上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

SJTU

Enhance Efficiency: 3D Gaussian Splatting for Speed and Memory Optimization

黄霄童

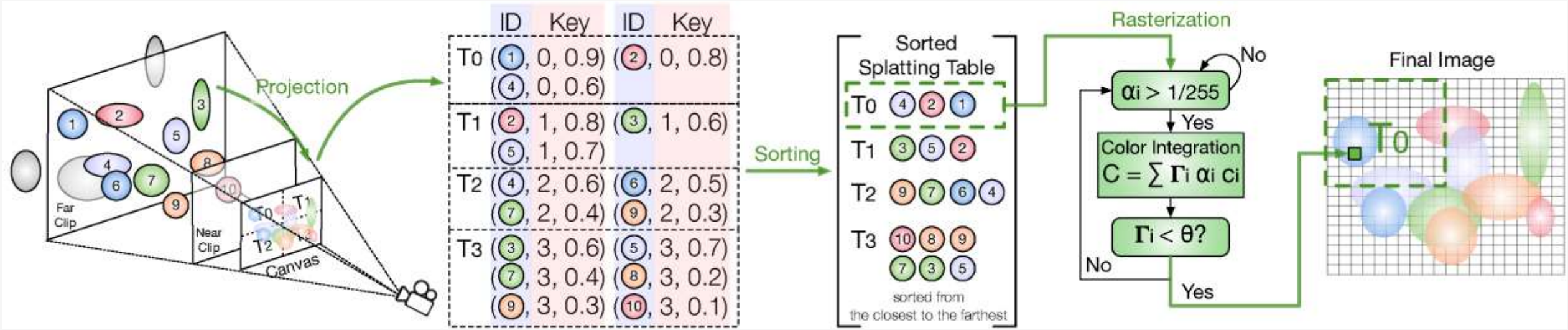
2024.12.06

饮 水 思 源 • 爱 国 荣 校

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

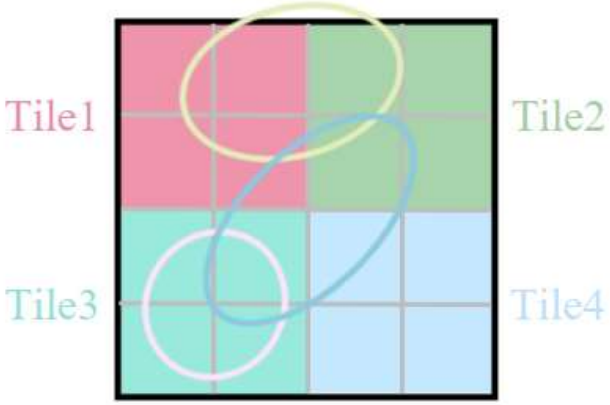
目录

3DGS Pipeline

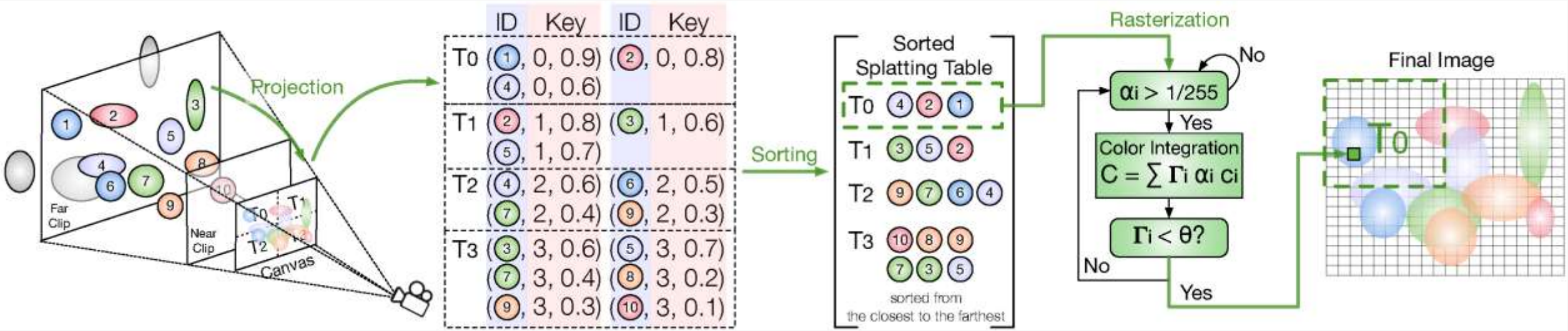


Step 1: Project 3D Gaussians into image space and replicate the Gaussians which cover several tiles.

$$\Sigma' = JW\Sigma W^T J^T$$

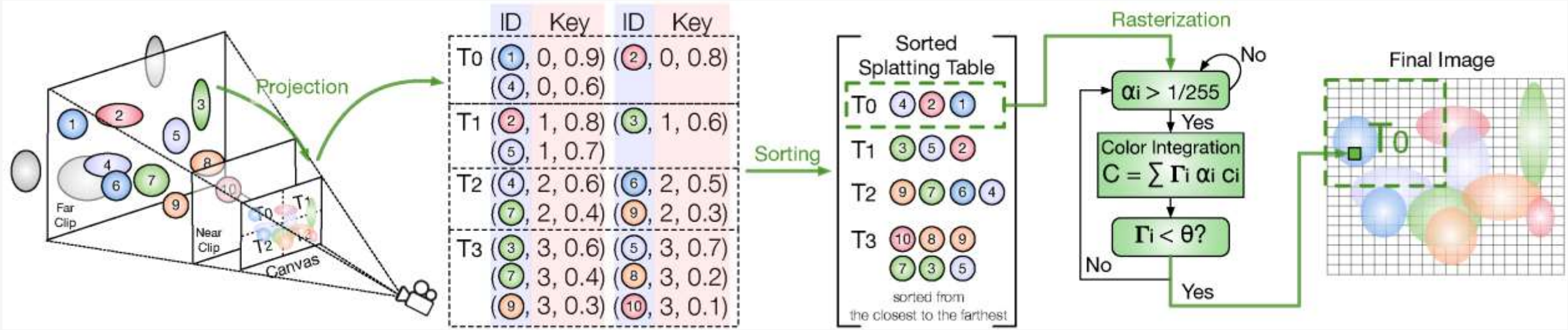


3DGS Pipeline



Step 2: Sort the intersected points based on their depth values, from the nearest to the farthest.

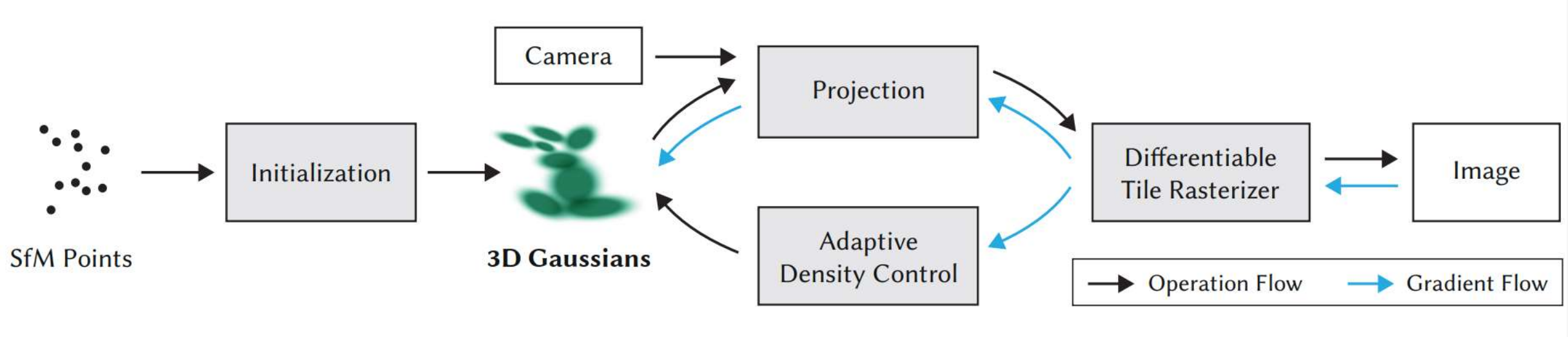
3DGS Pipeline



Step 3: Within each tile, render the Gaussians in the sorted order based on the radiance accumulation equation to obtain all pixels.

$$C = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i \quad \text{with} \quad T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right),$$

3DGS Optimization



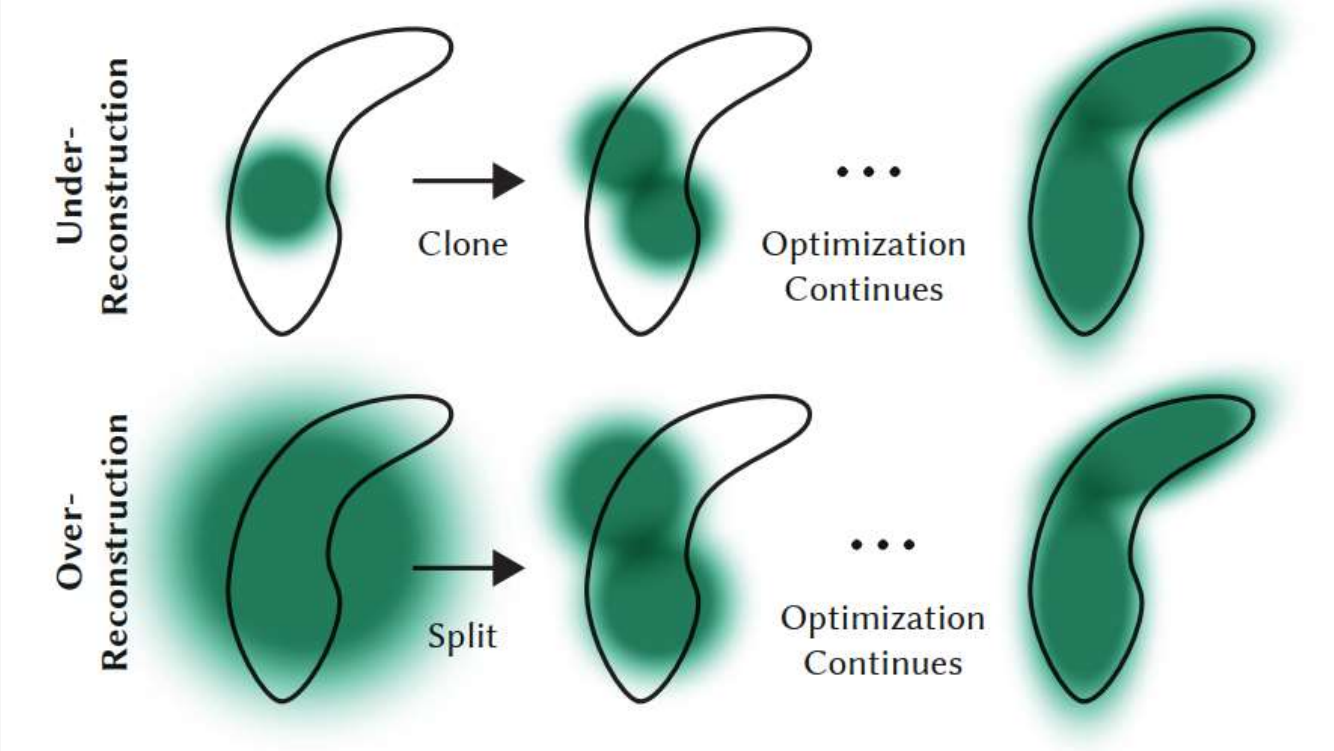
- Optimization starts with the sparse SfM point cloud and creates a set of 3D Gaussians.
- By comparing the rendering image to the training views, optimize the properties of the Gaussians and adaptively control the density of this set of Gaussians.

3DGS Optimization

Densification:

- For **cloning**, a copy of the Gaussian is created and moved towards the positional gradient.

- For **splitting**, a large Gaussian is replaced with two smaller ones, reducing their scale by a specific factor.



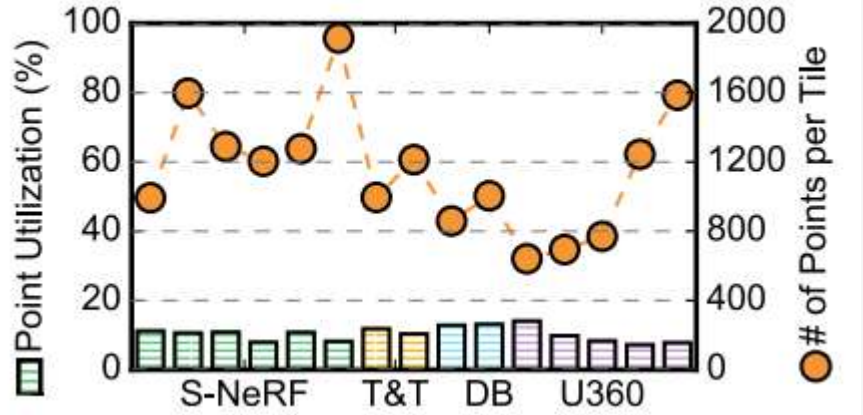
3DGS Optimization

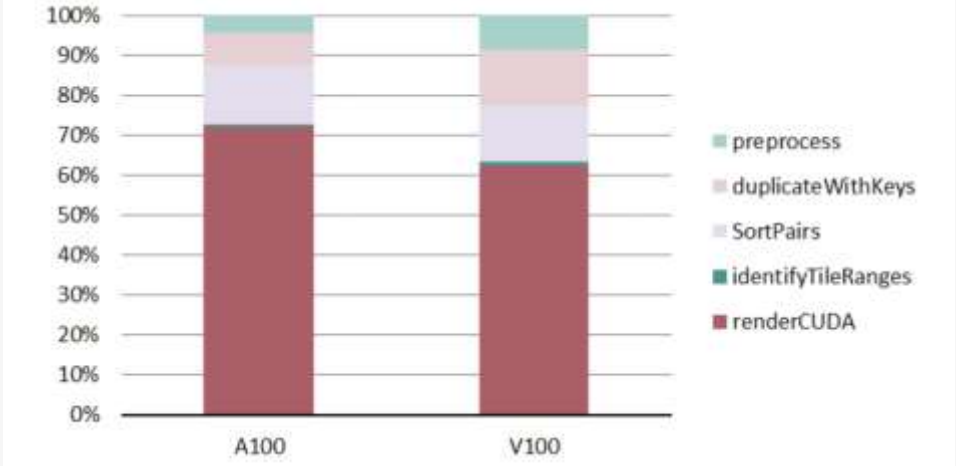
Pruning:

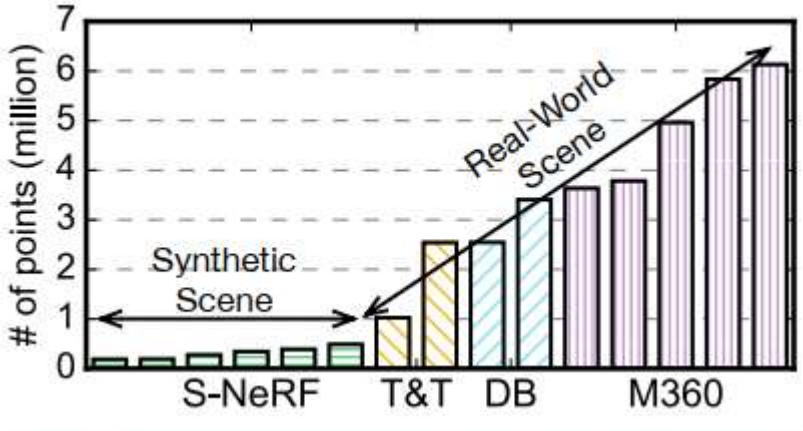
- Eliminate Gaussians that are virtually **transparent** (with α below a specified threshold) and those that are **excessively large** in either world-space or view-space.
- In Gaussian density **near input cameras**, the alpha value of the Gaussians is set close to zero after a certain number of iterations.

3DGS Challenges

- A large number of Gaussians, but only a small fraction of them are truly useful.







3DGS Challenges

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

- A large number of Gaussians, but only a small fraction of them are truly useful.
- **High-degree spherical harmonics (SH) coefficients. (16 * 3 for a splat)**

Table 1. Parameters for 3D Gaussian.

Parameter	Symbol	Size	Note
Position (mean)	μ	3	3D vector (x, y, z)
Scale	s	3	3D vector (x, y, z)
Rotation (quaternion)	q	4	scalar + 3D vector (i, j, k)
Opacity	o	1	scalar
SH coefficients	sh	48	4 bands of SH; (1+3+5+7)×3
Total (per Gaussian)		59	

Memory	4GB 64-bit LPDDR4 25.6GB/s
Storage	16GB eMMC 5.1
Video Encode	1x 4K30 (H.265) 2x 1080p60 (H.265)
Video Decode	1x 4K60 (H.265) 4x 1080p60 (H.265)
CSI Camera	Up to 4 cameras 12 lanes MIPI CSI-2 D-PHY 1.1 (up to 18 Gbps)
PCIe*	1 x4 (PCIe Gen2)
USB*	1x USB 3.0 (5 Gbps) 3x USB 2.0

Scene	bicycle			bonsai			drjohnson			playroom		
Method	PSNR	Storage	Mem.	PSNR	Storage	Mem.	PSNR	Storage	Mem.	PSNR	Storage	Mem.
3DGS	25.08	1.4 GB	9.4 GB	32.16	295 MB	8.7 GB	29.06	774 MB	7.5 GB	29.87	553 MB	6.4 GB

SJTU

3DGS Techniques: LightGuassian

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

LightGaussian

1. Gaussian Pruning and Recovery.
2. SH Distillation.
3. Vector Quantization.

Exp#	Model	FPS↑	Size↓	PSNR↑	SSIM↑	LPIPS↓
[1]	Baseline (3D-GS [32])	192.05	353MB	31.68	0.926	0.200
[2]	+ Gaussian Pruning.	312.30	116MB	30.32	0.911	0.222
[3]	+ Co-adaptation	303.99	116MB	31.85	0.925	0.206
[4]	+ SH Compactness.	318.97	77MB	30.54	0.914	0.217
[5]	+ Distillation	304.20	77MB	31.47	0.922	0.211
[6]	+ Pseudo-views	300.60	77MB	31.59	0.923	0.211
[7]	+ Codebook Quant.	300.60	20MB	31.16	0.915	0.218
[8]	+ VQ finetune	300.60	20MB	31.67	0.923	0.212
[9]	LightGaussian (Ours)	300.60	20MB	31.67	0.923	0.212

3DGS Techniques: LightGuassian

Gaussian Pruning and Recovery:

$$\mathrm{GS_j} = \sum_{i=1}^{MHW} \mathbb{1}(\mathrm{G}(\boldsymbol{X}_j), r_i) \cdot \sigma_j \cdot \gamma(\boldsymbol{\Sigma}_j), \tag{3}$$

where j is the Gaussian index, i means a pixel, M, H, and W represents the number of training views, image height, and width, respectively. $\mathbb{1}$ is the indicator function that determines whether a Gaussian intersects with a given ray.

$$\gamma(\boldsymbol{\Sigma}) = (\mathrm{V_{norm}})^{\beta}, \tag{4}$$

$$\mathrm{V_{norm}} = \min\left(\max\left(\frac{\mathrm{V}(\boldsymbol{\Sigma})}{\mathrm{V_{max90}}}, 0\right), 1\right).$$

Here, the calculated Gaussian volume is firstly normalized by the 90% largest of all sorted Gaussians, clipping the range between 0 and 1, to avoid excessive floating Gaussians derived from vanilla 3D-GS. The β is introduced to provide additional flexibility.

3DGS Techniques: LightGuassian

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

SH distillation:

Distill the knowledge from the high-degree teacher model to the low-degree student model.

Note: Distillation involves training the Gaussians not only to represent **known views** but also to learn views from **unseen(pseudo) views**.

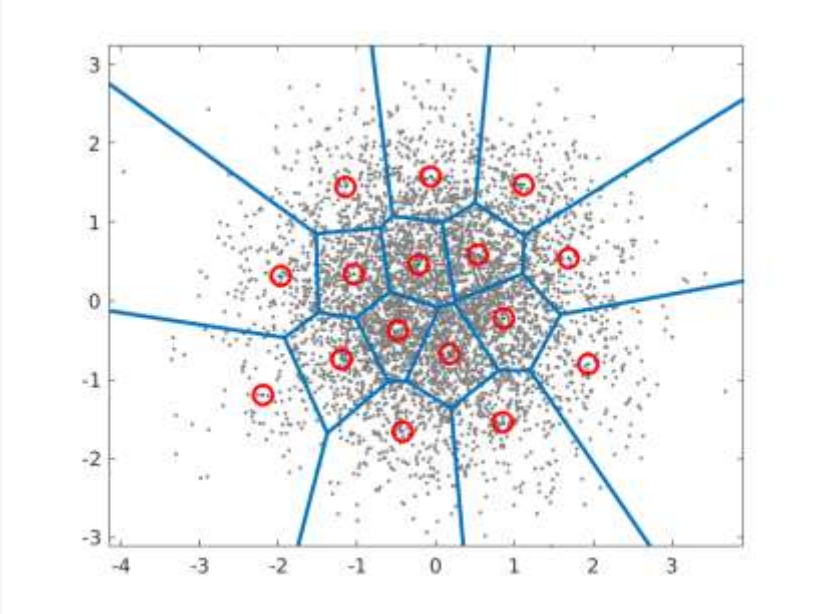
$$\mathcal{L}_{\text{distill}} = \frac{1}{HW} \sum_{i=1}^{HW} \| \boldsymbol{C}_{\text{teacher}}(r_i) - \boldsymbol{C}_{\text{student}}(r_i) \|_2^2 .$$

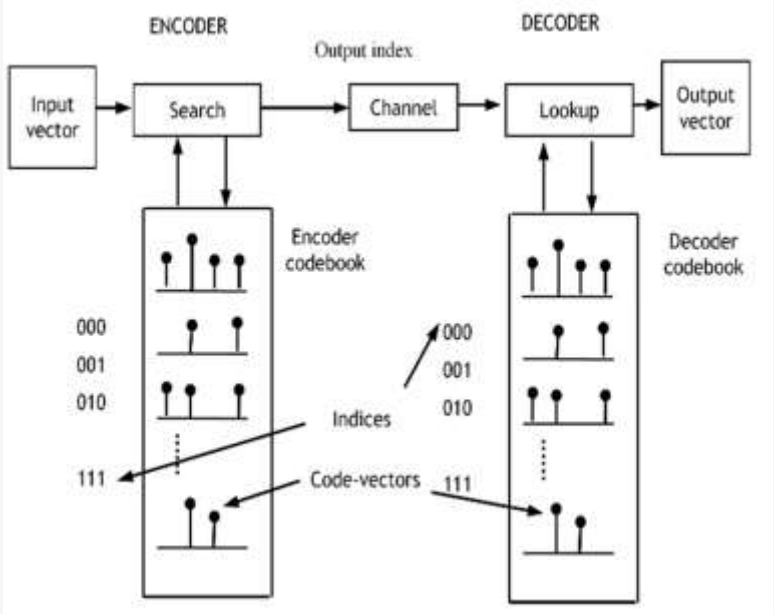
3DGS Techniques: LightGuassian

Vector quantization:

Assumption: that a subgroup of 3d gaussians typically exhibits a similar appearance.

1. Compress the **least significant elements** in the SHs,
2. but store Gaussian's position, shape, rotation, and opacity attributes in the float16 format.





3DGS Techniques: LightGuassian

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

LightGaussian

1. Gaussian Pruning and Recovery.
2. SH Distillation.
3. Vector Quantization.

Exp#	Model	FPS↑	Size↓	PSNR↑	SSIM↑	LPIPS↓
[1]	Baseline (3D-GS [32])	192.05	353MB	31.68	0.926	0.200
[2]	+ Gaussian Pruning.	312.30	116MB	30.32	0.911	0.222
[3]	+ Co-adaptation	303.99	116MB	31.85	0.925	0.206
[4]	+ SH Compactness.	318.97	77MB	30.54	0.914	0.217
[5]	+ Distillation	304.20	77MB	31.47	0.922	0.211
[6]	+ Pseudo-views	300.60	77MB	31.59	0.923	0.211
[7]	+ Codebook Quant.	300.60	20MB	31.16	0.915	0.218
[8]	+ VQ finetune	300.60	20MB	31.67	0.923	0.212
[9]	LightGaussian (Ours)	300.60	20MB	31.67	0.923	0.212

3DGS Techniques: EfficientGS

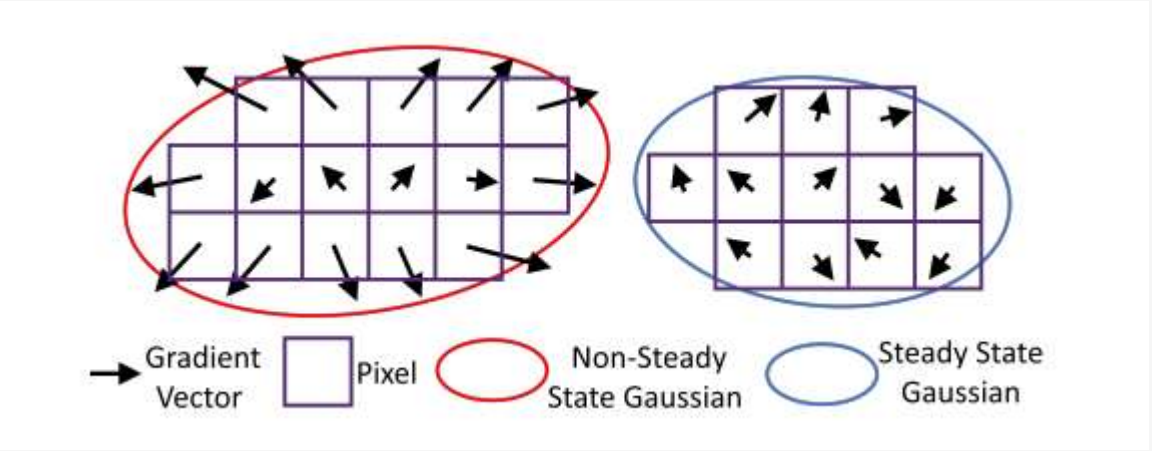
EfficientGS:

1. A selective densification strategy.
2. A pruning strategy.
3. A sparse SH order increment strategy.

3DGS Techniques: EfficientGS

1. A selective densification strategy.

They only clone and split non-steady Gaussians.



$$\mathbf{E}_g = \frac{\sum_{I \in M}^{M} (|\sum(\vec{\nabla}_{pI}^{g})|)}{|M|},$$

$$\mathbf{S}_g > \tau_s, \mathbf{S}_g = \frac{\sum_{I \in M}^{M} (\sum(|\vec{\nabla}_{pI}^{g}|))}{|M|},$$

3DGS Techniques: EfficientGS

2. A pruning strategy.

They calculate the weight of a Gaussian when contributing to the pixel p:

$$weight_p^i = \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), g_i \in \mathcal{N}_p.$$

$$C_p = \sum_{i \in \mathcal{N}_p} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j),$$

For each ray, they regard the Gaussians with top-K weights as dominant:

$$weight_p^g \geq sort(\{-weight_p^j, g_j \in \mathcal{N}_p\})[K].$$

As long as a Gaussian is not dominant for any certain pixel, the Gaussian will be deleted.

3DGS Techniques: EfficientGS

3. A sparse SH order increment strategy.

They selectively increase the SH order for only those Gaussians with large difference.

The view-based color difference is calculated as:

$$d_k = \sum_{p=1}^{NHW} \mathbb{1}(k \in \mathcal{N}_p) weight_p^k |C_p - C_p^{gt}|,$$

3DGS Techniques: EfficientGS

1. A selective densification strategy. (SD)
2. A pruning strategy. (GP)
3. A sparse SH order increment strategy. (SOI)

SD	GP	SOI	SSIM	PSNR	LPIPS	Train	FPS	Storage	Peak GPU↓
			0.812	27.48	0.222	25m57s	125	742MB	12.8G
✓			0.820	27.53	0.202	24m02s	140	544MB	11.0GB
✓	✓		0.821	27.48	0.209	20m42s	229	259MB	10.9GB
✓	✓	✓	0.817	27.38	0.216	19m41s	218	98MB	8.8GB

3DGS Techniques: EfficientGS

3DGS Techniques: GaussianSpa

Weakness of before methods:

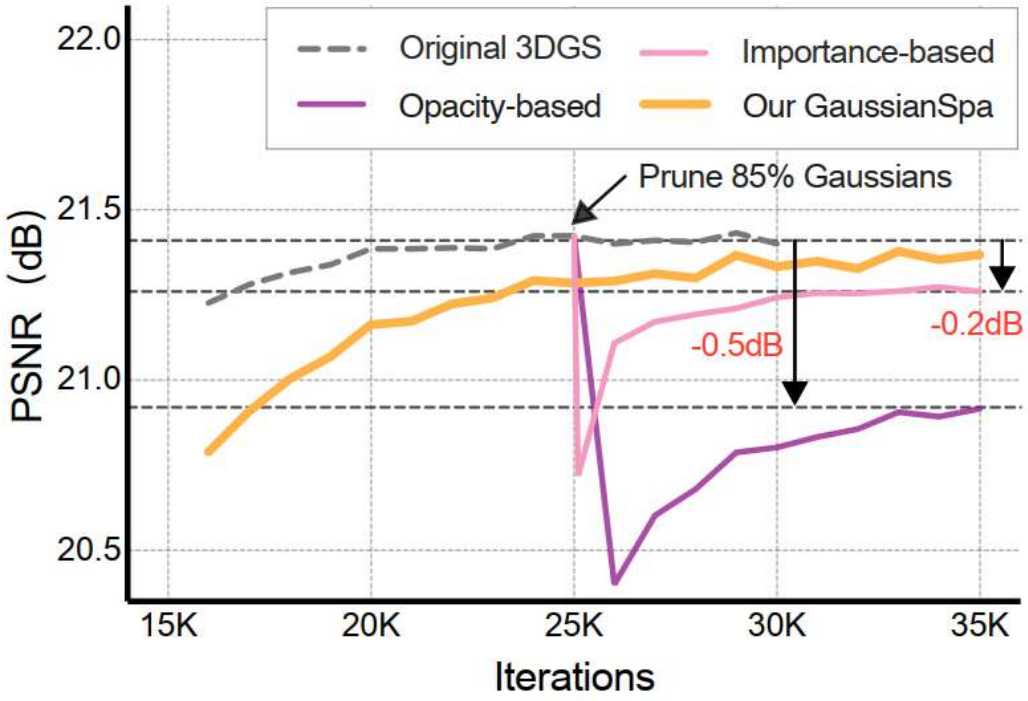


Figure 2. **PSNR curves of human-crafted criteria-based pruning methods.** All methods remove 85% of the Gaussians at iteration 25K.

3DGS Techniques: GaussianSpa

Innovations:

Formulate 3DGS simplification as a constrained optimization problem **under a target number of Gaussians.**

Optimizing Step:

$$\min_{\boldsymbol{a},\boldsymbol{\Theta}} \mathcal{L}(\boldsymbol{a},\boldsymbol{\Theta}) + \frac{\delta}{2}\|\boldsymbol{a}-\boldsymbol{z}+\boldsymbol{\lambda}\|^2.$$

Sparsifying Step:

$$\min_{\boldsymbol{z}} h(\boldsymbol{z}) + \frac{\delta}{2}\|\boldsymbol{a}-\boldsymbol{z}+\boldsymbol{\lambda}\|^2.$$

$$\boldsymbol{z} \leftarrow \mathbf{prox}_h(\boldsymbol{a}+\boldsymbol{\lambda}).$$

3DGS Techniques: GaussianSpa

Method	Mip-NeRF 360				Tanks&Temples				Deep Blending			
	PSNR↑	SSIM↑	LPIPS↓	#G/M↓	PSNR↑	SSIM↑	LPIPS↓	#G/M↓	PSNR↑	SSIM↑	LPIPS↓	#G/M↓
3DGS [25]	27.45	0.810	0.220	3.110	23.63	0.850	0.180	1.830	29.42	0.900	0.250	2.780
CompactGaussian [29]	27.08	0.798	0.247	1.388	23.32	0.831	0.201	0.836	29.79	0.901	0.258	1.060
LP-3DGS-R [54]	27.47	0.812	0.227	1.959	23.60	0.842	0.188	1.244	-	-	-	-
LP-3DGS-M [54]	27.12	0.805	0.239	1.866	23.41	0.834	0.198	1.116	-	-	-	-
EAGLES [22]	27.23	0.809	0.238	1.330	23.37	0.840	0.200	0.650	29.86	0.910	0.250	1.190
Mini-Splatting [20]	27.40	0.821	0.219	0.559	23.45	0.841	0.186	0.319	30.05	0.909	0.254	0.397
Taming 3DGS [34]	27.31	0.801	0.252	0.630	23.95	0.837	0.201	0.290	29.82	0.904	0.260	0.270
CompGS [38]	27.12	0.806	0.240	0.845	23.44	0.838	0.198	0.520	29.90	0.907	0.251	0.550
GaussianSpa	**27.85**	**0.825**	**0.214**	**0.547**	**23.98**	**0.852**	**0.180**	**0.269**	**30.37** 30.33	**0.914** 0.912	**0.249** 0.254	0.335 **0.256**

Method	PSNR↑	SSIM↑	LPIPS↓	Storage↓
EfficientGS [31]	27.38	0.817	0.216	98 MB
LightGaussian [19]	27.28	0.805	0.243	42 MB
GaussianSpa	**27.85**	**0.825**	**0.214**	**25 MB**

Table 2. **Storage comparison evaluated on the Mip-NeRF 360 dataset.** GaussianSpa's storage cost is reported based on the add-on compression methods (i.e., SH distillation and vector quantization) from LightGaussian [19].

3DGS Techniques: Summary

1. Remove a certain number of Gaussians:
    a. pruning and sampling aim to discard unimportant Gaussians by opacity, hit count, dominant primitives.
    b. formulate it as an optimization problem.

2. Avoid redundancy computations in duplicating gaussians, sorting, rendering….

3. Compression methods like VQ, distillation, mixed precision computation….

3DGS Exploration

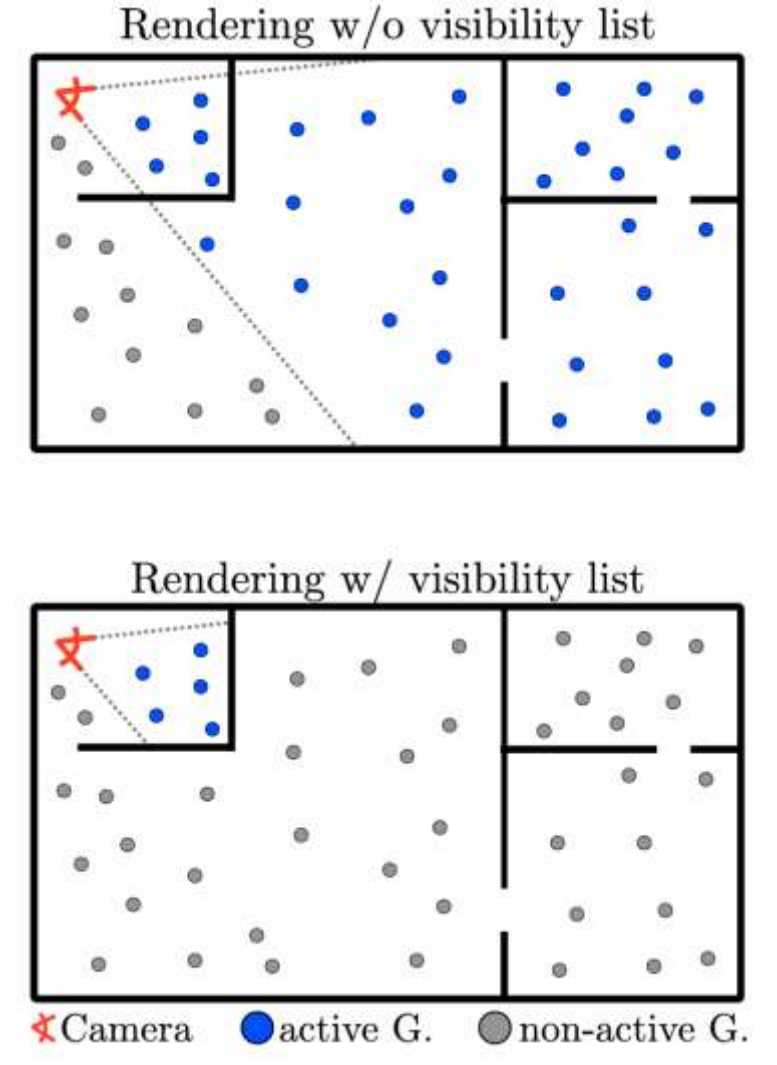
上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

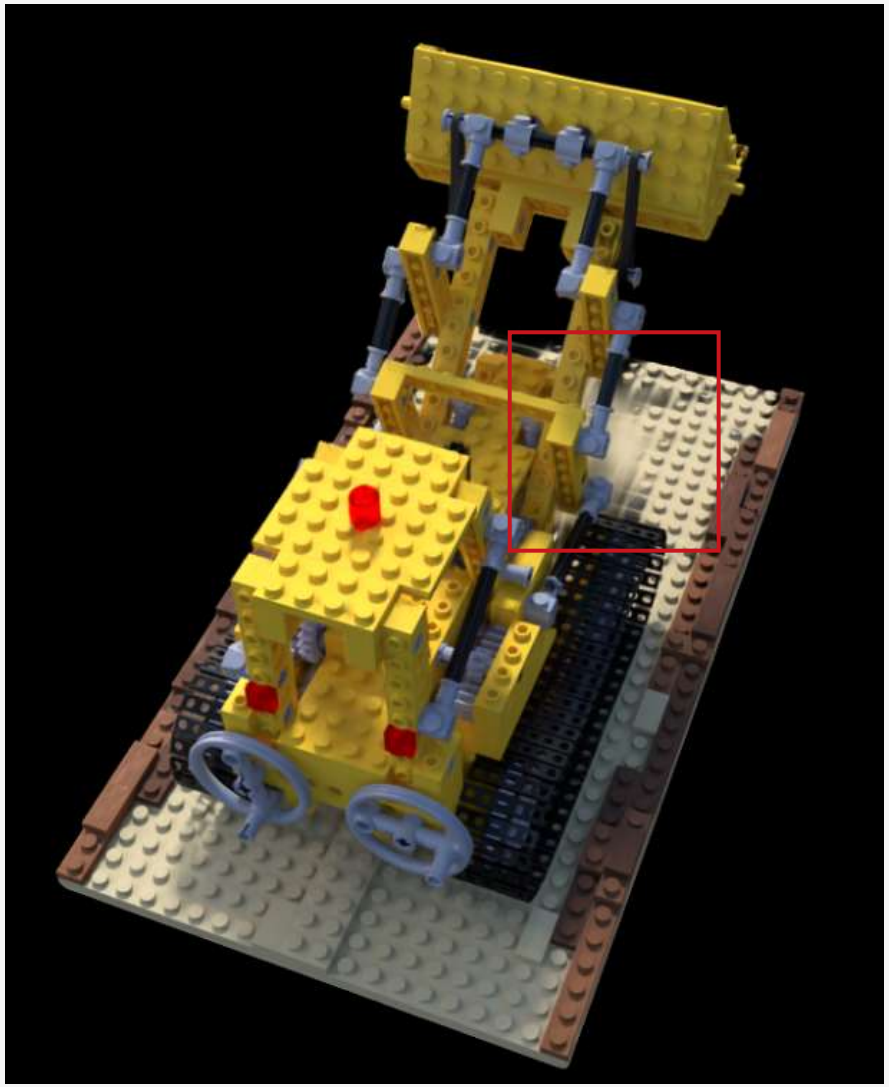
Observation:

Between two distant frames, the number of Gaussian points shared for rendering images is limited.

Frame	Ratio
0 & 9	0.77 | 0.81
8 & 9	0.97 | 0.96
17 & 9	0.71 | 0.61
10 & 0	0.75 | 0.75
20 & 0	0.40 | 0.36
30 & 0	0.22 | 0.14

3DGS Exploration





3DGS Techniques: Exploration

- How should the division into clusters be performed?

- When only a part of the Gaussian point cloud is loaded, is it possible to perform some fine-tuning? If so, how can we ensure the consistency of shared Gaussian points across different clusters?

- Can further optimizations be achieved by leveraging the division into clusters?

…

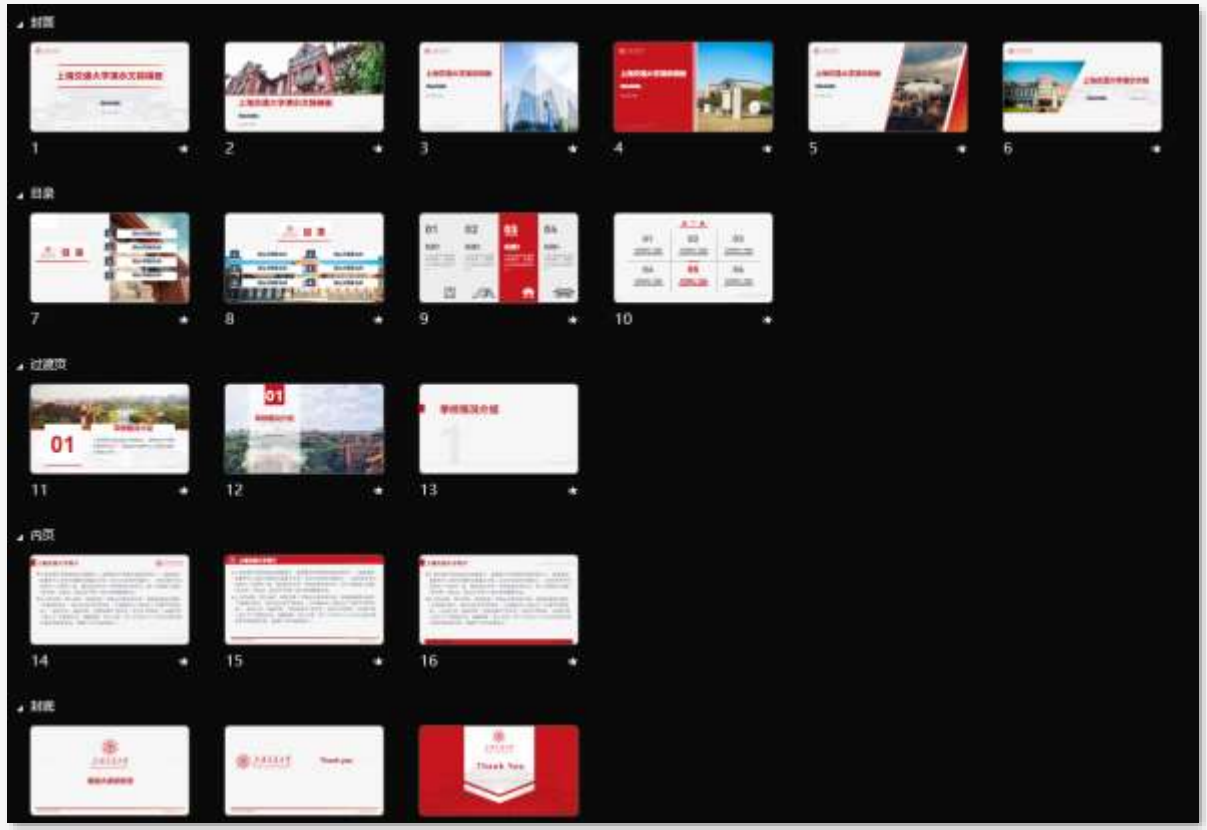
上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

Thank You

饮 水 思 源　爱 国 荣 校

版式类型



封面版式：6个

目录版式：4个

过渡页版式：3个

内页版式：3个

封底版式：3个

空白页：1个

字体与配色

标题：

华为鸿蒙字体（简体中文）-黑

HarmonyOS Sans SC Black

正文：

华为鸿蒙字体（简体中文）-轻

HarmonyOS Sans SC Light

说明：华为鸿蒙字体免费向社会开放使用，是鸿蒙操作系统的默认字体