NYU TANDON SCHOOL OF ENGINEERING

# Point Cloud Video Processing

Mingjian Li and Shihang Wei

NYU Video Lab. Department of Electrical and Computer Engineering. New York University

**Professor and Mentor**

Professor: Yao Wang

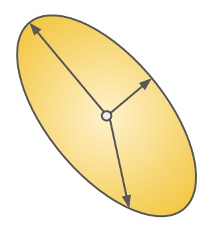
Mentor: Ran Gong

## Introduction

Point cloud video streaming remains challenging due to its high bandwidth and computational demands. As part of an interdisciplinary project at NYU Tandon focused on advancing dance education through volumetric video, we explore efficient representation, training, and rendering techniques for dynamic 3D content. Our work aims to reconstruct high-quality, streamable 3D scenes efficiently. This research contributes to enabling real-time point cloud streaming for applications in the performing arts, education, and beyond.

## Background

**3D Gaussian Splatting** has gained popularity for its ability to deliver fast and high-quality reconstruction and rendering of complex 3D scenes.



**Position(mean)**: (x,y,z) **Color**: (r,g,b)

**Scale**: ($s_x$,$s_y$,$s_z$) **Opacity**: $\alpha$

**Rotation**: r (quaternion) **Rendering**: projection with alpha blending

Training a separate 3DGS for each frame is time-consuming. Inspired by **4D Gaussian Splatting** we propose a more efficient solution by modeling temporal dynamics directly.

## Data Processing

**Camera Calibration:** Checkerboard images are used to estimate camera intrinsics and extrinsics.

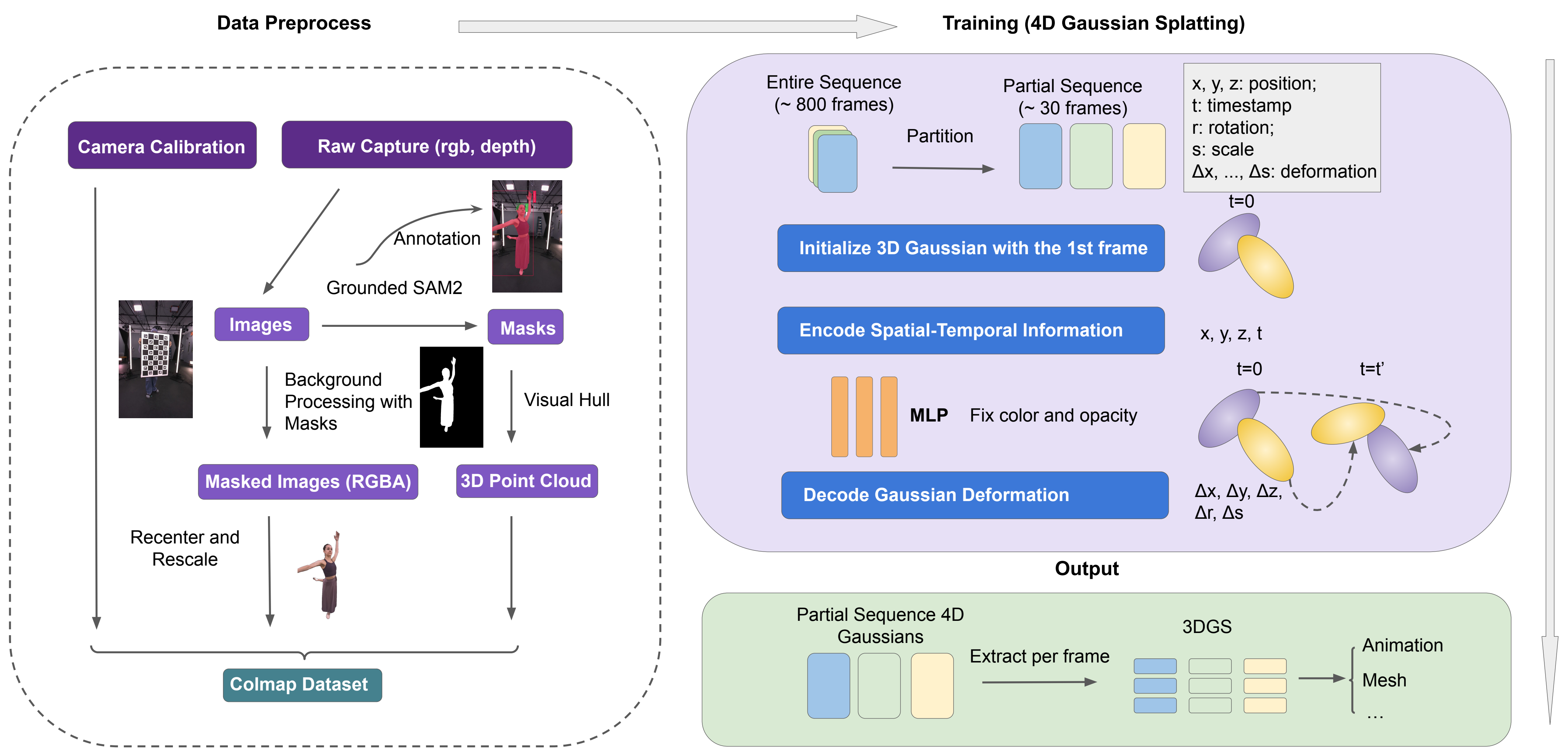
**Background Segmentation:** We use Grounded SAM2, a promptable model, to segment dancers from complex backgrounds. It reliably captures fine details such as hair and clothing with text prompts.

**Masked Images(RGBA):** The segmentation masks are also stored in the alpha channel of PNG images to create RGBA inputs.

**Visual Hull and Point Cloud:** Multiview masks are used to generate a visual hull, from which we construct the point cloud.

**Recenter and Rescale:** Since our camera principal points are not at the center of images, we recenter them by shifting the images, which is stable for pinhole cameras. Shifting can cause unwanted stripe cut on dancers, so we use a stable method to rescale the images without losing any information.

## Pipeline Overview



## Training Details

**Sequence Partitioning:** Each sequence is divided into smaller sub-sequences (30 frames) to enable more efficient training and achieve more stable, higher-quality scene reconstruction.

**3DGS Initialization:** The first frame is used to train a static 3D Gaussian Splatting (3DGS) model, which serves as the canonical base for subsequent frames.

**Spatial-Temporal Information Encoding:** Position and time are encoded to capture motion and deformation over time. Color and  Opacity remain unchanged.

**MLP Learning:** A lightweight multilayer perceptron (MLP) learns to map encoded features to Gaussian deformation parameters.

**Deformation Decoding:** The network decodes predicted offsets in position, rotation, and scale to dynamically update the Gaussian parameters across frames..

**3DGS Extraction and post-processing:** Per-frame 3DGS instances are extracted and optionally refined using filtering or pruning techniques to ensure visual quality and memory efficiency..

**Streaming 3D video:** The combined set of extracted 3DGS frames can be streamed or rendered in real time and interactable in common tools such as SuperSplat.

**Mesh Extraction:**  Meshes are extracted from 3DGS using Poisson Reconstruction or Marching Cubes algorithms.

## Evaluation

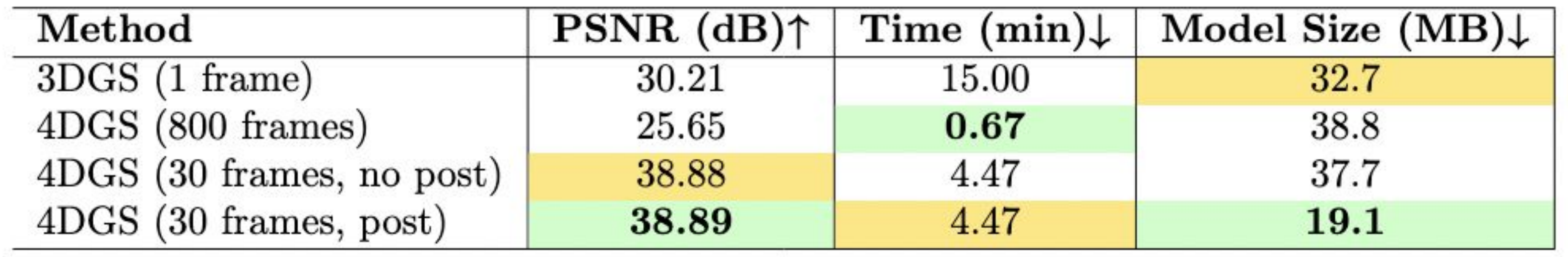
**Reconstruction Quality:** We use **PSNR** (Peak Signal-to-Noise Ratio) to quantitatively evaluate the fidelity of our rendered images against ground truth. Higher PSNR indicates better reconstruction accuracy.

**Training Time:** We assess training efficiency by measuring the average time required to train one frame.

**Storage Efficiency:** Model compactness is evaluated by the average size of the 3DGS representation per sequence. Lower storage footprint enables faster transmission and supports streaming applications.

**Visual Comparison:** We visually inspect the reconstructed 3D video to identify floaters, holes, and artifacts, providing qualitative assessment of geometric consistency and temporal stability.

## Result



| Method | PSNR (dB)↑ | Time (min)↓ | Model Size (MB)↓ |
|---|---|---|---|
| 3DGS (1 frame) | 30.21 | 15.00 | 32.7 |
| 4DGS (800 frames) | 25.65 | **0.67** | 38.8 |
| 4DGS (30 frames, no post) | 38.88 | 4.47 | 37.7 |
| 4DGS (30 frames, post) | **38.89** | 4.47 | **19.1** |

## Conclusion and Future Work

**Conclusion:** Our evaluation demonstrates that 4DGS with 30-frame sub-sequences and post-processing achieves the best overall performance in terms of reconstruction quality (highest PSNR), training efficiency, and model compactness.

**Future Work:**

- Incorporate depth maps to generate higher-quality point clouds
- Explore advanced compression techniques
- Enhance mesh extraction methods to produce cleaner, more complete 3D geometry.

## References

[1] Kerbl et al., 3D Gaussian Splatting for Real-Time Radiance Field Rendering, ACM TOG, 2023.

[2] G. Wu, Z el al., 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. CVPR, 2024.

ml8347@nyu.edu sw5672@nyu.edu