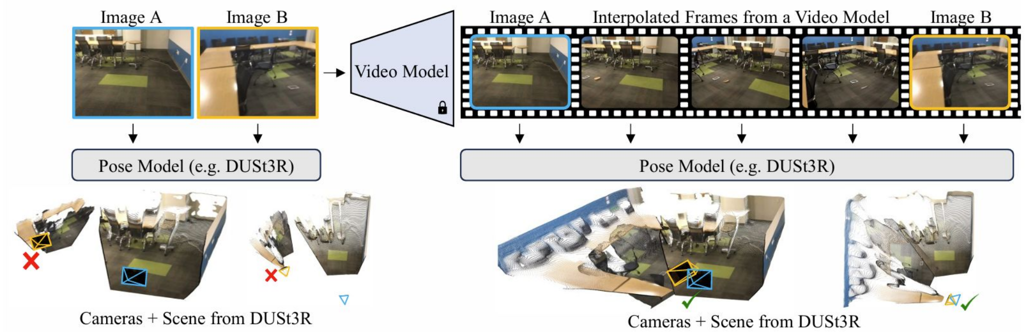
Problems – Overview

- Problem 1: Can Generative Video Models Help Pose Estimation? (50%)
- Problem 2: Sparse-View 3D Gaussian Splatting (50%)

Problem 1: Can Generative Video Models Help Pose Estimation?

Problem 1: Can Generative Video Models Help Pose Estimation?

- Task description
  - Recent groundbreaking models like DUSt3R perform impressively in standard scenarios.
  - **Limitation:** They often fail when applied to **wide-baseline** image pairs.
  - InterPose [CVPR'25 highlight] found that using video generative models to interpolate wide-baseline pairs can drastically boost performance.
  - In this problem, you will run DUSt3R on both the original wide-baseline pairs and the provided interpolated sequences, and then analyze the resulting performance difference.



Problem 1: Evaluation

[11/16 Updated]

Metric:

Mean Rotation Error (MRE), R_{Acc} , AUC

```
Evaluation Summary (Mode: R)
------------------------------------------------------------
Total samples processed: 40

Rotation Errors - 40 valid samples:
  Mean (MRE):   75.316°
    Acc < 5°:   2.50%
    Acc < 15°:   20.00%
    Acc < 30°:   42.50%

AUC (R) @ Threshold:
  AUC @ 30°:    0.2208
  AUC @ 15°:    0.0950
  AUC @ 5°:    0.0100
  AUC @ 3°:    0.0000
================================================================================
```

Evaluation Script:

```
bash scripts/public/run_check_metrics.sh
```

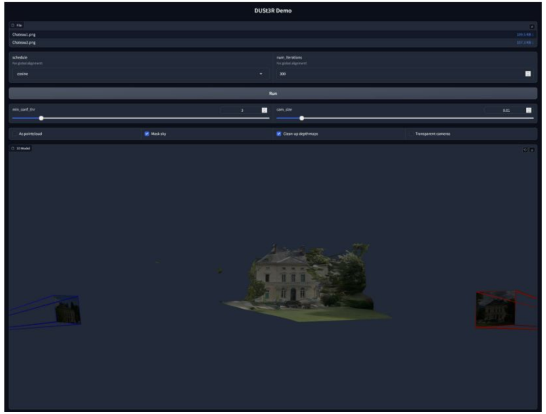
[11/16 updated]

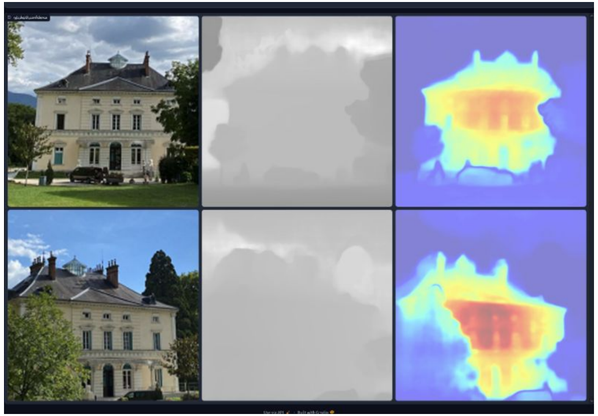
You are only required to implement the **TODO section** within the **inference_one_scene** function in **dust3r_inference.py**.

If implemented correctly, **your code will produce the same results as the TA's solution when run in the TA's environment**. Due to variances in hardware (device) and package versions, **you may observe minor differences when executing the code on your end.**

Problem 1: Grading – Report (30%)

- Q1 (10%)
    - Explain the DUSt3R and VGGT models and **analyze their key differences**. (5%)
    - **Run a demo** of DUSt3R using your own image sequence and **present the results**. (5%)





Problem 1: Grading – Report (30%) [11/16 Updated]

- Q2 (10%)

[11/16 updated]
Due to variations in **device or package versions**, the execution results **may exhibit minor discrepancies**. However, you should still observe a **clear performance gap** between **Pair Input** and **Video Input**.

(1) Quantitative Evaluation (On Public Dataset) (5%)

- Run DUSt3R on the **original wide-baseline pairs** and the **interpolated sequences** .
- Report the final evaluation summary for both runs.

```
================================================================================
Evaluation Summary (Mode: R)
------------------------------------------------------------
Total samples processed: 40

Rotation Errors - 40 valid samples:
  Mean (MRE):   84.004°
    Acc < 5°:   0.00%
    Acc < 15°:   20.00%
    Acc < 30°:   25.00%

AUC (R) @ Threshold:
  AUC @ 30°:     0.1483
  AUC @ 15°:     0.0717
  AUC @ 5°:     0.0000
  AUC @ 3°:     0.0000
================================================================================
```

```
================================================================================
Evaluation Summary (Mode: R)
------------------------------------------------------------
Total samples processed: 40

Rotation Errors - 40 valid samples:
  Mean (MRE):   75.316°
    Acc < 5°:   2.50%
    Acc < 15°:   20.00%
    Acc < 30°:   42.50%

AUC (R) @ Threshold:
  AUC @ 30°:     0.2208
  AUC @ 15°:     0.0950
  AUC @ 5°:     0.0100
  AUC @ 3°:     0.0000
================================================================================
```

Problem 1: Grading – Report (30%) [11/14 Updated]

- Q2 (10%)

(2) Case Study: idx 3781 PnP Solver Failure (5%):

The interpolated sequence for sample idx 3781 (from the public dataset) is known to fail during the PnP solving step.

- Present the MRE for **both** the original pair and the interpolated sequence relative to the ground truth.
- Use visual evidence (both the **DUSt3R-generated point cloud** and the **visual effect of the interpolated sequence itself**) to analyze *why* the PnP solver failed for this sequence.

Hint: A PnP solver is likely to fail when the 3D points are co-planar or co-linear.

Problem 1: Grading – Report (30%) [11/14 Updated]

- Q2 (10%)

(2) Case Study: idx 3781 PnP Solver Failure (5%):

The interpolated sequence for sample idx 3781 (from the public dataset) is known to fail during the PnP solving step.

Add try...except handling in dust3r/dust3r/cloud_opt/init_im_poses.py to catch the cv2.error.

```python
# success, R, T, inliers = cv2.solvePnPRansac(pts3d[msk], pixels[msk], K, None,
#                                              iterationsCount=niter_PnP, reprojectionError=5, flags=cv2.SOLVEPNP_SQPNP)

try:
    # --- This is the line that can crash ---
    success, R, T, inliers = cv2.solvePnPRansac(pts3d[msk], pixels[msk], K, None,
                                                iterationsCount=niter_PnP,
                                                reprojectionError=5,
                                                flags=cv2.SOLVEPNP_SQPNP)

except cv2.error as e:
    # --- This block "catches" the crash ---
    print(f"WARN: PnP failed for focal {focal} due to degenerate points: {e}")
    # We manually set success to False to make the loop continue
    success = False
```

(Note: You do not need to upload the dust3r folder).

Problem 1: Grading – Report (30%)

- Q3 (10%)
    - Compare the overall performance difference from your two runs.
    - Analyze these quantitative results and explain your high-level hypothesis. (i.e., *Why, in general, does this performance gap exist?*)

Problem 1: Grading – Implementation (20%)

Private Run 1 (10%):

- **Task:** Model inference on the **private wide-baseline pairs**.
- **Metric:** Graded based on MRE performance.

Private Run 2 (10%):

- **Task:** Model inference on the **private interpolated sequences**.
- **Metric:** Graded based on MRE performance.

If your model is implemented correctly, you should receive full credit for these sections.

Problem 2: Sparse-View 3D Gaussian Spaltting

Problem 2:

- Task description
  - In this problem, you will have to train your own **sparse-view 3D gaussians** as explicit scene representation for the given scene.
  - To be more specific, given a set of images (with camera pose) of this scene, learn the parameters of all 3D gaussians to fit this scene. After that, given the camera poses of testing images, render the test images for evaluation.
  - Please refer to InstantSplat LINK

Problem 2: Grading – Report (50%)

Q1 (15%)

- Try to explain the difficulty of sparse-view 3DGS in your own words. (5%)
- Compare 3D Gaussian Splatting with NeRF (pros & cons) (5%)
- Which part of 3D Gaussian Splatting is the most important you think? Why? (5%)

Problem 2: Grading – Report (50%)

Q2 (35%) Given novel view camera pose, your 3D gaussians should be able to render novel view images. Please evaluate your generated images and ground truth images with the following three metrics (mentioned in the InstantSplat paper). Try to use at least **three** different hyperparameter settings and discuss/analyze the results.

- **Please report the PSNR/SSIM/LPIPS on the image set.**
- **Different settings such as learning rate, iterations, densification, etc.**
- **Also attach visualization to the report and analyze the differences (additional visualization e.g. supersplat is encouraged.)**

Setting	PSNR	SSIM	LPIPS (vgg)
Setting 1 (You need to write your setting)	*TODO*	*TODO*	*TODO*
Setting 2 (You need to write your setting)	*TODO*	*TODO*	*TODO*
…			

Dataset

Tools for Dataset

- Download the dataset
  - Manually download the dataset https://drive.google.com/drive/folders/1QnxUnuygh6d9zIDGS8y83spoaugLG5Np?usp=drive_link

Problem 1

- The dataset consists of 40 public image pairs and 10 private image pairs.
- Format:

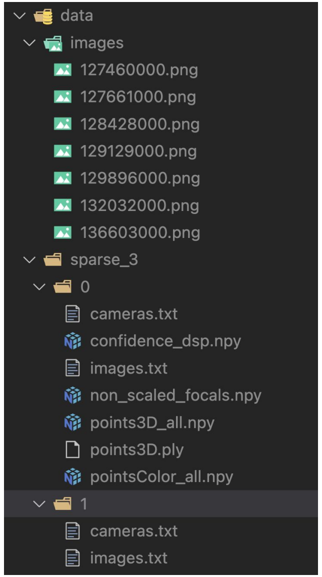
```
hw4_1_data
    private_yaw_80.0_to_90.0.txt        # 10 image pairs
    public_yaw_80.0_to_90.0.txt         # 40 image pairs
    public
        images
        interpolated_images
        gt.npy                          # ground truths for public dataset
    private
```

Problem 2

Folder structure

```
data/
    images/                    # training + public test images (7 images)
        {id}.png
    sparse_3/
      0/
          cameras.txt
          confidence_dsp.npy
          images.txt
          non_scaled_focals.npy
          points3D_all.npy
          points3D.ply
          pointsColor_all.npy
      1/
          cameras.txt      # public testing set (4 images)
          images.txt
```

- You **CANNOT** use testing data for training purposes.



Submission & Rules

Submission

- Click the following link to get submission repo with your GitHub account:
  https://classroom.github.com/a/4AUGnMVG
  - You should connect your Github account to the classroom with your **student ID**
  - If you cannot find your student ID in the list, please contact us (ntu-dlcv-2025-fall-ta@googlegroups.com)
- By default, we will grade your **last submission (commit) before the deadline**. Please e-mail the TAs if you'd like to submit another version of your repository and let us know which commit to grade.
- We will clone the **main** branch of your repository.

Submission

- Your GitHub repository should include the following files
  - hw4_<studentID>.pdf (report)
  - hw4_1_1.sh (for Problem 1)
  - hw4_1_2.sh (for Problem 1)
  - Python files (e.g., training code & inference code & visualization code)
  - Model files (can be loaded by your python file)
- <other notes>
- **Don’t push the dataset to your repo.**
- **If any of the file format is wrong, you will get zero point.**

Bash Script - Problem 1

- TA will run your code as shown below

Inference on Original Pairs

  - **bash hw4_1_1.sh $1 $2 $3 $4**
    - **$1:** Path to the TXT file containing the index.
    - **$2:** Path to the original image pair directory.
    - **$3:** Path to the model checkpoint.
    - **$4:** Path for the output prediction file.

Inference on Interpolated Sequences

  - **bash hw4_1_2.sh $1 $2 $3 $4 $5**
    - **$1:** Path to the TXT file containing the index.
    - **$2:** Path to the original image pair directory.
    - **$3:** Path to the interpolated sequence directory.
    - **$4:** Path to the model checkpoint.
    - **$5:** Path for the output prediction file.

- Note that you should **NOT** hard code any path in your file or script.
- Your testing code have to be finished in **30 minutes**.

Bash Script - Problem 2

- TA will run your code as shown below
  - **bash hw4_2.sh $1 $2**
    - $1: path to the **folder** containing test data and train data (e.g. hw4/p2_data/), will be exactly the same folder structure as the given dataset, with public test images/cameras changed to private images/cameras.
    - $2: path to the output **png** files (e.g. hw4/output_p2/), **you only have to render test images**

- Note that you should **NOT** hard code any path in your file or script.
- Your testing code have to be finished in **3 minutes**.

Rules – Bash Script

- You must **not** use commands such as **rm, sudo, CUDA_VISIBLE_DEVICES, cp, mv, mkdir, cd, pip** or other commands to change the environment.
- In your submitted script, please use the command **python3** to execute your testing python files.
    - For example: **python3 test.py $1 $2**
- We will execute you code on **Linux** system, so try to make sure you code can be executed on Linux system before submitting your homework.

Rules – Download Checkpoints

- If your checkpoints are larger than GitHub's maximum capacity (50 MB), you could download them in hw3_download.sh
  - TAs will run `**bash hw4_download.sh**` prior to any inference if the download script exists, i.e. it is NOT necessary to create a blank `hw4_download.sh` file.
- Do **NOT** delete your model checkpoints before the TAs release your score and before you have ensured that your score is correct.
- Your download script have to be finished in **10 minutes**.

Rules – Download Checkpoints (cont'd)

- Please use **wget** to download the model checkpoints from cloud drive (e.g. Dropbox) or your working station.
  - You should use **-O argument** to specify the filename of the downloaded checkpoint.
- Please refer to this Dropbox Guide for a detailed tutorial.
- Google Drive is a widely used cloud drive, so it is allowed to use gdown to download your checkpoints from your drive.
  - It is also recommended to use **-O argument** to specify the filename.
  - Remember to **set the permission visible to public**, otherwise TAs are unable to grade your submission, resulting in zero point.
  - If you have set the permission correspondingly but failed to download with gdown because of Google's policy, TAs will manually download them, no worries!!

Rules – Environment

- Ubuntu 22.04.4 LTS
- NVIDIA RTX A4500(20 GB)
- GNU bash, version 5.1.16(1)-release

Rules – Environment (cont'd)

- Ensure your code can be executed successfully on Linux system before your submission.
- Use only Python3 and Bash script conforming to our environment, do not use other languages and other shell during inference.
- You must **NOT** use commands such as **sudo, CUDA_VISIBLE_DEVICES** to interfere with the environment; **any malicious attempt against the environment will lead to zero point in this assignment.**
- You shall **NOT** hardcode any path in your python files or scripts, while the dataset given would be the absolute path to the directory.

Packages (problem1)

- check DUSt3R for packages and versions
- Others if you need (e.g., tqdm, gdown, glob, yaml, etc.)
- Any dependencies of above packages, and other standard python packages

*** E-mail or ask TAs first if you want to import other packages.**

Packages (problem2)

- check InstantSplat for packages and versions
- Others if you need (e.g., tqdm, gdown, glob, yaml, etc.)
- Any dependencies of above packages, and other standard python packages
- If you use the official 3DGS repo and your GPU has higher CUDA version, you can install corresponding Pytorch version

*** E-mail or ask TAs first if you want to import other packages.**

Ohter Reminders & Common Problems

- When using **wget**, surround the url with quotation marks (“) to prevent running in background.
- Use **os.path.join** to deal with path as often as possible.
- Avoid making assumptions about the arguments (such as paths or filenames).
- If you train on GPU ids other than 0, remember to deal with the “map location” issue when you load model.

Deadline and Academic Honesty

- **Deadline: 2025/12/2 (Tue.) 11:59 PM (GMT+8)**
- Late policy : Up to 3 free late days in a semester. After that, late homework will be deducted 30% each day.
- **Taking any unfair advantages over other class members (or letting anyone do so) is strictly prohibited. Violating university policy would result in F for this course.**
- Students are encouraged to discuss the homework assignments, but you must complete the assignment by yourself. TA will compare the similarity of everyone's homework. Any form of cheating or plagiarism will not be tolerated, which will also result in F for students with such misconduct.

Reproducibility

- If we cannot execute your code, TAs will give you a chance to make minor modifications to your code. After you modify your code,
    - You can not train a new model as a modification.
    - If we can execute your code, you will still receive a 30% penalty in the score corresponding to the modified code.
    - If we still cannot execute your code, no point will be given.

DOs and DONTs for the TAs & Instructor

- Do NOT send messages to TAs via their individual emails / social media.
- TAs are happy to help, but they are not your tutors 24/7.
- TAs will NOT debug for you, including addressing coding, environmental, library dependency problems.
- TAs do NOT answer questions not related to the course.
- If you cannot make the TA hours, please email the TAs to schedule an appointment instead of stopping by the lab directly.

How to Find Help

- Google or ChatGPT!
- Use TA hours (please check course website for time / location).
    - Please seek help from the **TAs in charge of this assignment** as possible as you can.
- Post your question under HW4 discussion section on NTU COOL.
- Contact TAs by e-mail: ntu-dlcv-2025-fall-ta@googlegroups.com
    - **Title should start with [DLCV 2025 Fall HW4]**
    - **Email with the wrong title may be filtered and not be replied.**