3DGS-QuaNTiC: Live Streaming Dynamic 3D Gaussian Splatting Scenes with Quantized Neural Transformation Cache

Públio Elon Correa da Silva
Department of Computer Science
Federal University of São Carlos (UFSCar)
Sorocaba, São Paulo, Brazil
publio@estudante.ufscar.br

Jurandy Almeida
Department of Computer Science
Federal University of São Carlos (UFSCar)
Sorocaba, São Paulo, Brazil
jurandy.almeida@ufscar.br

Fábio Luciano Verdi
Department of Computer Science
Federal University of São Carlos (UFSCar)
Sorocaba, São Paulo, Brazil
verdi@ufscar.br

Abstract

In this demo, we present 3DGS-QuaNTiC, a framework that enables streaming of dynamic 3D Gaussian Splatting scenes using quantized Neural Transformation Caches (NTCs). 3DGS-QuaNTiC transmits an initial 3DGS representation and progressively streams compressed NTC updates through a TCP socket. It supports FP16, INT8, and INT4 NTC compressed models, as well as reduced update-rate streaming using a stride parameter. The combined NTC compression and stride-based update reduction decrease the bandwidth required to stream dynamic 3DGS scenes. In our demo scene, the FP32 NTC baseline requires approximately 1.88 Gbps at 30 FPS, while INT4 streaming with an NTC stride of 5 reduces the NTC update stream to approximately 48.5 Mbps.

CCS Concepts

• **Information systems** → **Multimedia streaming**; • **Computing methodologies** → *Rendering*; • **Theory of computation** → *Data compression.*

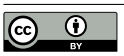
Keywords

3D Gaussian Splatting, dynamic 3DGS streaming, model compression, TCP/IP streaming

ACM Reference Format:
Públio Elon Correa da Silva, Jurandy Almeida, and Fábio Luciano Verdi. 2026. 3DGS-QuaNTiC: Live Streaming Dynamic 3D Gaussian Splatting Scenes with Quantized Neural Transformation Cache. In *Proceedings of the 34th ACM International Conference on Multimedia (MM '26), November 10–14, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3767308.3839005

1 Introduction

3D Gaussian Splatting (3DGS) represents radiance fields as anisotropic Gaussians with a fast differentiable rasterizer [2], enabling interactive novel-view synthesis. However, network delivery remains challenging due to large scene representations and high startup latency. Although prior work has explored 3DGS compression [1], radiance-field streaming [3, 5], and dynamic 3DGS rendering [6], research involving streaming of dynamic 3DGS scenes over the network remains limited and this demo aims to reduce this gap.



This work is licensed under a Creative Commons Attribution 4.0 International License.
MM '26, Rio de Janeiro, Brazil
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2213-4/2026/11
https://doi.org/10.1145/3767308.3839005

3DGStream [4] addresses dynamic free-viewpoint video using an initial 3D Gaussian representation and a sequence of Neural Transformation Caches (NTCs), which are learned neural models that predict per-frame transformations of the initial Gaussians. This avoids storing a complete 3DGS model for every frame, but full-precision NTC delivery remains bandwidth intensive: in our demo scene, the FP32 baseline requires approximately 2.3 GB for a 10-second sequence, or about 1.88 Gbps at 30 FPS.

3DGS-QuaNTiC extends 3DGStream with quantized and temporally subsampled NTC streaming over TCP/IP. It sends the initial 3DGS scene once, progressively streams FP16, INT8, or INT4 NTC updates, and supports reduced update-rate modes that transmit sparse NTCs. When an intermediate NTC is unavailable, the receiver reuses the most recent available NTC, preserving interactive 3D rendering while reducing temporal bandwidth. The demo implementation and instructions on how to run it are available online.[1]

2 System Design

3DGS-QuaNTiC follows a sender-receiver architecture for dynamic 3DGS streaming, as shown in Figure 1. The sender sends the initial 3DGS scene once and then progressively streams Neural Transformation Cache (NTC) updates that encode temporal deformation. The receiver caches the initial scene and incoming NTCs, reconstructs the dynamic sequence, and renders it in the interactive viewer shown in Figure 2.

Sender. The sender operates on a prepared 3DGStream scene containing an initial Gaussian representation and precomputed NTC checkpoints. It first transfers the initial scene and NTC configuration, then streams the NTCs according to the selected precision and update-rate mode. NTCs may be stored as FP16 checkpoints or as INT8/INT4 blockwise-quantized checkpoints with scale metadata. In full-rate mode, one NTC is sent per timestep; in reduced update-rate modes, the sender transmits sparse NTCs using a stride parameter. Together, quantization and stride-based transmission reduce the bandwidth required for dynamic NTC updates.

Transport. 3DGS-QuaNTiC uses a TCP/IP socket. Each file carries lightweight metadata, including its relative path and byte size, allowing the receiver to rebuild the scene cache. TCP provides reliable ordered delivery, keeping the demo simple and avoiding packet-loss recovery logic.

Receiver. The receiver caches the initial scene, NTC configuration, and incoming NTC updates. Playback starts after the initial scene and early NTCs are available, forming a small cache-ahead buffer. During playback, the receiver lazily loads only the NTC

[1] https://github.com/publioelon/QNTC-Stream-Demo.

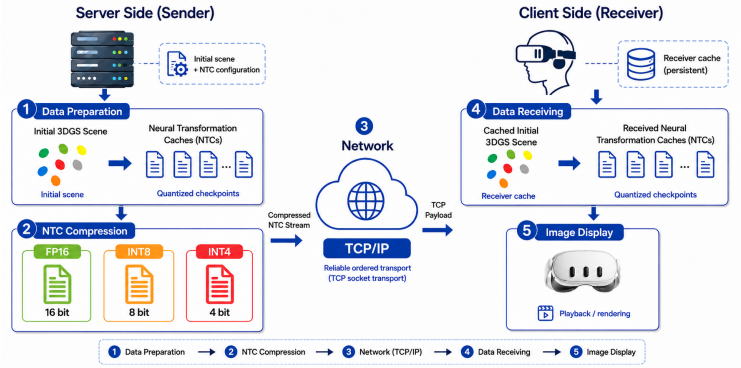


Figure 1: Overview of 3DGS-QuaNTiC. The sender streams the initial 3DGS scene and quantized NTC updates over TCP/IP, while the receiver reconstructs playback and locally monitors streaming statistics.

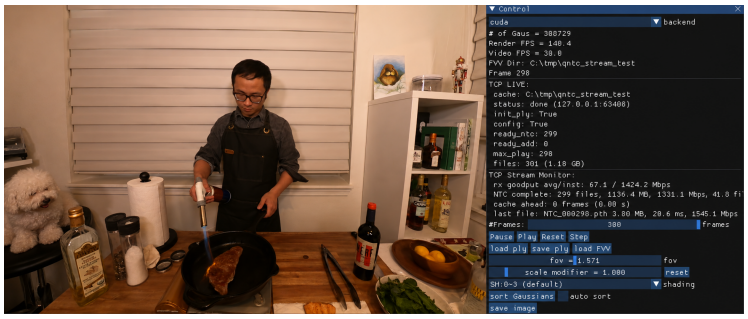


Figure 2: 3DGS-QuaNTiC demo interface showing dynamic 3DGS playback, compression statistics, TCP stream monitoring, NTC goodput, cache-ahead, and supported-mode indicators.

state required for the current timestep. Quantized checkpoints are dequantized before inference, so FP16, INT8, and INT4 streams share the same reconstruction path.

If the NTC for the current timestep has arrived, the renderer applies it to deform the cached Gaussian scene. If it is missing because reduced update-rate streaming is used, the receiver reuses the most recent available NTC. This preserves interactive 3D rendering and camera motion while reducing temporal bandwidth. Late NTCs can be ignored to avoid processing stale updates and accumulating delay.

Demo interface. Figure 2 shows the 3DGS-QuaNTiC viewer. The interface displays playback controls, compression statistics, TCP throughput, useful NTC goodput, NTC reception rate, cache-ahead, and supported-mode indicators. Throughput measures all bytes written to the cache, while useful NTC goodput isolates the bandwidth used by NTC updates. Cache-ahead indicates how much future playback data is available, and the supported-mode indicator reports which precision and update-rate mode can be sustained by the measured connection.

3 Demo

In this demo, we present three components for dynamic 3DGS streaming. First, we implement a TCP socket-based sender-receiver pipeline. The sender transmits the initial 3DGS scene, the NTC configuration, and compressed NTC checkpoints. Following 3DGStream [4], the initial scene stores the base Gaussian state, while each NTC provides deformation updates over time. The receiver reconstructs the scene cache and renders the dynamic 3DGS sequence.

Second, we demonstrate NTC compression by transmitting checkpoints in FP16, INT8, or INT4 format. FP16 uses half-precision parameters, while INT8 and INT4 use post-training blockwise quantization with scale metadata. The receiver dequantizes the compressed checkpoints before applying them to the cached 3DGS scene. These modes reduce update bandwidth while allowing different quality–bandwidth tradeoffs.

Third, we demonstrate sparse NTC streaming by transmitting only a subset of NTC updates, such as one update every three or five timesteps. For skipped timesteps, the receiver reuses the latest available NTC. A small cache-ahead buffer helps prevent playback stalls, while late updates are discarded. This reduces temporal bandwidth while maintaining interactive rendering and camera motion.

Acknowledgment

This work was supported by the R&D and Innovation Center, Ericsson Ltda, and by the São Paulo Research Foundation - FAPESP (grant 2021/00199-8), CPE SMARTNESS. The research activity of J. Almeida was supported by the São Paulo Research Foundation - FAPESP (grants 2023/17577-0 and 2024/22985-3) and the Brazilian National Council for Scientific and Technological Development - CNPq (grants 315220/2023-6 and 420442/2023-5).

References

[1] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. 2024. HAC: Hash-grid Assisted Context for 3D Gaussian Splatting Compression. In *European Conference on Computer Vision (ECCV)*. 422–438. doi:10.1007/978-3-031-72667-5_24

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4, Article 139 (July 2023), 14 pages. doi:10.1145/3592433

[3] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. 2022. Streaming Radiance Fields for 3D Video Synthesis. In *Advances in Neural Information Processing Systems*, Vol. 35. 13485–13498.

[4] Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 2024. 3DGStream: On-the-Fly Training of 3D Gaussians for Efficient Streaming of Photo-Realistic Free-Viewpoint Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20675–20685.

[5] Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. 2023. Neural Residual Radiance Fields for Streamably Free-Viewpoint Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 76–87.

[6] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 2024. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20310–20320.