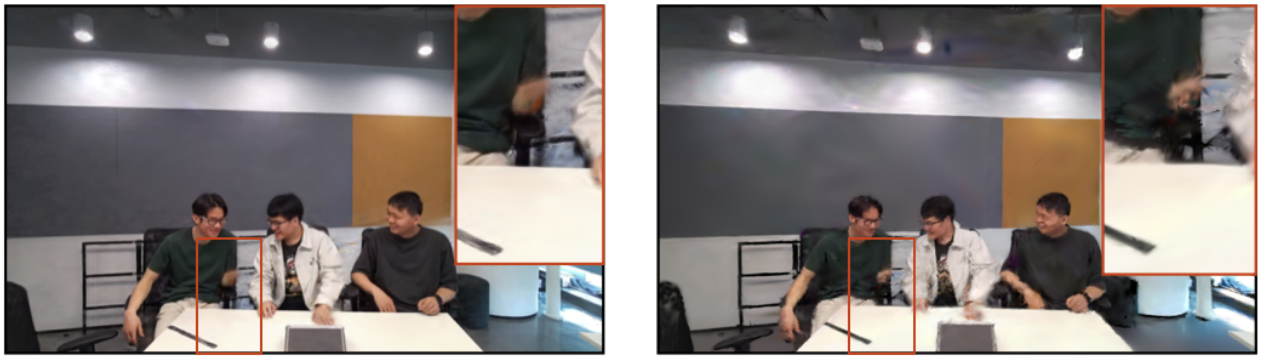
New Object Detection in Existing Dynamic Gaussian Splatting Methods

Semester Project Report

Ewa Miazga
ewa.miazga@epfl.ch
Supervisor: Saqib Javed



3DGStream HiCoM

Dynamic 3DG GT

Abstract

Reconstruction of 3D scenes using multi-view cameras remains a significant challenge today. Training is often computationally expensive and does not always guarantee high-quality reconstructions. Furthermore, existing methods may not be suitable for all types of 3D scenes—such as indoor or outdoor environments with many dynamic objects. One notable limitation is the inability to reconstruct objects that emerge after the first frame. Methods like HiCoM and 3DG-Stream rely on a per-frame optimization strategy that is initialized from the first frame, making them unable to model objects that appear later in the sequence. In contrast, Dynamic 3D Gaussian Splatting adopts a frame-to-frame optimization approach, which allows for continuous updates and appears better suited for handling newly emerging elements in dynamic scenes. This report provides a qualitative and quantitative comparison of the aforementioned methods, focusing on their ability to reconstruct newly appearing objects in dynamic 3D scenes.

1. Introduction

Efficient and effective 3D scene representation has the potential to significantly impact various industries, including robotics, gaming, and architecture. In architecture, it could enable interactive, localized modifications to building models and facilitate the generation of novel renderings from arbitrary viewpoints. In the gaming and film industries, it opens the door to dynamic object generation and manipulation for animations. Furthermore, accurate tracking of object motion is essential for applications such as autonomous driving and navigation in unfamiliar environments.

Despite these promising applications, 3D object reconstruction remains vulnerable to real-world challenges and limitations in spatial data acquisition. Current methods fall short in several scenarios. First, achieving high-resolution reconstruction for each frame while maintaining real-time performance is challenging. Second, scenes with multiple moving objects require high-frequency queries to the point cloud, which can lead to prohibitive computational costs per frame. Finally, and perhaps most critically, is the issue of reconstructing objects that appear after the first frame. Although this last challenge may seem trivial, it often results in misleading or incomplete reconstructions. In rendered scenes, these regions may appear partially transparent, revealing background content where newly appeared objects should have been reconstructed.

We ideally would seek to have a method that is able to overcome a challange and reconstruct every object appearing at the scene at any time. Still perserving the qualities of the methods like low storage, fast-rendering and high quality of the rendered output.

2. Dataset

The dataset plays a crucial role in this report, as it must present a fair and consistent challenge to all evaluated methods. It was important that the dataset be both accessible and compatible with the frameworks under test.

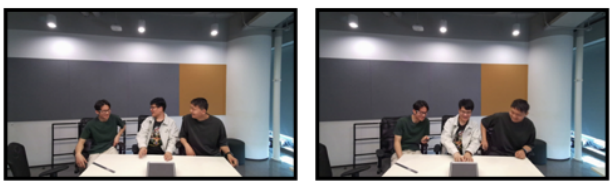


Figure 1. Meetroom dataset example. The individual takes a phone out of his pocket.

The *Meetroom* dataset was captured using a 13-camera multi-view setup. It consists of dynamic scenes recorded at a resolution of 1280×720 and 30 FPS. Specifically, the *discussion* sequence was used for evaluation. It is particularly suitable for testing new object reconstruction, as one of the individuals in the scene takes a phone out of his pocket after a few seconds—making it the primary focus of our evaluation (Figure 1).

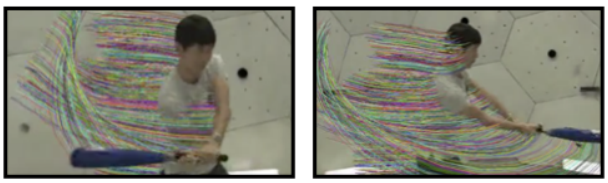


Figure 2. Panoptic dataset example. The batter swings through the air as no ball appears.

The appearance of a new object after the first captured frame was the decisive criterion in dataset selection. During our research, it became evident that datasets containing such characteristics are rarely used in evaluations—representing a significant but often overlooked limitation in existing benchmarks.

For instance, although the *Panoptic* dataset is a standard benchmark for evaluating dynamic Gaussian splatting methods, it does not feature objects that emerge during the sequence (Figure 2). This omission, while perhaps unintentional, can conflict with the logical and expected behavior of individuals in realistic scenes. As such, the evaluation may not reflect the true capabilities of reconstruction frameworks under dynamic conditions.

3. Methods

3.1. 3DGStream [1]

3DG-Stream is an on-the-fly rendering technique designed for fast and efficient free-viewpoint video reconstruction from multi-view video captures. Its key strengths include accelerated training, reduced storage requirements, and high-quality image synthesis.

Initial frame is optimized with 3D Gaussian Splatting technique [2]. Subsequent frames are obtained with two-stage optimization process on previous frame as shown in Figure 3.

Stage 1: Neural Transformation Cache Firstly, each frame passes through the Neural Transformation Cache — a shallow, fully-fused MLP — to obtain the position and rotation of individual gaussians for the consecutive frame.

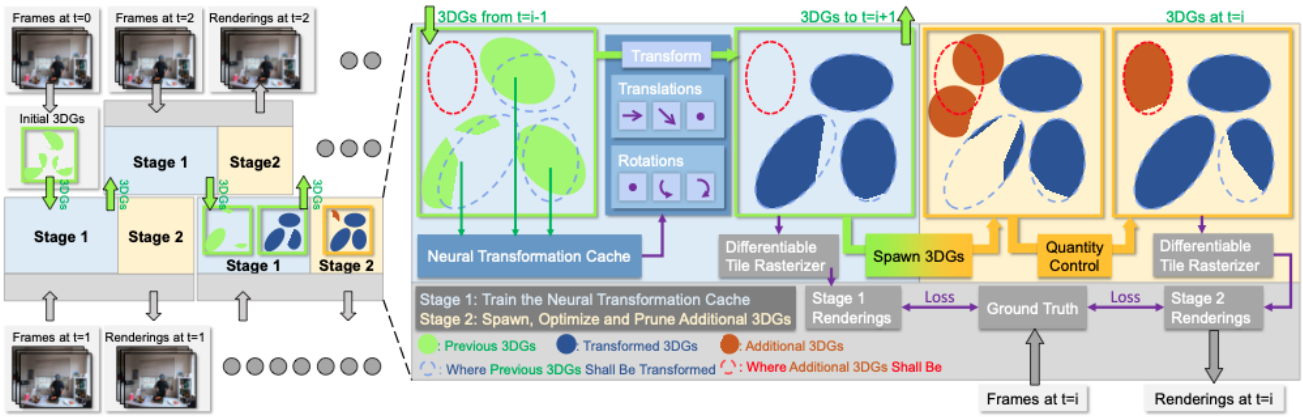


Figure 3. **3DGStream framework**. Reconstruction of 0 frame representation followed by reconstruction of subsequent frame representation using Hierarchical Coherent Motion mechanism and Continual Refinement with new gaussians.

To ensure storage efficiency, the method splits the scene into multi-resolution voxel grids. For each 3D position $\mathbf{x} \in \mathbb{R}^3$, its hash encoding at resolution level ℓ is denoted by:

$$h(\mathbf{x}, \ell) \in \mathbb{R}^d,$$

a d-dimensional feature vector. The function $h(\mathbf{x}, \ell)$ is constructed by an interpolation of the feature vectors at the eight corners of the voxel grid cell that surrounds the point $\mathbf{x}$. The multi-resolution hash encoding is defined as:

$$h(\mathbf{x}) = [h(\mathbf{x}, 0), h(\mathbf{x}, 1), \ldots, h(\mathbf{x}, L-1)] \in \mathbb{R}^{Ld},$$

which serves as input to the fully-fused MLP. For a gaussian centered at position $\boldsymbol{\mu}$, the MLP predicts the updates to its translation and rotation (1):

$$\Delta\boldsymbol{\mu}, \Delta\mathbf{q} = \text{MLP}(h(\boldsymbol{\mu})) \tag{1}$$

where $\Delta\boldsymbol{\mu} \in \mathbb{R}^3$ is the translation offset and $\Delta\mathbf{q} \in \mathbb{R}^4$ is the rotation quaternion.

Based on the MLP outputs—translations, rotations, and spherical harmonics (SH) rotation coefficients—the attributes of the gaussians are updated for the subsequent frame.

Loss Function The optimization of Neural Transformation Cache (NTC) parameters for future frames relies on the computation of a loss function between the rendered image and the ground truth. Following the formulation introduced in 3D Gaussian Splatting [2], the loss is defined as a weighted combination of L1 and D-SSIM losses:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{\text{D-SSIM}} \tag{2}$$

where $\lambda = 0.2$ is a balancing factor.

The gaussians obtained at the end of Stage 1 are frozen and serve as initialization for the next frame's optimization.

Stage 2: Representing Emerging Objects Stage 2 focuses on modeling newly emerging objects—such as flames, smoke, or liquids—that appear in the scene. These phenomena are not adequately captured by the gaussians inherited from the previous frame. To address this, new gaussians are introduced in frame-specific regions exhibiting poor reconstruction. This ensures that only a minimal number of new gaussians are added, avoiding unnecessary accumulation over time.

Gradient-Based Localization and Adaptation To localize regions where new objects are emerging, the method monitors high view-space positional gradients. These occur when the model attempts—but fails—to approximate unrepresented scene elements using existing gaussians. Since the color attributes of the gaussians are frozen during Stage 1, this leads to large positional updates without photometric improvements.

To handle this, an adaptive 3DG spawning strategy is employed. New gaussians are initialized in high-gradient regions and optimized using the same loss function $\mathcal{L}$ (2).

Furthermore, at the end of each training epoch, an adaptive 3DG quantity control strategy is applied: gaussians in under-reconstructed regions are either deleted or split to enhance spatial coverage and reconstruction quality.

3.2. HiCoM [3]

The method was introduced as learning and storage efficient alternative to the other existing streamable frameworks.

The initalization step optimizes first frame view using 3D Gaussian Splatting [2]. As shown in Figure 4 (a), small

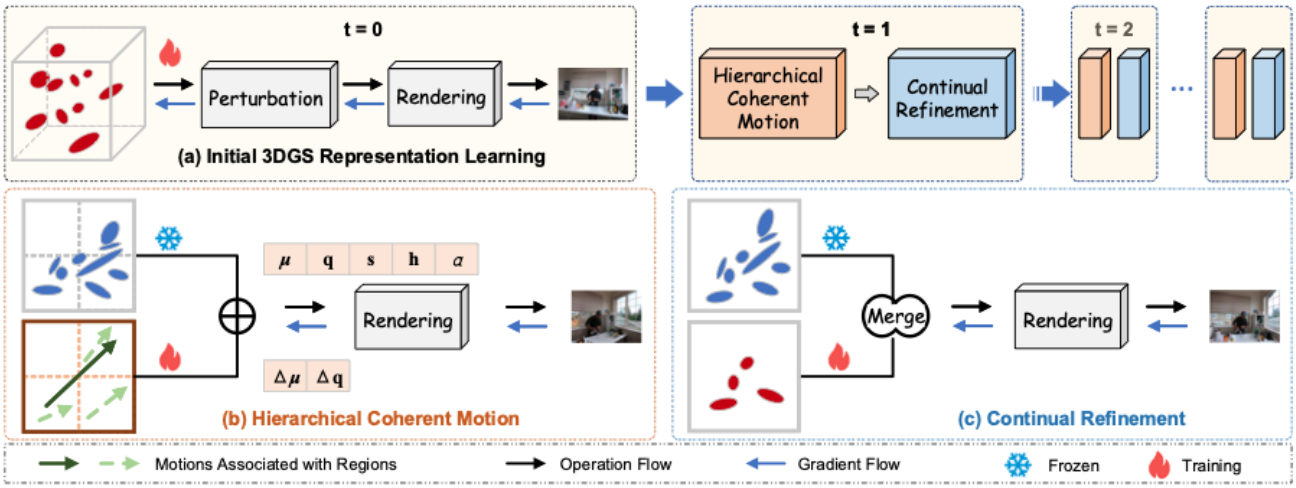


Figure 4. **HiCoM framework**. Reconstruction of 0 frame representation (a), Reconstruction of subsequent frame representation using Hierarchical Coherent Motion mechanism (b), Continual Refinement with new gaussians.

gaussian noise is added to the position attribute of each 3D gaussian. To accomdate for possible overfitting caused by limited access to camera views.

Hierarchical Coherent Model For the next frames the training follows online framework, where the latest view is continuously fetched and used for optimization of current state representation. Then, Hierarchical Coherent Model Figure 4 (b) is used to capture the movement of the gaussians from frame to frame. The model is based on partitioning the space into regions in gradual manner (from more coarse to fine) and then calculating vector $\Delta\mu \in \mathbb{R}^3$ and a quaternion $\Delta q \in \mathbb{R}^4$ for each granulation level. Then for each gaussian we calculate the motion according to formula (3):

$$\Delta\boldsymbol{\mu}_g = \sum_{l=1}^{L} \Delta\boldsymbol{\mu}^l, \quad \Delta\mathbf{q}_g = \sum_{l=1}^{L} \Delta\mathbf{q}^l \tag{3}$$

Continual Refinement Following the motion update of gaussians, 3D Gaussian Splatting is applied to achieve the recent scene renders. However, the hierarchical coherent model is not capable of capturing finer details, especially in the regions with significant gradient changes. There the gaussians are densified and optimized, so the discrepancy betweeen the rendered and target image is reduced. To control the number of gaussians, the same amount that was injected, is removed based on low opacity criterion. The forementioned mechanism is called Continual Refinement and it is shown in Figure 4 (c).

The training in this framework may be done in parallel. Chosen frame t is a referance for k next frames, which makes simultanous training of $\{t+1, \cdots, t+k\}$ possible.

3.3. Dynamic 3D Gaussians [4]

The method combines tasks of dynamic scene reconstruction and tracking of all dense scene elements using multi-view videos with segementation masks.

The first frame is optimized following the 3D Gaussian Splatting method [2]. After the optimization, the gaussians are configured to represent the first frame, including their size, color, opacity, and a background logit that indicates whether a gaussian belongs to the static background. These attributes are then frozen for the remainder of the sequence.

In subsequent frames, only the 3D centers (x_t, y_t, z_t) and 3D rotations $(q_{w,t}, q_{x,t}, q_{y,t}, q_{z,t})$ of the gaussians are updated, enabling motion. In other words, each gaussian is treated as a persistent attribute of an object that existed in the first frame.

Physically-Inspired Regularization Losses Fixing only the appearance-related attributes (color, opacity, size) is not sufficient to ensure accurate gaussian tracking, especially in textureless or ambiguous regions. Therefore, the optimization process is constrained using three physically-based priors:

1. Local Rigidity Loss This term encourages nearby gaussians to move as if part of a locally rigid body between consecutive frames:

$$\mathcal{L}_{\text{rigid}} = \frac{1}{k|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}(i)} w_{i,j} \left\| \Delta \mu_{ij}^{t-1} - \mathbf{R}_{i,t-1} \mathbf{R}_{i,t}^{-1} \Delta \mu_{ij}^{t} \right\|_2 \tag{4}$$

- $\mathbf{R}_{i,t}$: rotation matrix of gaussian i at time t
- $w_{i,j}$: weighting function (e.g., gaussian based on spatial proximity)
- $|\mathcal{S}|$: number of gaussians being considered
- $\mathcal{N}(i)$: the set of the k nearest neighbors of gaussian i, determined based on the Euclidean distance in 3D space.
- $\Delta \boldsymbol{\mu}_{ij}^{t} = \boldsymbol{\mu}_{j,t} - \boldsymbol{\mu}_{i,t} \in \mathbb{R}^3$ represents the relative position vector from gaussian i to gaussian j at time step t.

2. Rotation Similarity Loss This loss encourages consistent rotation behavior across neighboring gaussians:

$$\mathcal{L}_{\text{rot}} = \frac{1}{k|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{j \in \text{knn}(i;k)} w_{i,j} \left\| \hat{q}_{j,t} \hat{q}_{j,t-1}^{-1} - \hat{q}_{i,t} \hat{q}_{i,t-1}^{-1} \right\|_2 \tag{5}$$

- $\hat{q}_{i,t}$: unit quaternion representing the rotation of gaussian i at time t

This Local Rigidity Loss and Rotation Similarity Loss are applied only between the current and previous timestep.

3. Isometry Loss The isometry loss preserves relative distances between neighboring gaussians across frames to prevent global drift:

$$\mathcal{L}_{\text{iso}} = \frac{1}{k|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}(i)} w_{i,j} \left| \left\| \Delta \boldsymbol{\mu}_{ij}^{0} \right\|_2 - \left\| \Delta \boldsymbol{\mu}_{ij}^{t} \right\|_2 \right| \tag{6}$$

- This penalizes deviations from the original pairwise distances defined at frame $t = 0$, helping maintain structural consistency.

Each of these loss terms contributes to a robust and physically plausible deformation of the gaussian field over time.

4. Results

4.1. Quantitive Comparison

All three methods were trained on a dataset prepared according to the specifications outlined by the original authors. In most cases, this required separate processing for the first frame and subsequent frames.

The reported PSNR values indicate that all three methods achieve similar reconstruction quality, consistent with the results presented in their respective papers.

Method	**PSNR** ↑ (dB)	**Storage** ↓ (MB)
3DG-Stream	23.85	7.60 / 7.66*
HiCoM	25.11	0.21 / 0.25*
Dynamic Gaussians	24.49	7.34

Table 1. **Evaluation on the distorted dataset.** * Indicates per-frame storage when including the initial point cloud.

These results suggest that the majority of the scene was reconstructed successfully and with high fidelity. The HiCoM method achieves significantly lower storage requirements, highlighting a key improvement in efficiency compared to other approaches.

HiCoM achieves the highest PSNR, likely due to its accurate reconstruction of static background regions that closely match the ground truth. However, a closer inspection reveals that the method introduces a significant amount of noise in areas corresponding to moving objects. This noise suggests poor handling of dynamic content, which is an undesirable characteristic of the algorithm despite its strong performance on static regions.

It is worth noting that the training time for 3DG-Stream and HiCoM was significantly shorter—by a factor of approximately 6—compared to Dynamic Gaussians, making them more efficient for practical use.

4.2. Qualitative Comparison

Eventhough all the methods yield satisfactory results in terms of metrics, it is impossible to tell if our evaluation objective was met. To gain comprehensive insight into quality of new object reconstruction in the rendered frames from test camera.

3DGStream The emerging object—a phone—fails to be reconstructed, indicating a shortcoming in Stage 2 of the optimization process, which is designed to account for new objects appearing in the scene. Stage 2 relies on detecting high view-space positional gradients to localize under-reconstructed regions. However, in this case, the phone is small, blue in color, and partially blends with a green T-shirt background. Its motion is also minimal. These combined factors likely result in positional gradients that are too weak to surpass the threshold required for new Gaussian spawning. As a result, the model does not recognize this region as requiring additional representation.

Moreover, the loss function used is based on SSIM, computed over the full frame. Because the phone occupies only about 0.25% of the image area, reconstruction errors in this region contribute negligibly to the overall loss. This weak

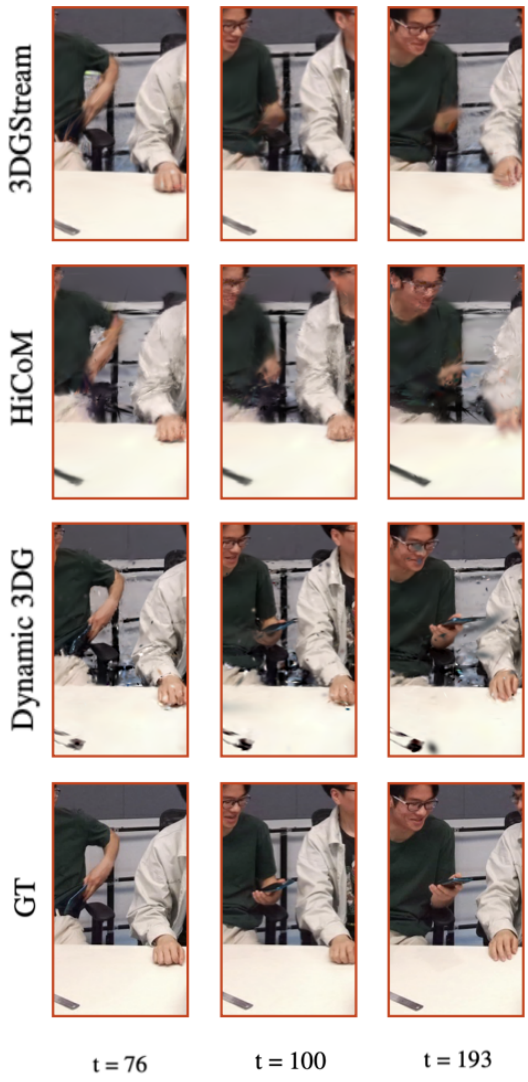


Figure 5. Cropped renderings of selected frames from test camera 0 on the Meetroom dataset.

supervision fails to incentivize the model to make localized corrections.

One possible solution could involve introducing an attention mechanism to focus on small, under-reconstructed regions. However, such mechanisms may be sensitive to noise and could introduce instability, making this approach potentially unreliable in practice.

HiCoM Similar to 3DG-Stream, HiCoM also fails to reconstruct the phone. However, the underlying reasons are fundamentally different. The method does not incorporate any explicit mechanism for detecting or modeling newly emerging objects in the scene. Its Continual Refinement stage only densifies existing gaussians by duplicating or splitting them in regions identified as underrepresented—typically based on local rendering error. Since new gaussians can only be created through refinement of already existing ones, the model cannot represent objects that were entirely absent in the first frame. If no gaussians were initially placed in the region where the phone appears, there are no anchor points for refinement, and the phone cannot be reconstructed.

The visual artifacts around moving individuals are likely due to limitations of the Coherent Motion model. The hierarchical motion estimation operates over spatial partitions and applies shared transformations, which may be too coarse to capture fine-grained, per-gaussian movements—especially those involving small or fast-moving body parts like limbs. This highlights a key limitation: although moving objects may occupy a small portion of the scene, they are perceptually dominant. A model focused on gradual, region-level motion may reconstruct static backgrounds well, but struggle to represent the subtle and complex dynamics of foreground motion, leading to unnatural blending or ghosting artifacts.

Dynamic 3D Gaussians In contrast to the previous methods, Dynamic 3D Gaussians successfully reconstructs the phone, despite the authors stating that modeling newly emerging objects is a limitation of their approach. While no new gaussians are introduced during training, the existing ones are allowed to move and rotate freely across frames. Unlike HiCoM, which applies motion in a coarse, region-based manner, Dynamic 3D Gaussians performs motion estimation at the level of individual gaussian. This flexibility enables some gaussians—originally associated with the T-shirt in the first frame—to be displaced toward the region where the phone appears.

This behavior is guided by a combination of physically-inspired regularization losses: local rigidity, rotation consistency, and isometry. These priors ensure that gaussians move in a coherent and physically plausible manner while allowing sufficient flexibility to capture local deformations. Additionally, the phone's color similarity to the surrounding T-shirt may have contributed to the reconstruction, as it allows nearby gaussians with similar appearance attributes to approximate the phone without requiring new gaussians to be added.

To fully assess the limitations of Dynamic 3D Gaussians in modeling newly emerging objects, one would need to evaluate it on a dataset where a new object with distinct color and structure enters the scene—such that no similar gaussians exist nearby in the first frame. In such a scenario,

it is likely that the method would struggle to reconstruct the object, just as 3DG-Stream and HiCoM do. This suggests that none of the examined methods provide a complete solution for handling new object emergence in dynamic 3D scenes.

4.3. Undistorted dataset

Method	**PSNR** ↑ (dB)	**Storage** ↓ (MB)
3DG-Stream	<u>33.30</u>	7.6 / 7.66⋆
HiCoM	24.93	0.31 / 0.35⋆
Dynamic Gaussians	22.73	7.71

Table 2. Evaluation on **undistorted dataset** . ⋆ indicates storage per frame with initial point cloud.

3DG-Stream emphasizes the importance of undistorting subsequent camera frames. This process removes lens distortion from the input images and updates the associated camera parameters, resulting in a rectified version of the scene that is more suitable for geometric processing.

To ensure a fair comparison, additional experiments were conducted on the undistorted version of the dataset to assess whether this preprocessing step benefits other methods. However, a significant improvement was observed only for 3DG-Stream, which demonstrated a 39% performance increase. The results for the other methods remained within the standard deviation observed across multiple runs.

4.4. Dynamic 3D Gaussians with Meetroom Dataset

Segmentation The method inherently uses segmentation masks to distinguish between static background and dynamic foreground gaussians. However, in the experiments conducted for this report, segmentation masks were not available. To enable training, all gaussians were assigned to the foreground by setting their background logits to zero. This effectively disables the segmentation-based split that is normally used to prevent losses such as rigidity, rotation similarity, and isometry from being applied between dynamic and static components of the scene.

In the standard setup, several loss terms rely on the segmentation split:

$$\mathcal{L}_{\text{floor}} = \frac{1}{N_{\text{fg}}} \sum_{i \in \mathcal{F}} \max(y_i, 0), \tag{7}$$

$$\mathcal{L}_{\text{bg}} = \frac{1}{N_{\text{bg}}} \sum_{i \in \mathcal{B}} \left(\left\| \mathbf{x}_i - \mathbf{x}_i^{(0)} \right\|_1 + \left\| \mathbf{R}_i - \mathbf{R}_i^{(0)} \right\|_1 \right), \tag{8}$$

$$\mathcal{L}_{\text{seg}} = 0.8 \cdot \text{L1}(\hat{\mathbf{s}}, \mathbf{s}) + 0.2 \cdot (1 - \text{SSIM}(\hat{\mathbf{s}}, \mathbf{s})), \tag{9}$$

- $\mathcal{F}$ and $\mathcal{B}$ denote the sets of foreground and background Gaussians respectively,
- y_i is the Y-coordinate of the i-th Gaussian,
- $\mathbf{x}_i^{(0)}$ and $\mathbf{R}_i^{(0)}$ are the initial positions and rotations of background Gaussians,
- $\hat{\mathbf{s}}, \mathbf{s}$ are the predicted and ground truth segmentation maps.

In experiment setting, where no segmentation masks are used, we omit the background loss $\mathcal{L}_{\text{bg}}$ and segmentation loss $\mathcal{L}_{\text{seg}}$ from the final loss function. This effectively allows all gaussians to move freely across the entire scene. On one hand, this flexibility enables the model to reconstruct new objects such as the phone. On the other hand, it introduces instability, leading to visible artifacts—commonly referred to as "flying gaussians."

Learning Rates and Loss Balancing Adjusting the weight of the floor loss $\mathcal{L}_{\text{floor}}$ was found to have minimal impact on the suppression of artifacts. In particular, setting this term to zero did not eliminate artifacts, while increasing it too much dominated the total loss and led to degraded overall performance, indicating an imbalance in the optimization dynamics.

Overfitting and Camera Layout A potential contributing factor to the observed artifacts is overfitting to initial frame, particularly in the absence of sufficient view diversity. The Meetroom dataset includes 27 cameras, but they are primarily positioned in front of the subjects, offering limited angular coverage. This sparse view distribution may lead to incorrect 3D placement of gaussians—especially in depth—during the per-frame optimization, as the model may overfit to 2D projections without strong multi-view constraints. Consequently, background gaussians may drift into dynamic regions, producing visual artifacts such as "flying gaussians."

To test this hypothesis, we repeated the same training setup—without segmentation masks—on the Panoptic dataset, which features a more diverse camera layout. In this case, the artifact pattern did not emerge, supporting the assumption that limited view diversity contributes to the issue.

5. Conclusion

In conclusion, none of the evaluated methods fully address the challenge of reconstructing new objects that emerge in a scene. **3DGStream** fails likely due to the limitations of its threshold-based detection mechanism, **HiCoM** lacks any explicit strategy to accommodate the appearance of new objects.

Dynamic 3D Gaussians shows some potential in handling emerging objects, but a conclusive assessment is not possible in this study. The method depends on segmentation

masks to differentiate between static and dynamic regions; however, such masks were unavailable for the dataset used. Moreover, the camera placement in the Meetroom dataset provides limited depth variation, which may lead to incorrect initial 3D positioning of Gaussians and visible artifacts in subsequent frames. This undermines the reliability of the method in this setting.

6. Future Work

Future investigations should utilize datasets in which newly emerging objects have distinct appearances, clearly different from any other objects present in the initial frame. Additionally, datasets should feature more diverse and comprehensive multi-view camera placements that offer better coverage of the scene—particularly from different angles and depths—to ensure robust 3D reconstruction.

Since Dynamic 3D Gaussians relies on accurate segmentation and sufficient depth cues to properly track and reconstruct dynamic elements, future work should explore how this method performs when segmentation masks are available and when camera configurations provide adequate spatial information. These factors are critical for validating whether the method can truly generalize to scenes with emergent objects.

The challenge of reconstructing objects that appear after the first frame remains largely unresolved. Existing methods—such as 3DGStream, HiCoM, and Dynamic 3D Gaussians—each fall short for different reasons. Therefore, future methods should explicitly incorporate mechanisms for detecting and modeling novel scene elements as they appear over time, without requiring prior knowledge or visibility in the initial frame. Addressing this gap will be crucial for advancing dynamic scene reconstruction in real-world applications.

References

[1] J. Sun, H. Jiao, G. Li, Z. Zhang, L. Zhao, and W. Xing, "3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos," 2024. [Online]. Available: https://arxiv.org/abs/2403.01444 2

[2] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," 2023. [Online]. Available: https://arxiv.org/abs/2308.04079 2, 3, 4

[3] Q. Gao, J. Meng, C. Wen, J. Chen, and J. Zhang, "Hicom: Hierarchical coherent motion for streamable dynamic scene with 3d gaussian splatting," 2024. [Online]. Available: https://arxiv.org/abs/2411.07541 3

[4] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," 2023. [Online]. Available: https://arxiv.org/abs/2308.09713 4