SAINT LOUIS
UNIVERSITY
EST. 1818

CSCI 4830/5830 Computer Vision

# Evaluation Report on LocoTrans: Rotation-Invariant Point Cloud Learning via Local-Context-Aware Transformers

**Final Project**

**Due date** 05/07/2025

Maryam Moshrefizadeh

Spring 2025

Summary of the Paper ........ 3
Implementation & Experimental Setup ........ 3
    Dataset Access & Format Issue ........ 3
    Memory Constraints ........ 4
    Custom DataLoader Implementation ........ 5
Evaluation Results on Original ModelNet40 Dataset ........ 5
Observations & Limitations ........ 6
Conclusion ........ 6
References ........ 7

## Summary of the Paper

The paper *"LocoTrans: Rotation-Invariant Point Cloud Learning via Local-Context-Aware Transformers"* proposes a novel transformer-based architecture for robust 3D point cloud classification under arbitrary rotations. The method achieves rotation invariance by combining local geometric priors with context-aware attention mechanisms. The authors evaluate their model on the ModelNet40 dataset using multiple rotation settings, achieving state-of-the-art accuracy, especially under SO(3) transformations.

## Implementation & Experimental Setup

I attempted to replicate the test results reported in the paper using their official GitHub repository LocoTrans[1]. However, several technical and practical challenges emerged during implementation:

### *Dataset Access & Format Issue*

The LocoTrans authors evaluate their models on the modelnet40_normal_resampled dataset, a preprocessed version of ModelNet40 in which:

- Each 3D object is uniformly resampled into 1024 points.
- The point clouds are normalized (centered and scaled to fit inside a unit sphere).
- The dataset is organized as .txt files, each containing XYZ coordinates (3 columns) for a single object.

Unfortunately, this specific dataset is no longer accessible through its original Stanford ShapeNet server. Alternative links (e.g., Google Drive, Hugging Face) are either invalid, truncated, or formatted differently (e.g., webdataset archives), and not compatible with the official LocoTrans codebase.

To work around this, I downloaded the **original ModelNet40 dataset from Kaggle**, which provides .off mesh files for 40 object classes. I then implemented a **custom preprocessing pipeline** that:

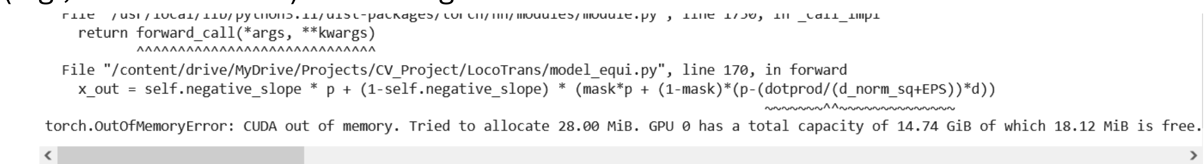
- Parses .off mesh files,

- Samples 1024 points uniformly from each mesh,
- Normalizes each point cloud to unit scale and center,
- Saves them into .txt files matching the expected format.

While this allowed the code to run, the test results did not match the ones reported in the paper.

### *Memory Constraints*

- The full model (main_cls.py) is very large and demands at least **4 high-end GPUs** (e.g., NVIDIA TITAN V) for training.




- Even testing the pretrained full model caused **CUDA out-of-memory errors** on a standard Google Colab GPU.
- As a result, I focused on the **lightweight version** (main_cls_l.py), which is significantly more efficient and suitable for limited hardware.

*Custom DataLoader Implementation*

- The provided data loading pipeline in provider.py was tightly coupled to the unavailable modelnet40_normal_resampled.
- I rewrote the DataLoader logic to support my locally preprocessed version of ModelNet40, ensuring compatibility with the training and evaluation pipeline.

# Evaluation Results on Original ModelNet40 Dataset

I tested the pretrained **lightweight model** (model_cls_l.py with model_1_z.t7) using my custom preprocessed dataset. The test accuracy was unexpectedly low:

```
100% 617/617 [42:22<00:00,  4.12s/it]
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:2524: UserWa
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:2524: UserWa
  warnings.warn("y_pred contains classes not in y_true")
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:2524: UserWa
  warnings.warn("y_pred contains classes not in y_true")
Test, test acc: 0.015802, test avg acc: 0.031111, best test acc: 0.015802
Test, test acc vn: 0.043355, test avg acc vn: 0.024406, best test acc vn: 0.043355
Test, test acc 1: 0.012966, test avg acc 1: 0.022980, best test acc 1: 0.012966
```

These values are far below the ~91.5% accuracy reported by the authors. Based on this, it is clear that **the pretrained models are tightly coupled to the exact version of the dataset used in the paper**. Even subtle deviations in point ordering, sampling distribution, class ordering, or normalization strategy can lead to dramatic drops in performance. Without access to the original modelnet40_normal_resampled version used by the authors, replicating their results becomes infeasible

## Observations & Limitations

- Due to the missing original dataset and reliance on custom preprocessing, **my evaluation results did not match** those reported in the paper (e.g., I obtained ~1–5% accuracy instead of ~91%).
- This discrepancy is likely due to subtle differences in data normalization, class ordering, and point sampling methods.
- Training from scratch was not feasible due to GPU memory limitations, and the pretrained models failed to generalize to my alternative dataset

## Conclusion

While the LocoTrans paper presents a strong theoretical contribution in rotation-invariant learning, replicating its results requires careful alignment with the exact data and hardware used by the authors. Inaccessible datasets and resource-heavy models pose major challenges for reproducibility. Future work may include implementing a full training pipeline with an open alternative dataset and verifying performance under various rotation conditions.

# References


[1] https://github.com/wdttt/LocoTrans

[2] https://arxiv.org/abs/2403.11113

[3] https://www.kaggle.com/datasets/balraj98/modelnet40-princeton-3d-object-dataset