浙江大學 ZHEJIANG UNIVERSITY

MONASH University

NEURAL INFORMATION PROCESSING SYSTEMS

ZPressor: Bottleneck-Aware Compression for Scalable Feed-Forward 3DGS


Weijie Wang[1]


Donny Y. Chen[2]


Zeyu Zhang[2]

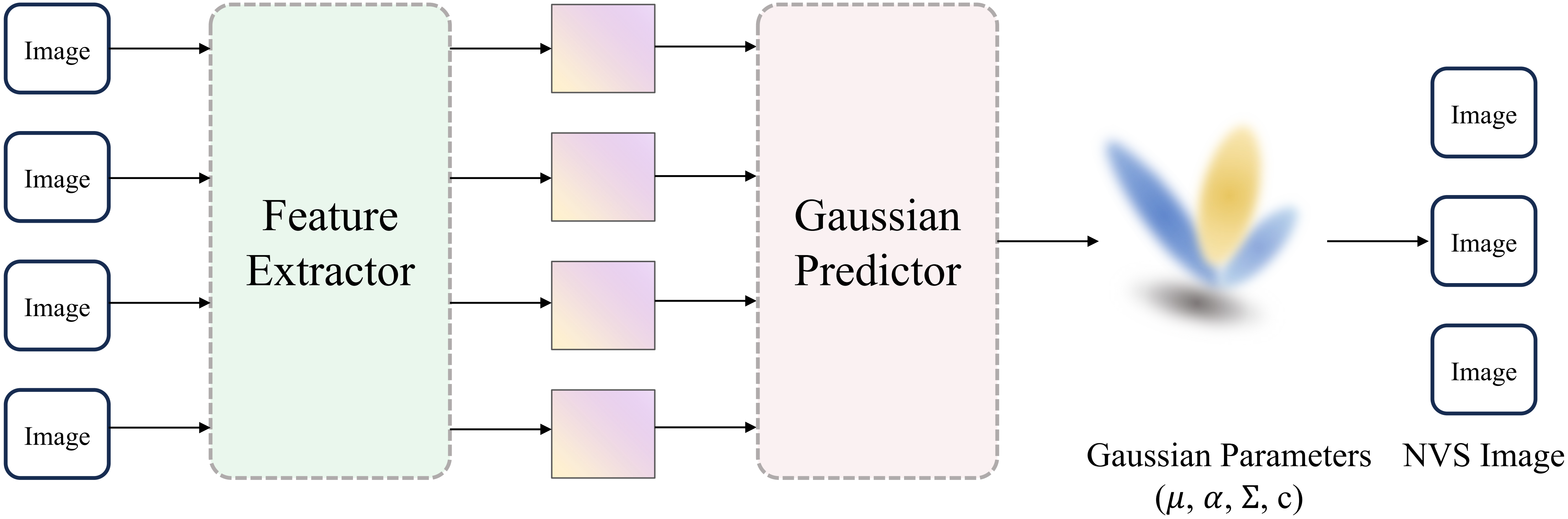

Duochao Shi[1]


Akide Liu[2]


Bohan Zhuang[1]

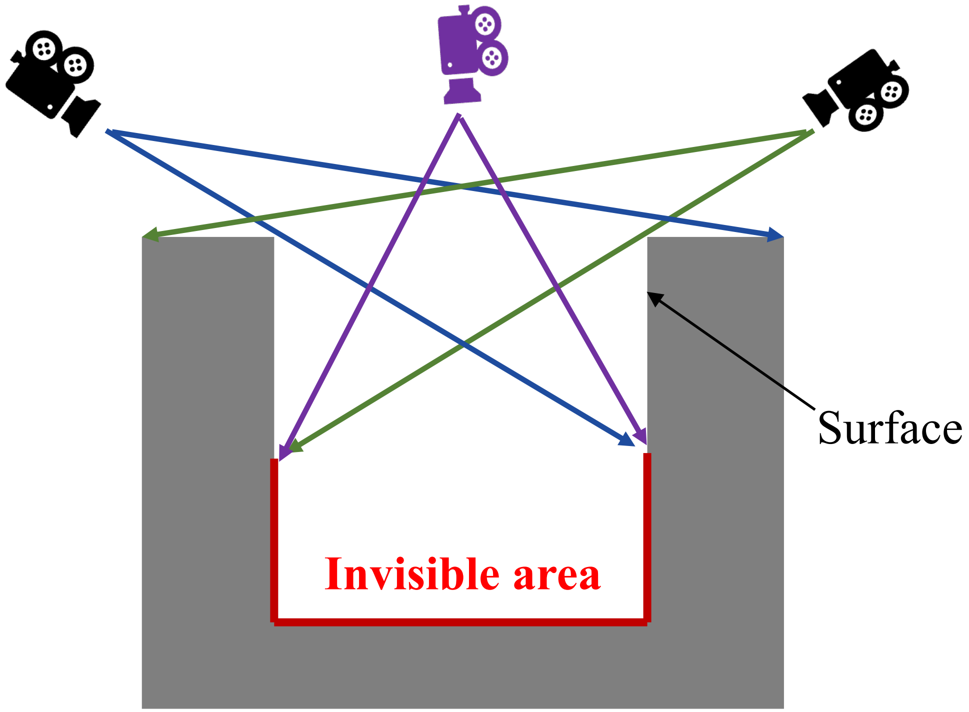
[1]Zhejiang University [2]Monash University

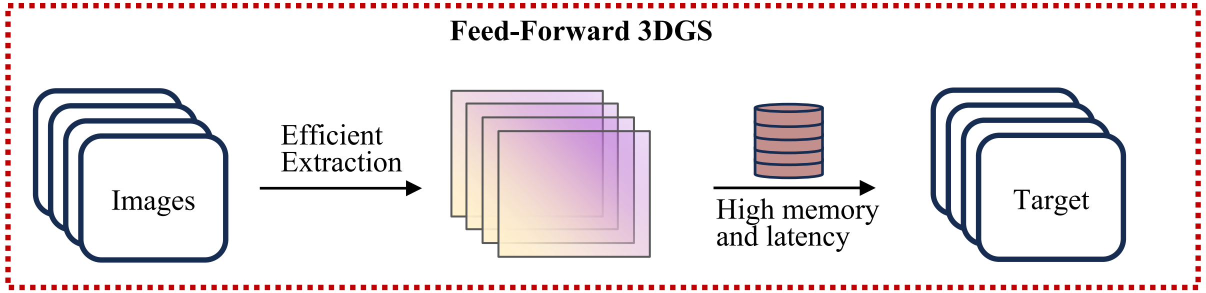
Pipeline of Feed-Forward 3DGS



Almost all feed-forward 3DGS networks use this paradigm.

Challenges in Feed-Forward 3DGS



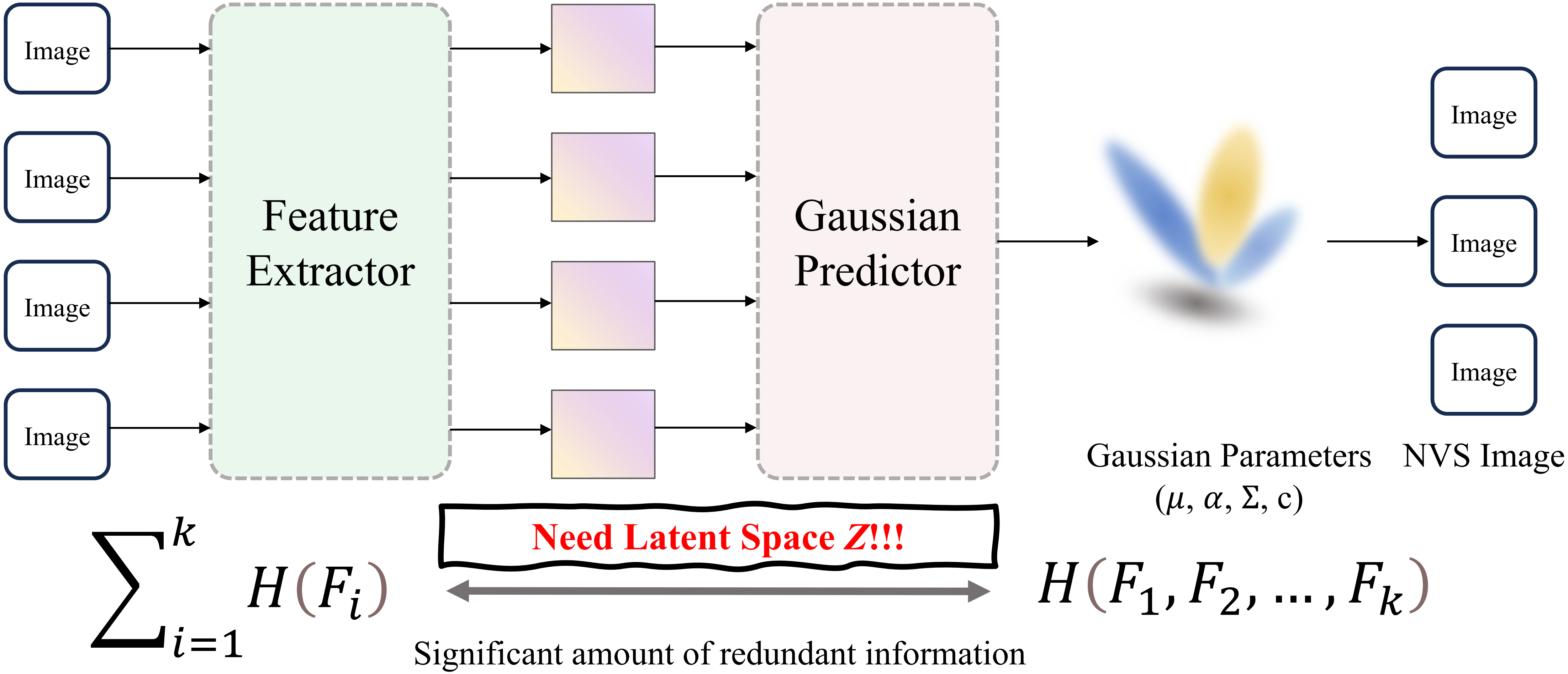


We need denser views to **provide more information**, but at the same time not be influenced by **redundancy**.

The scalability of feed-forward 3DGS is fundamentally constrained by the **limited capacity** of their networks.

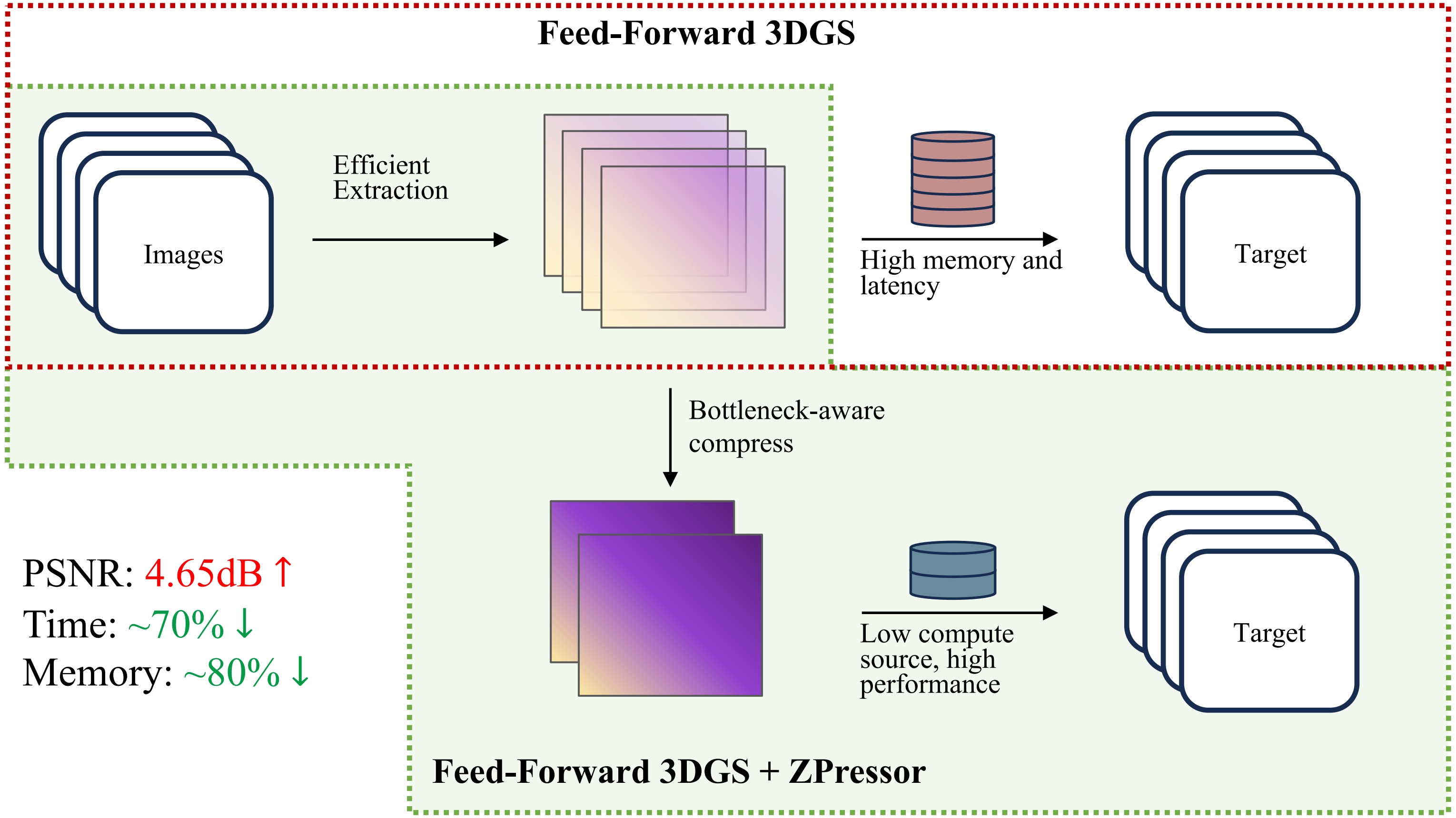
Fast3R: Towards 3D Reconstruction of 1000+ Images in One Forward Pass. CVPR 2025.

Information Flow in FF 3DGS



$$\sum_{i=1}^{k} H(F_i) \quad \longleftrightarrow \quad H(F_1, F_2, \ldots, F_k)$$

Bottleneck-Aware Compression



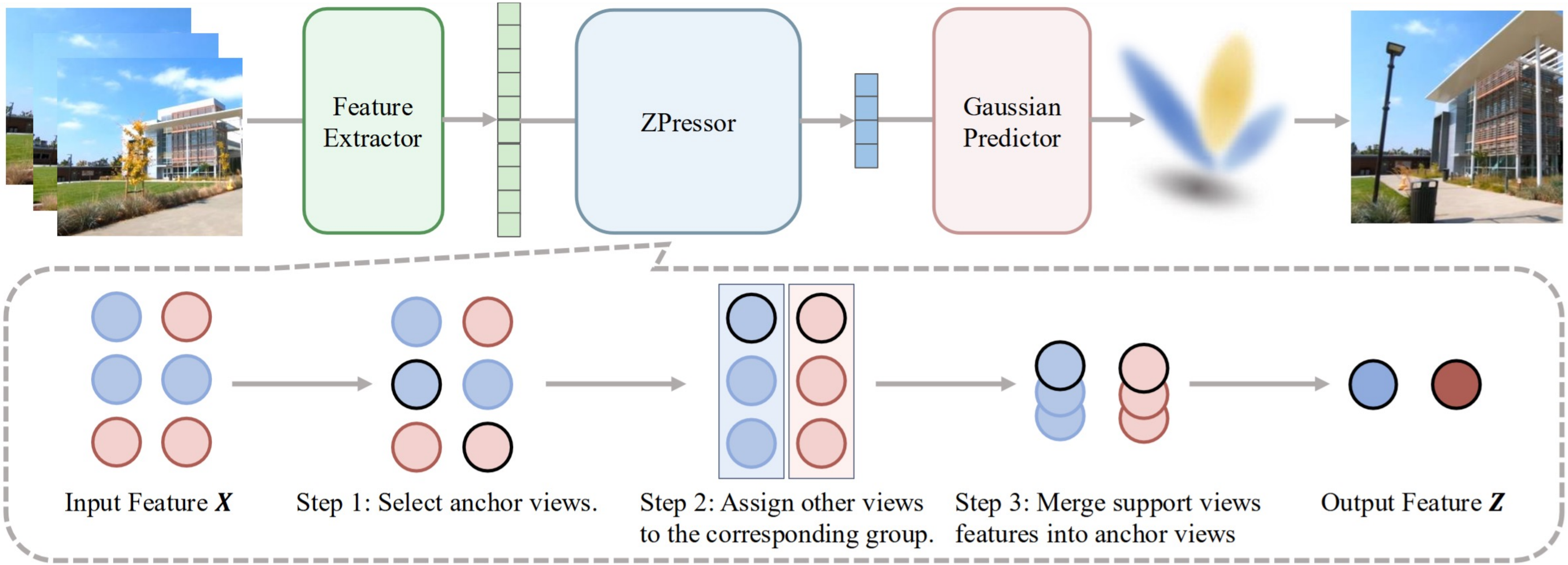
PSNR: 4.65dB ↑
Time: ~70% ↓
Memory: ~80% ↓

$$\min_{\mathcal{Z}} IB = \underbrace{\beta\, I(\mathcal{X}, \mathcal{Z})}_{\text{Compression Score}} - \underbrace{I(\mathcal{Z}, \mathcal{Y})}_{\text{Prediction Score}}$$

1. **Compression Score:** Minimizing I($\mathcal{X}$, $\mathcal{Z}$)
2. **Prediction Score:** Maximizing I($\mathcal{Z}$, $\mathcal{Y}$)

Note: The mutual information (MI) of two random variables I(·,·) is a measure of the mutual dependence between the two variables.

Zpressor: Overview



Anchor View Selection

Support-to-anchor Assignment

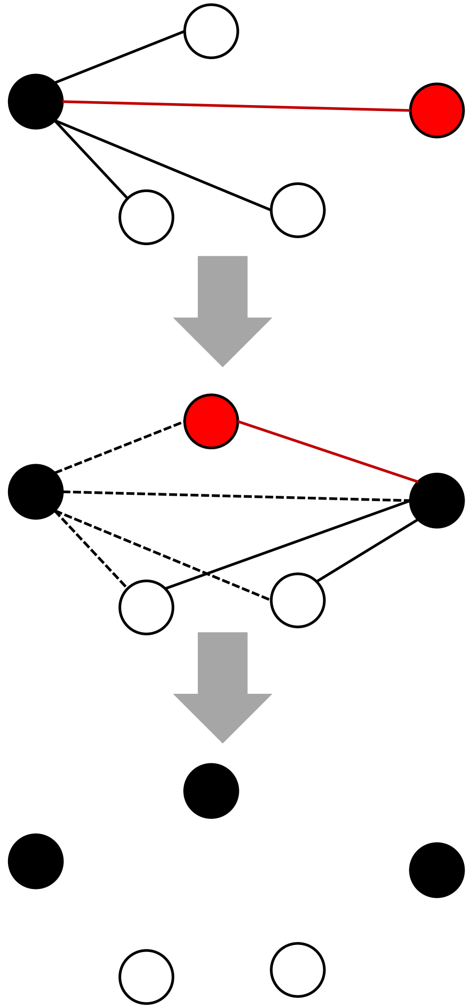
Views Information Fusion

Anchor View Selection

Algorithm 2 Farthest Point Sampling for Anchor View Selection

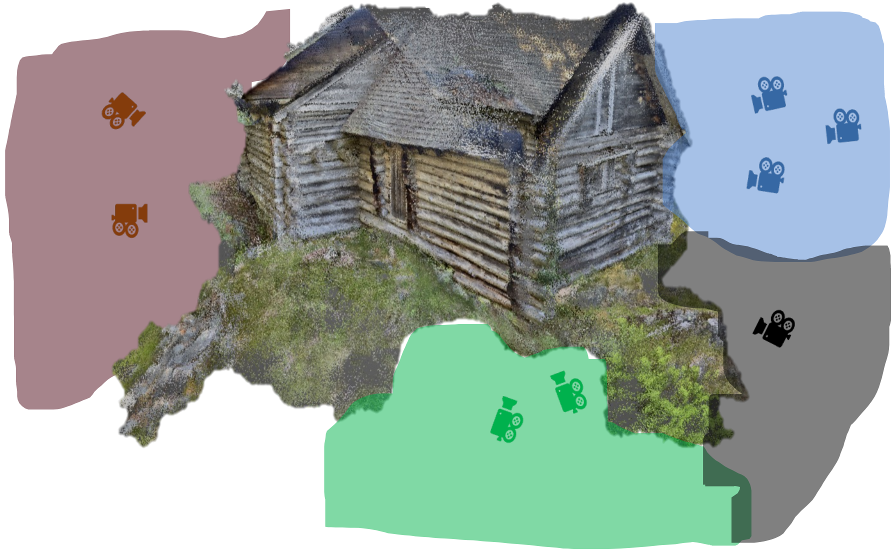
Input: Set of view camera positions $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_K\}$, Number of anchor views N
Output: Indices of the selected anchor views $\mathcal{S} = \{\mathbf{T}_{a_1}, \mathbf{T}_{a_2}, ..., \mathbf{T}_{a_n}\}$

```
Initialize the set of anchor view indices S ← ∅
Randomly select a random anchor view T_{a_1} ∈ T, where T_{a_1} ~ Uniform(T)
Add T_{a_1} to S: S ← {T_{a_1}}
for j ← 2 to N do
    Initialize a dictionary to store minimum distances D ← {}
    for k ← 1 to K do
        if k ∉ S then
            Calculate the minimum distance d_k ← min_{i∈S} ||T_k − T_i||_2
            Store the distance: D[k] ← d_k
        end if
    end for
    Find the view position T_{a_j} with the maximum minimum distance: T_{a_j} ← arg max_{k∉S} D[k]
    Add a_j to S: S ← S ∪ {T_{a_j}}
end for
return S
```



K=5; N=3

Support-to-anchor Assignment

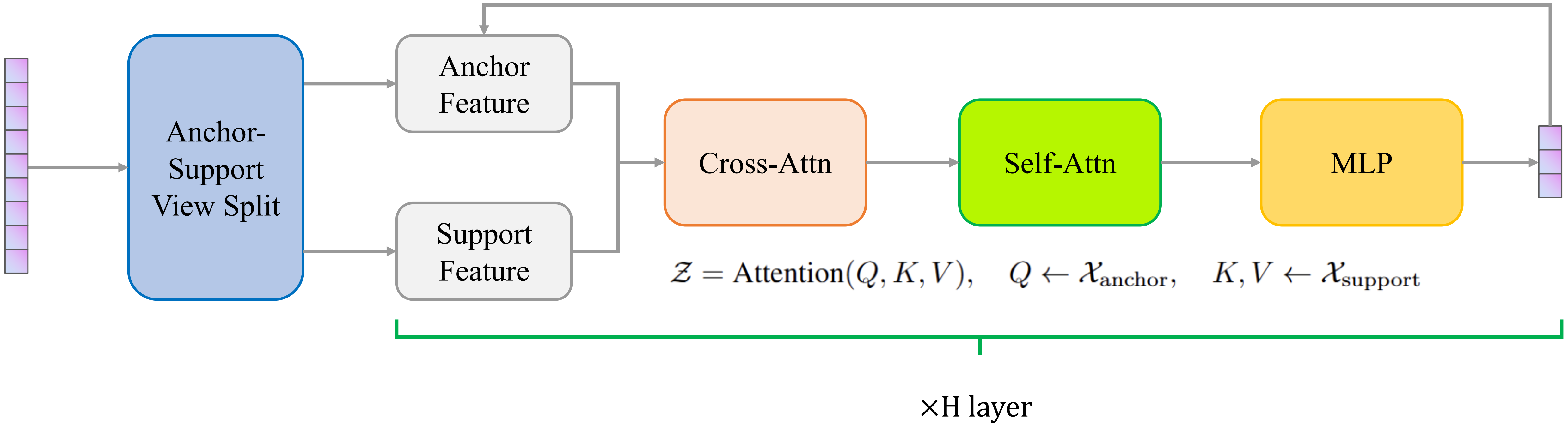


View Groups after Step 1 and Step 2

- Once anchor views are selected, each support view is assigned to its nearest anchor based on **camera position**.
- This grouping ensures that support views, which capture complementary scene details, are paired with **the most spatially relevant** anchor views.
- This pairing thereby ensures the effectiveness of information fusion.
- Formally, the cluster assignment to the i-th anchor view can be denoted as:

$$\mathcal{C}_i = \{f(\mathbf{T}) \in \mathcal{X}_{\text{support}} \mid \|\mathbf{T} - \mathbf{T}_{a_i}\| \leq \|\mathbf{T} - \mathbf{T}_{a_j}\|, \forall j \neq i\}$$

Views Information Fusion



Design of Feature Fusion Networks. Feature Fusion by Cross-Attention, Self-Attention and MLP.

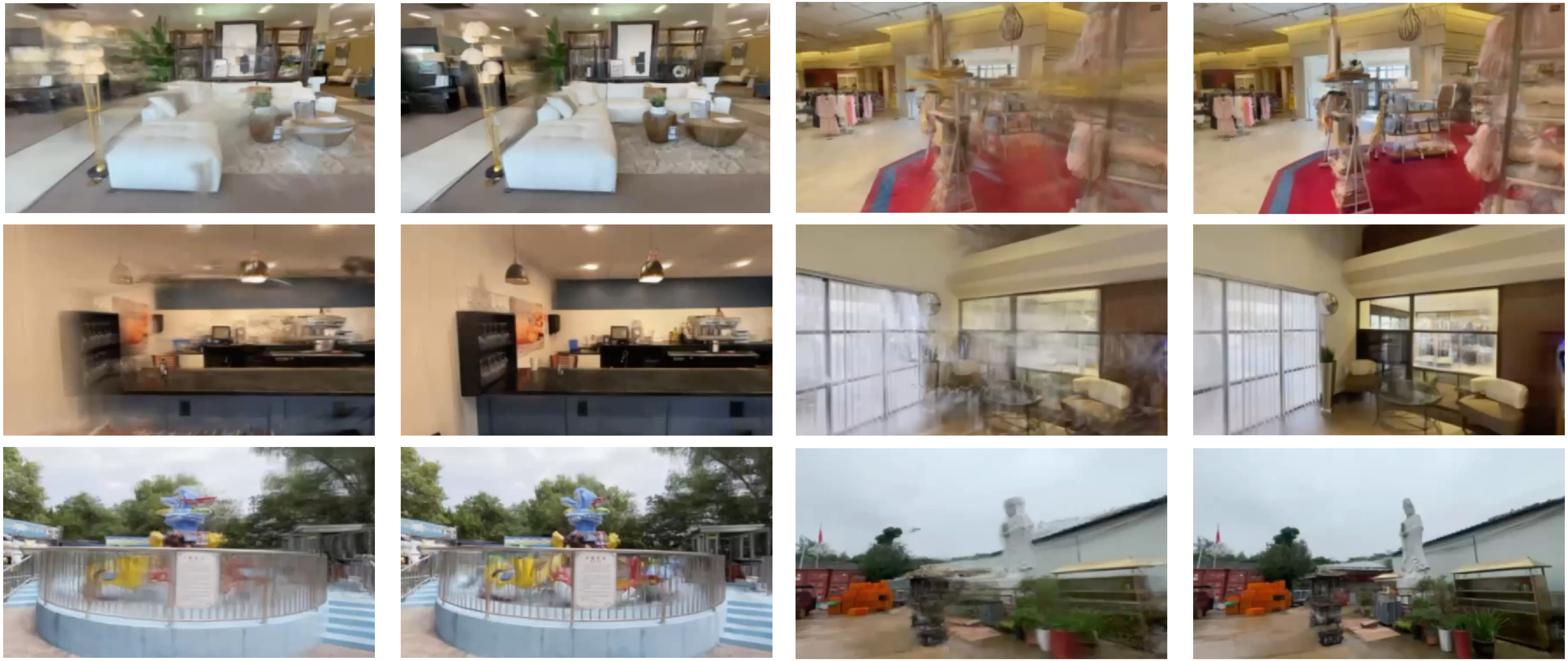
Results on DL3DV with DepthSplat

Views	Methods	PSNR↑	SSIM↑	LPIPS↓
36 views	DepthSplat	19.23	0.666	0.286
	DepthSplat + ZPressor	**23.88**+4.65	**0.815**+0.149	**0.150**-0.136
24 views	DepthSplat	20.38	0.711	0.253
	DepthSplat + ZPressor	**24.26**+3.88	**0.820**+0.109	**0.147**-0.106
16 views	DepthSplat	22.07	0.773	0.195
	DepthSplat + ZPressor	**24.25**+2.18	**0.819**+0.046	**0.147**-0.047
12 views	DepthSplat	23.32	0.807	0.162
	DepthSplat + ZPressor	**24.30**+0.97	**0.821**+0.014	**0.146**-0.017

Results on RE10K with MVSplat and pixelSplat

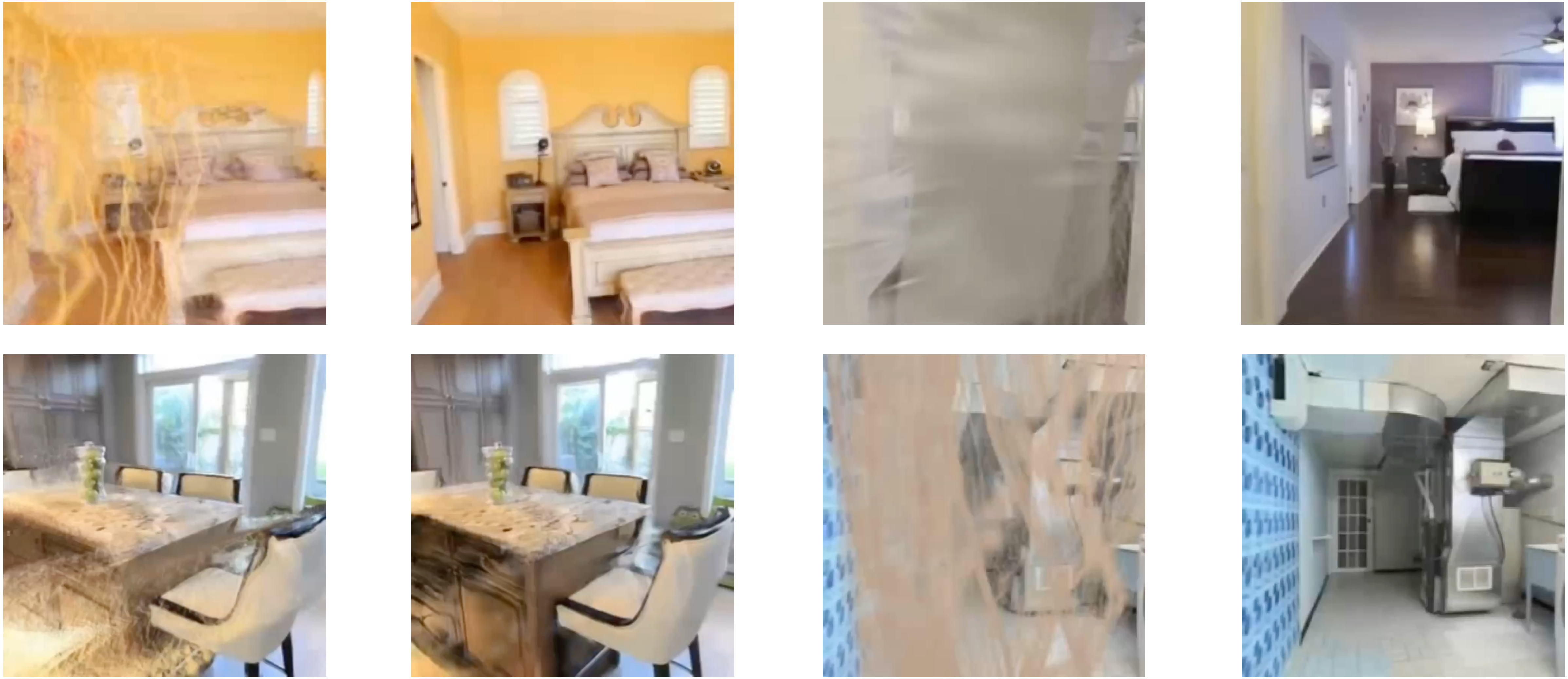
Views	Methods	PSNR↑	SSIM↑	LPIPS↓
36 views	pixelSplat	OOM	OOM	OOM
	pixelSplat + ZPressor	**26.59**	**0.849**	**0.225**
	MVSplat	24.19	0.851	0.155
	MVSplat + ZPressor	**27.34**+3.15	**0.893**+0.042	**0.113**-0.042
24 views	pixelSplat	OOM	OOM	OOM
	pixelSplat + ZPressor	**26.72**	**0.851**	**0.223**
	MVSplat	25.00	0.871	0.137
	MVSplat + ZPressor	**27.49**+2.49	**0.895**+0.024	**0.111**-0.026
16 views	pixelSplat	OOM	OOM	OOM
	pixelSplat + ZPressor	**26.81**	**0.853**	**0.221**
	MVSplat	25.86	0.888	0.120
	MVSplat + ZPressor	**27.60**+1.74	**0.896**+0.008	**0.110**-0.010
8 views	pixelSplat	26.19	0.852	**0.215**
	pixelSplat + ZPressor	**26.86**+0.67	**0.854**+0.002	0.219+0.004
	MVSplat	26.94	**0.902**	**0.107**
	MVSplat + ZPressor	**27.72**+0.78	0.897-0.005	0.109+0.002

Qualitative comparison



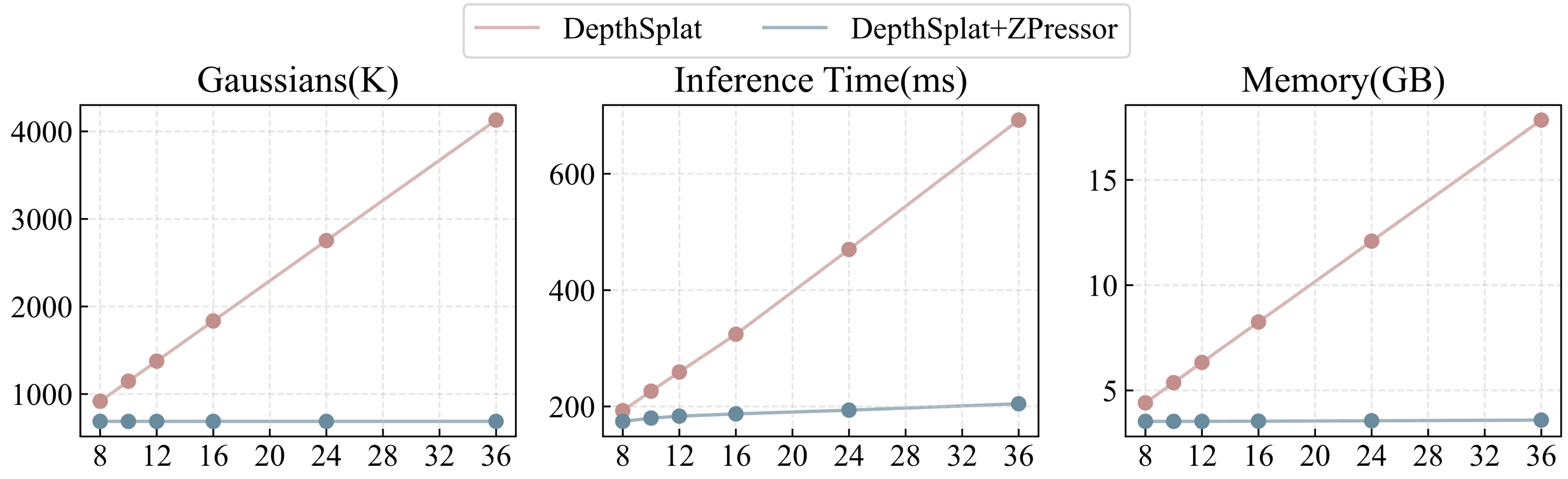
DepthSplat | DepthSplat+ZPressor | DepthSplat | DepthSplat+ZPressor

Qualitative comparison



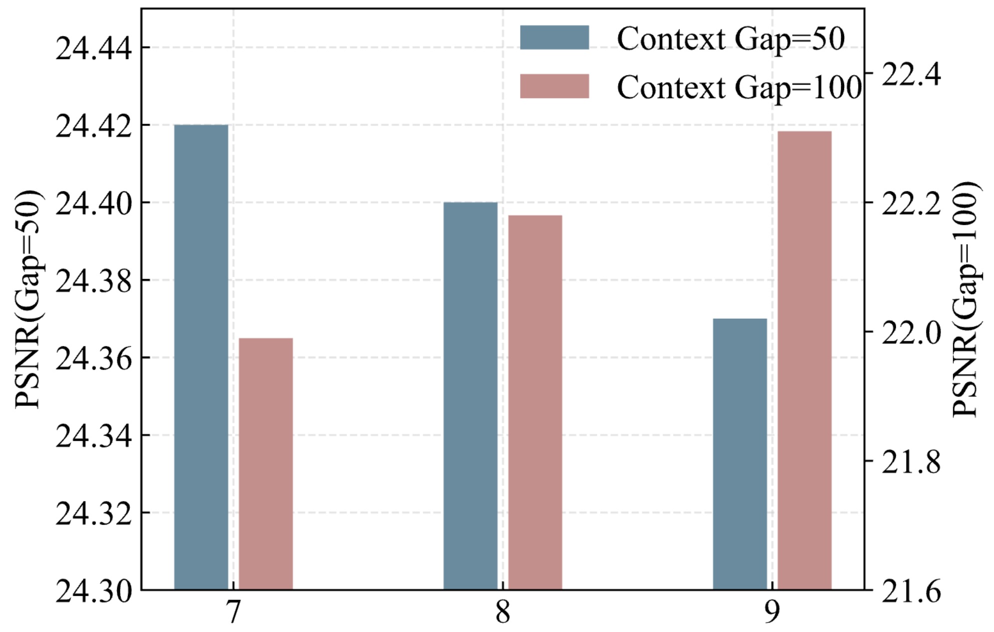
MVSplat | MVSplat+ZPressor | MVSplat | MVSplat+ZPressor

Model Efficiency



Linear no more: constant memory, constant time.

Bottleneck Analysis and Ablation Study

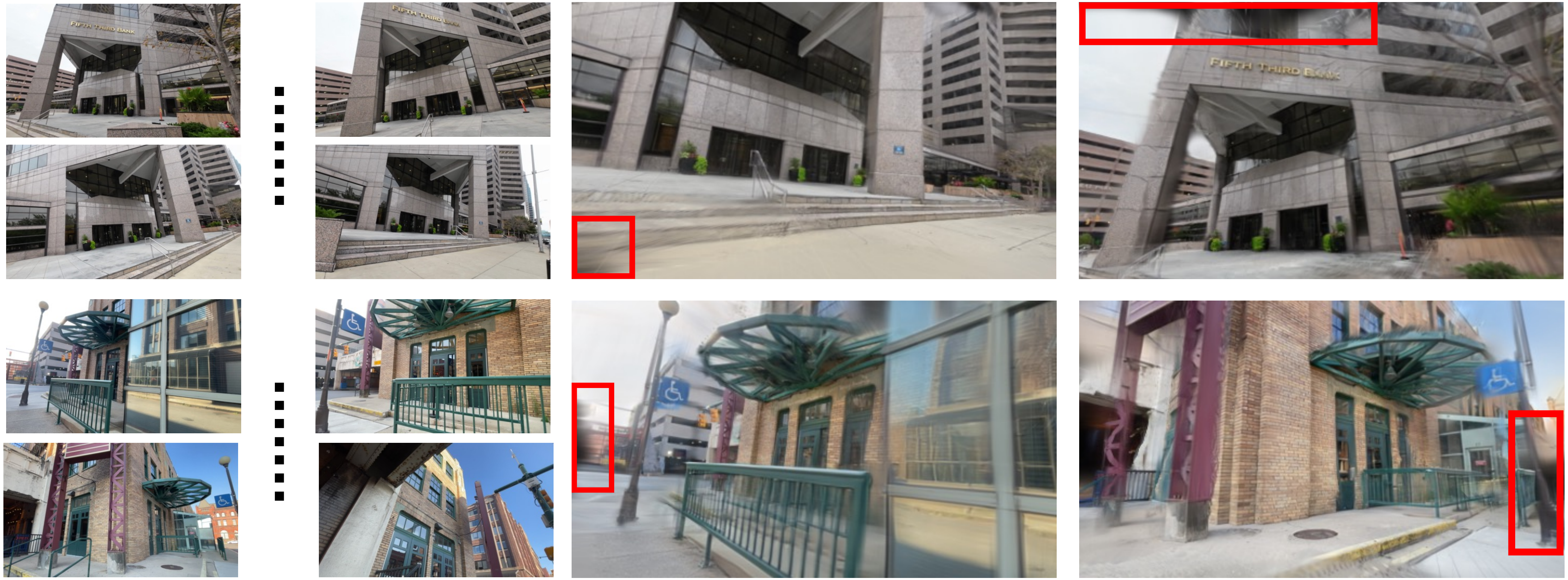


Analysis of bottleneck:

- Different levels of complexity benefit from different bottlenecks
- Effective compression preserves essential scene information.

Methods	PSNR↑	SSIM↑	LPIPS↓	Time (s)	Peak Memory (GB)
DepthSplat + ZPressor	**24.30**	**0.821**	**0.146**	0.184	3.80
w/o multi-blocks	24.18	0.817	0.149	**0.140**	**3.79**
w/o self-attention	23.85	0.810	0.156	0.183	3.80
DepthSplat	23.32	0.808	0.162	0.260	6.80

Limitations



Inputs (~500 views) DepthSplat + ZPressor

ZPressor exhibits limitations when processing scenarios with an **extremely high** density of input views.

More Information

Paper, code and models are available on our project page.

Weijie Wang's homepage. Actively seeking internship opportunities.

Conclusion:

- ZPressor is a **lightweight, architecture-agnostic** module designed for scalable feed-forward 3DGS
- We bridges IB principle and 3D generative modeling, offering a new perspective on scalable 3D scene reconstruction.