MEGA: Memory-Efficient 4D Gaussian Splatting for Dynamic Scenes

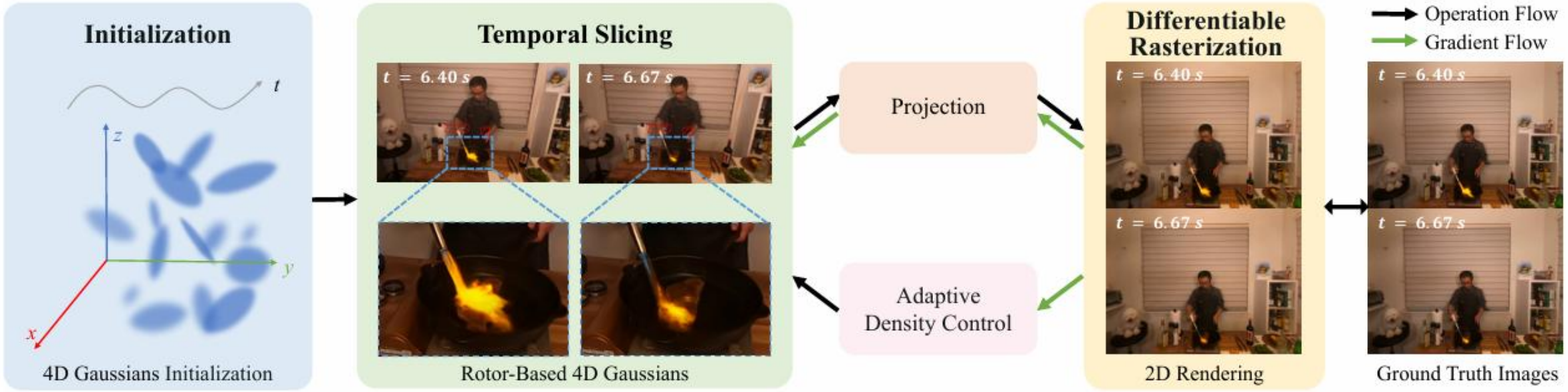
Xinjie Zhang, Zhening Liu, Yifan Zhang, Xingtong Ge, Dailan He,
Tongda Xu, Yan Wang, Zehong Lin, Shuicheng Yan, Jun Zhang

香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Background: 4D Gaussian Splatting

4DGS models a dynamic scene:

- Temporal Slicing to get time-related 3D Gaussians
- Projection (3D Gaussian → 2D Gaussian)
- Rasterization



THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

[1] Zeyu Yang, et al. Real-time Photorealistic Dynamic Scene Representation and Rendering with 4D Gaussian Splatting. ICLR, 2024.
[2] Yuanxing Duan, et al. 4d rotor gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. SIGGRAPH, 2024.

Challenge: 4D Gaussian Splatting

When rendering a dynamic scene, 4DGS leads to substantial storage overhead.

- 4D Gaussian Representation:
  - Each 4D Gaussian has 161 parameters
  - 4D spherical harmonics (SH) coefficients: 144 parameters
  - 4DSH: highly redundant

- 4DGS requires millions of Gaussians.
  - Birthday scene: 13 million Gaussian points
  - Linear motion modeling
  - Low Gaussians participation ratio: only 6% of Gaussians actively participate in rendering at each time step

PSNR: 31.00dB, Mem: 7.79GB, N_G:13M

Challenge: 4D Gaussian Splatting

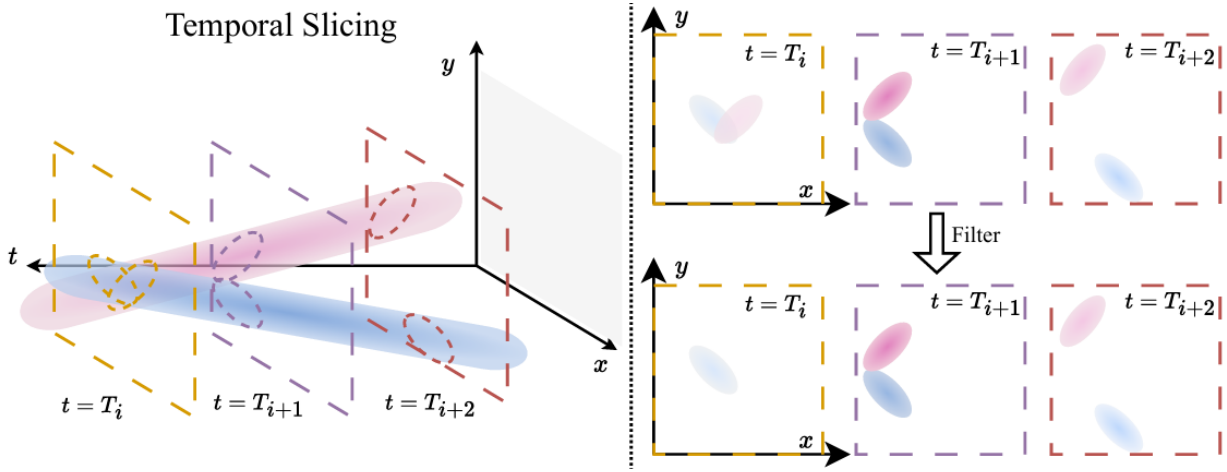
- Linear motion modeling
  - Each sliced 4D Gaussian with constant covariance exhibits only linear movement over time
  - Complex motion in the scene is modeled by a combination of multiple Gaussians.

$$G_{3D}(\mathbf{x}, t) = \mathrm{e}^{-\frac{1}{2}\lambda(t-\mu_t)^2}\mathrm{e}^{-\frac{1}{2}[\mathbf{x}-\boldsymbol{\mu}(t)]^T\Sigma_{3D}^{-1}[\mathbf{x}-\boldsymbol{\mu}(t)]},$$

$$\lambda = \mathbf{W}^{-1}, \quad \Sigma_{3D} = \mathbf{A}^{-1} = \mathbf{U} - \frac{\mathbf{V}\mathbf{V}^T}{\mathbf{W}},$$

$$\boldsymbol{\mu}(t) = (\mu_x, \mu_y, \mu_z)^T + (t - \mu_t)\frac{\mathbf{V}}{\mathbf{W}}.$$

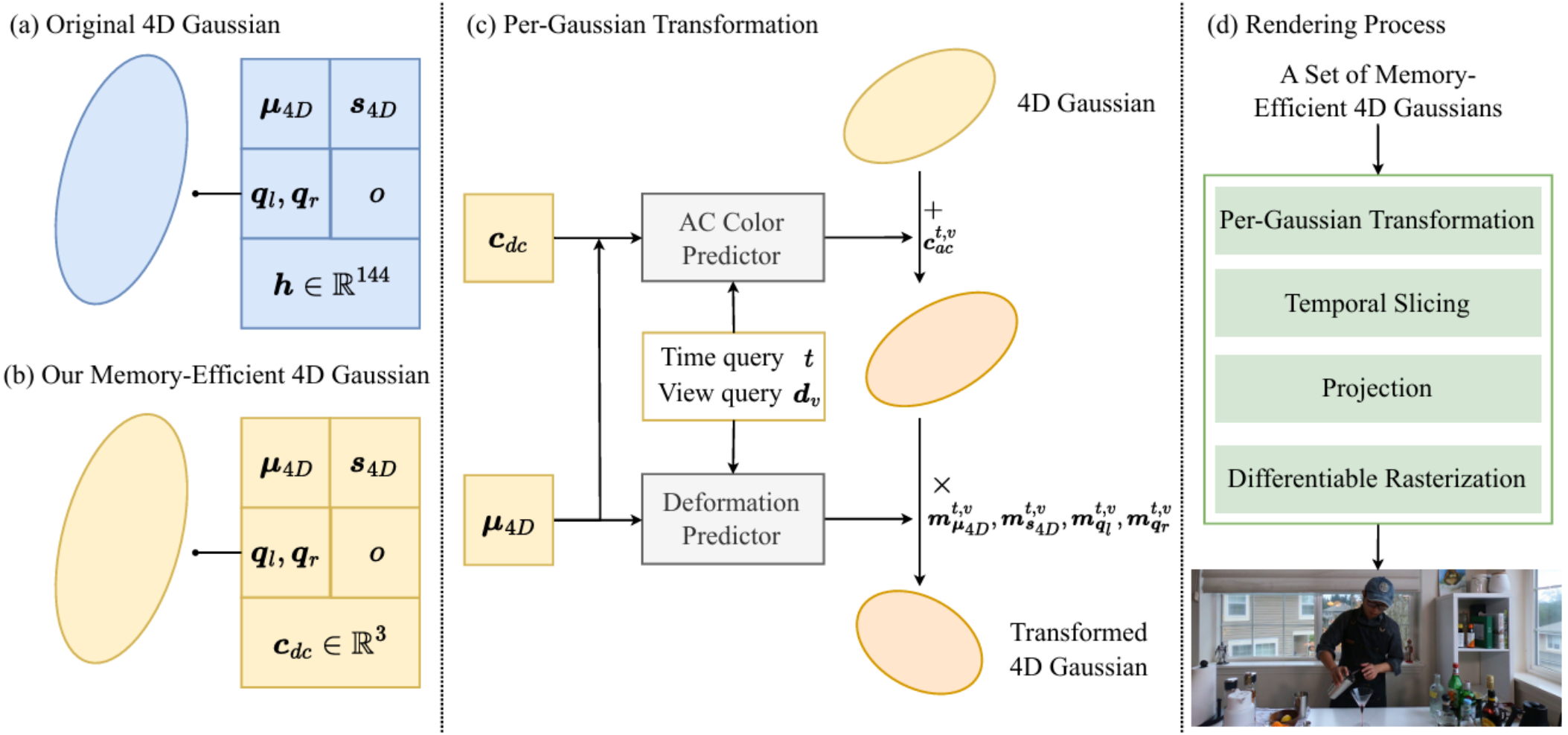
- Low Gaussians participation ratio
  - Temporal decay opacity forces each Gaussian to be visible only near its mean time center and invisible at other times.
  - Limit the effective utilization of each Gaussian



THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

MEGA: Overview

- Memory-efficient 4D Gaussian
- Entropy Constrained Gaussian Deformation



MEGA: Memory-efficient 4D Gaussian

- Memory-efficient 4D Gaussian:
  - Compact DC-AC color (DAC) representation
  - Decouple the color attribute as a per-Gaussian DC color component $\boldsymbol{c}_{dc} \in \mathbb{R}^3$ and a lightweight temporal viewpoint aware AC color predictor $\mathcal{F}_{\phi}$

$$\boldsymbol{c}_{t,v} = \text{sigmoid}(\boldsymbol{c}_{dc} + \mathcal{F}_{\boldsymbol{\phi}}(\text{sg}(\boldsymbol{\mu}_{3D}), \text{sg}(\boldsymbol{d}_v), t, \boldsymbol{c}_{dc})),$$

  - $\boldsymbol{\mu}_{3D}$: 3D center position of Gaussian; $\boldsymbol{d}_v$: normalized view direction

  - Benefits:
    - Preserves the individual information using DC component and supplements the missing viewpoint and time information using the AC predictor to maintain high rendering quality
    - Reduces the storage overhead by up to about 8× compared to the original 4DGS

MEGA: Entropy-Constrained Gaussian Deformation

- Gaussian Deformation:
  - Introduce a temporal-viewpoint aware deformation predictor to enlarge the action range of Gaussians.

$$(\boldsymbol{m}^{t,v}_{\boldsymbol{\mu}_{4D}}, \boldsymbol{m}^{t,v}_{\boldsymbol{s}_{4D}}, \boldsymbol{m}^{t,v}_{\boldsymbol{q}_l}, \boldsymbol{m}^{t,v}_{\boldsymbol{q}_r}) = \mathcal{F}_{\boldsymbol{\theta}}(\gamma(\mathrm{sg}(\boldsymbol{\mu}_{4D})), \gamma(\mathrm{sg}(\boldsymbol{d}_v)), \gamma(t)),$$

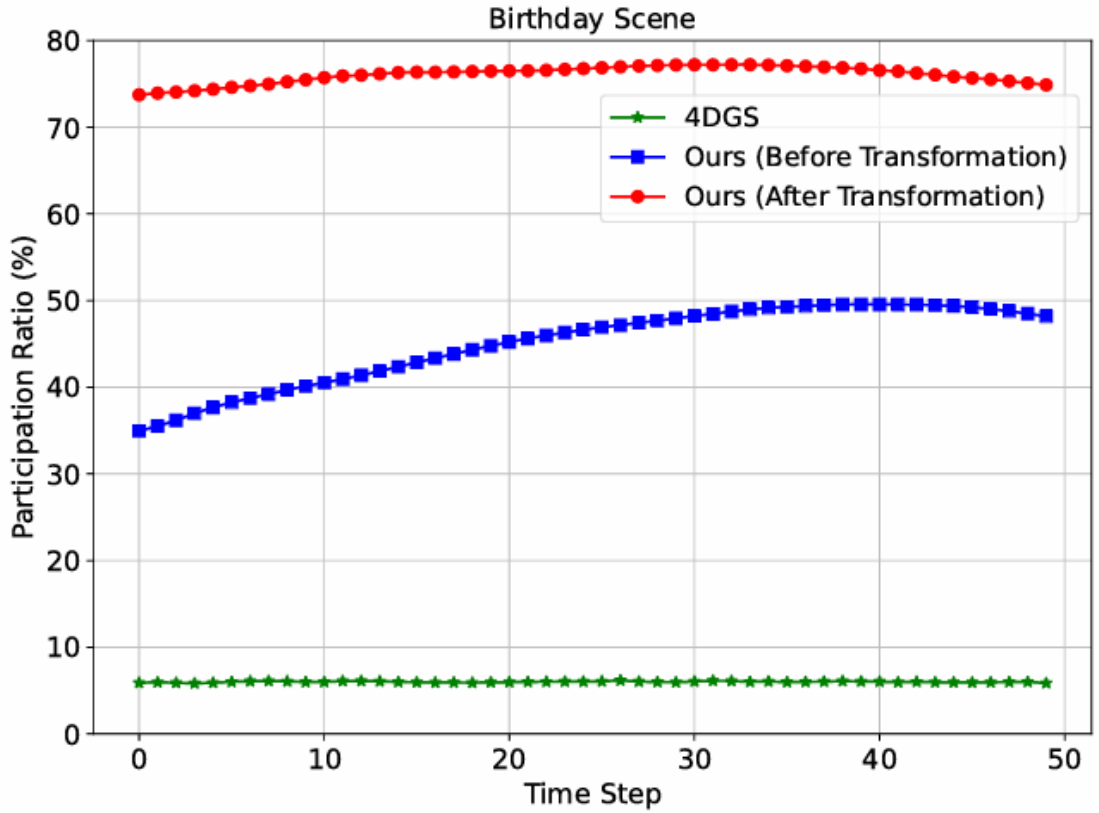
$$\boldsymbol{\mu}^{t,v}_{4D} = \boldsymbol{\mu}_{4D} \times \boldsymbol{m}^{t,v}_{\boldsymbol{\mu}_{4D}}, \quad \boldsymbol{s}^{t,v}_{4D} = \boldsymbol{s}_{4D} \times \boldsymbol{m}^{t,v}_{\boldsymbol{s}_{4D}}, \quad \boldsymbol{q}^{t,v}_{l} = \boldsymbol{q}_{l} \times \boldsymbol{m}^{t,v}_{\boldsymbol{q}_l}, \quad \boldsymbol{q}^{t,v}_{r} = \boldsymbol{q}_{r} \times \boldsymbol{m}^{t,v}_{\boldsymbol{q}_r}.$$

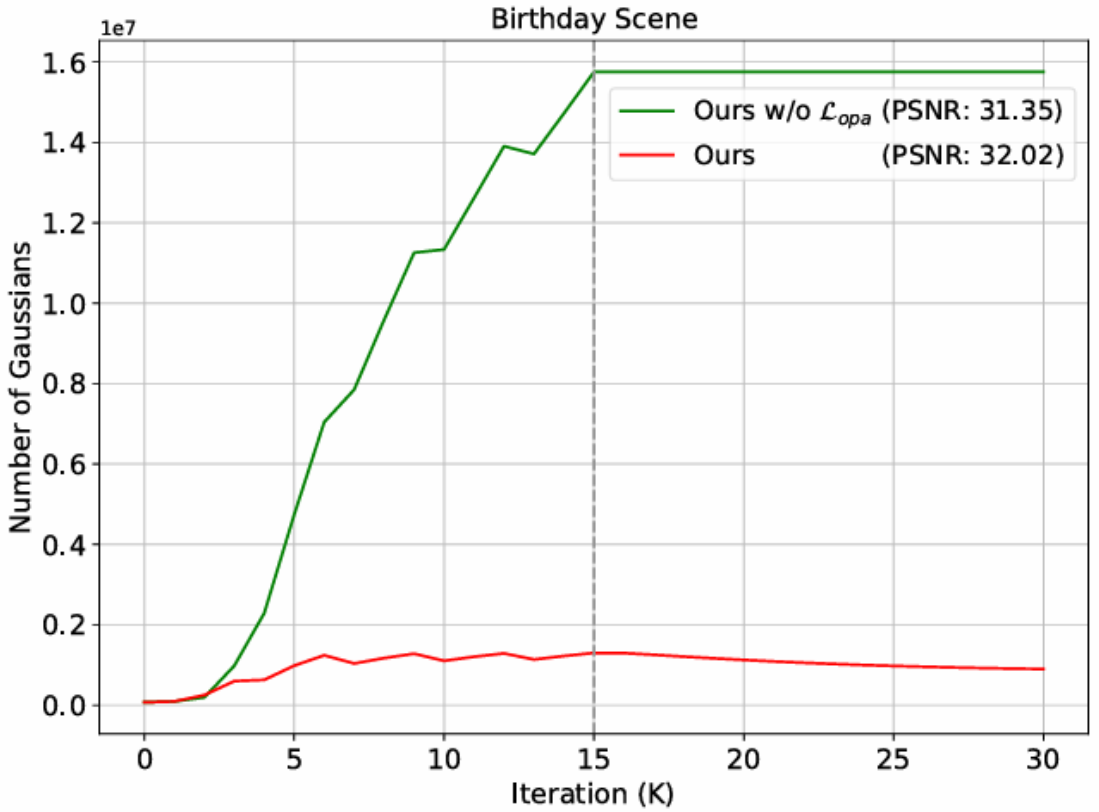
  - Improve flexibility in the motion representation and geometric structure of each 4D Gaussian
  - However, without constraints on the number of Gaussians, a significant proliferation occurs where Gaussians are continuously split and cloned during the densification process

MEGA: Entropy-Constrained Gaussian Deformation

- Entropy-constrained opacity loss:
  - $\mathcal{L}_{opa}$ encourages the spatial opacity of each Gaussian to approach one or zero

$$\mathcal{L}_{opa} = \frac{1}{N}\sum_{j=1}^{N}(-o_j \log(o_j))$$





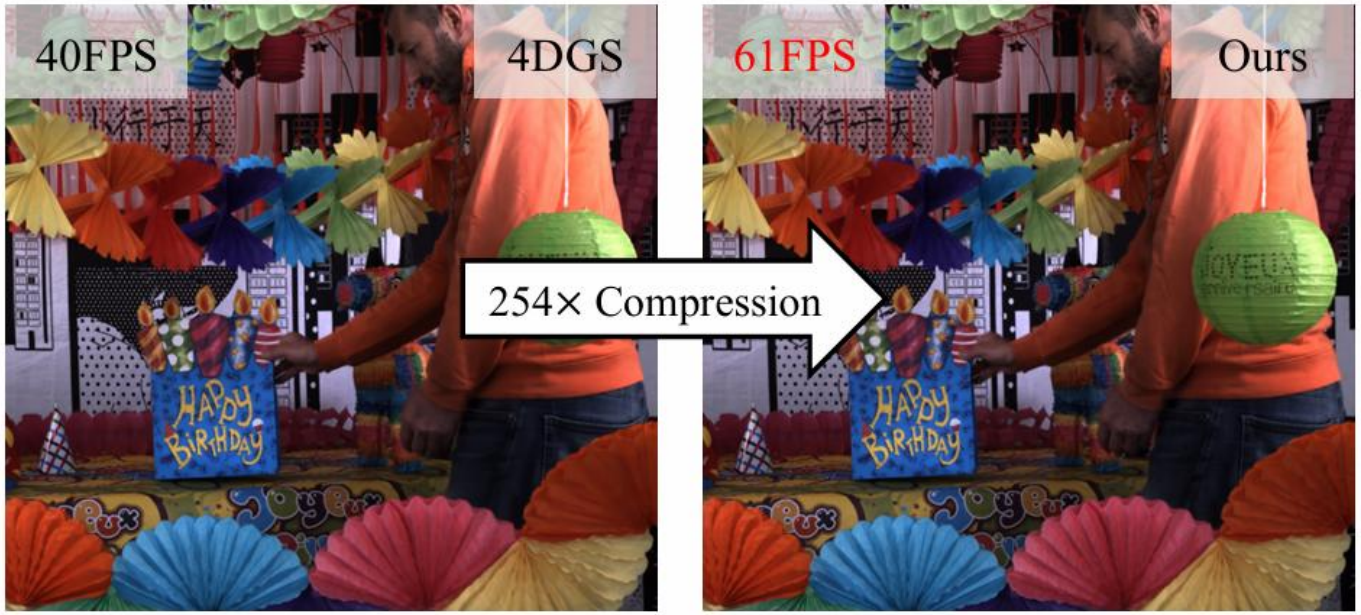
MEGA: Training and Compression Pipeline

- Loss function:

$$\mathcal{L} = \mathcal{L}_{photo} + \kappa\mathcal{L}_{opa} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{ssim} + \kappa\mathcal{L}_{opa},$$

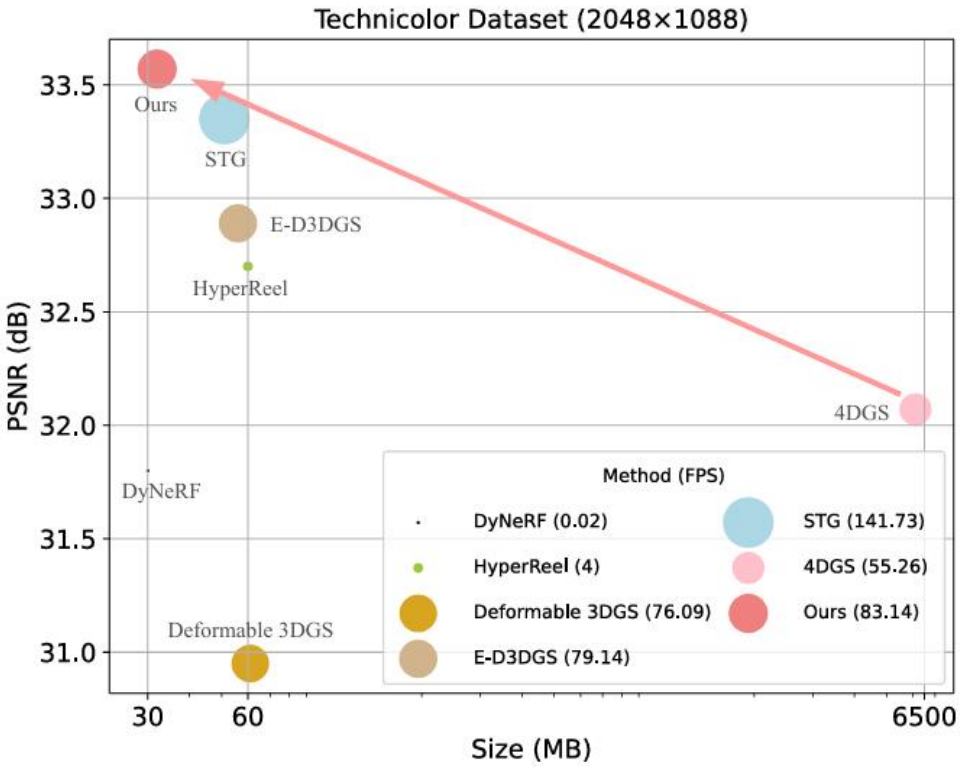
- Compression pipeline:
  - Half-precision training → Half-precision storage + ZIP DELTA lossless algorithm
  - This lossless compression technique typically reduces storage overhead by about 10%

Experimental Results: Technicolor Dataset



PSNR: 31.00dB, Mem: 7.79GB

PSNR: 32.02dB, Mem: 31.42MB



Experimental Results: Technicolor Dataset

Table 1: Quantitative comparison with various competitive baselines on the Technicolor Dataset. "Storage" refers to the total model size for 50 frames.

Method	PSNR↑	$DSSIM_1$↓	$DSSIM_2$↓	LPIPS↓	FPS↑	Storage↓
DyNeRF	31.80	-	0.0210	0.1400	0.02	30.00MB
HyperReel	32.70	0.0470	-	0.1090	4.00	60.00MB
Deformable 3DGS	30.95	0.0696	0.0353	0.1553	76.09	61.36MB
STG	33.35	0.0404	0.0187	0.0846	141.73	51.35MB
E-D3DGS	32.89	0.0494	0.0231	0.1114	79.14	56.07MB
4DGS	32.07	0.0535	0.0263	0.1189	55.26	6107.07MB
Ours	33.57	0.0442	0.0204	0.1014	83.14	32.45MB

Experimental Results: Neural 3D Video Dataset

Table 2: Quantitative comparisons with various competitive baselines on the Neural 3D Video Dataset. "Storage" refers to the total model size for 300 frames. 1: Only report the result on the *Flame Salmon* scene. 2: Exclude the *Coffee Martini* scene. 3: These methods train each model with a 50-frame video sequence to prevent memory overflow, requiring six models to complete the overall evaluation.

Method	PSNR↑	$DSSIM_1$↓	$DSSIM_2$↓	LPIPS↓	FPS↑	Storage↓
Neural Volume[1]	22.80	-	0.0620	0.2950	-	-
LLFF[1]	23.24	-	0.0200	0.2350	-	-
DyNeRF[1]	29.58	-	0.0200	0.0830	0.015	28.00MB
HexPlane[2,3]	31.71	-	0.0140	0.0750	0.56	200.00MB
StreamRF	28.26	-	-	-	10.90	5310.00MB
NeRFPlayer[3]	30.69	0.0340	-	0.1110	0.05	5130.00MB
HyperReel	31.10	0.0360	-	0.0960	2.00	360.00MB
K-Planes	31.63	-	0.0180	-	0.30	311.00MB
MixVoxels-L	31.34	-	0.0170	0.0960	37.70	500.00MB
MixVoxels-X	31.73	-	0.0150	0.0640	4.60	500.00MB
Dynamic 3DGS	30.46	0.0350	0.0190	0.0990	460.00	2772.00MB
C-D3DGS	30.46	-	-	0.1500	118.00	338.00MB
Deformable 3DGS	30.98	0.0331	0.0191	0.0594	29.62	32.64MB
E-D3DGS	31.20	0.0259	0.0151	0.0304	69.70	40.20MB
STG[3]	32.04	0.0261	0.0145	0.0440	273.47	175.35MB
4DGS	31.57	0.0290	0.0164	0.0573	96.69	3128.00MB
Ours	31.49	0.0290	0.0165	0.0568	77.42	25.05MB

THE HONG KONG
UNIVERSITY OF S
AND TECHNOLO

Ablation Study

Table 3. Ablation study of the proposed components. N denotes the number of Gaussians. The last row represents our final solution.

(a) Technicolor Dataset

Variants	*Birthday*				*Fabien*			
	PSNR↑	DSSIM$_1$↓	N ↓	Params↓	PSNR↑	DSSIM$_1$↓	N ↓	Params↓
4DGS [46]	31.00	0.0383	13.00M	2093.56M	33.57	0.0582	5.43M	874.14M
w/ grid [20]	30.49	0.0410	16.33M	293.07M	32.99	0.0620	4.61M	93.77M
w/ DAC	31.60	0.0355	15.43M	308.65M	34.21	0.0587	4.57M	91.48M
w/ DAC+Deformation	31.35	0.0368	15.75M	315.36M	33.02	0.0604	11.56M	231.53M
w/ DAC+$\mathcal{L}_{opa}$	31.46	0.0370	9.15M	183.23M	33.96	0.0603	2.32M	46.40M
w/ DAC+Deformation+$\mathcal{L}_{opa}$	32.02	0.0309	0.91M	18.48M	34.89	0.0597	0.31M	6.43M

(b) Neural 3D Video Dataset

Variants	*Flame Steak*				*Sear Steak*			
	PSNR↑	DSSIM$_1$↓	N ↓	Params↓	PSNR↑	DSSIM$_1$↓	N ↓	Params↓
4DGS [46]	33.19	0.0204	5.17M	831.88M	33.44	0.0204	3.52M	567.30M
w/ grid [20]	31.07	0.0279	4.82M	97.35M	31.313	0.0281	3.25M	70.76M
w/ DAC	33.34	0.0210	5.31M	106.33M	33.67	0.0206	3.61M	72.18M
w/ DAC+Deformation	33.47	0.0209	6.34M	127.16M	33.46	0.0208	4.17M	83.78M
w/ DAC+$\mathcal{L}_{opa}$	33.45	0.0208	2.76M	55.22M	33.58	0.0215	1.99M	39.74M
w/ DAC+Deformation+$\mathcal{L}_{opa}$	32.27	0.0242	0.87M	17.79M	33.67	0.0200	0.56M	11.50M

Summary

- We are among the first to develop a memory-efficient framework for 4D Gaussian Splatting.

- Propose compact DC-AC color representation:
  - Significantly reduce the per-Gaussian parameters without compromising performance.

- Introduce entropy-constrained Gaussian deformation:
  - Expand the action range of each Gaussian to enhance the effective utilization rate
  - Enable the model to render high-quality scenes with as few Gaussians as possible.

Thank you!

香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY