

C3G: Learning Compact 3D Representations with 2K Gaussians

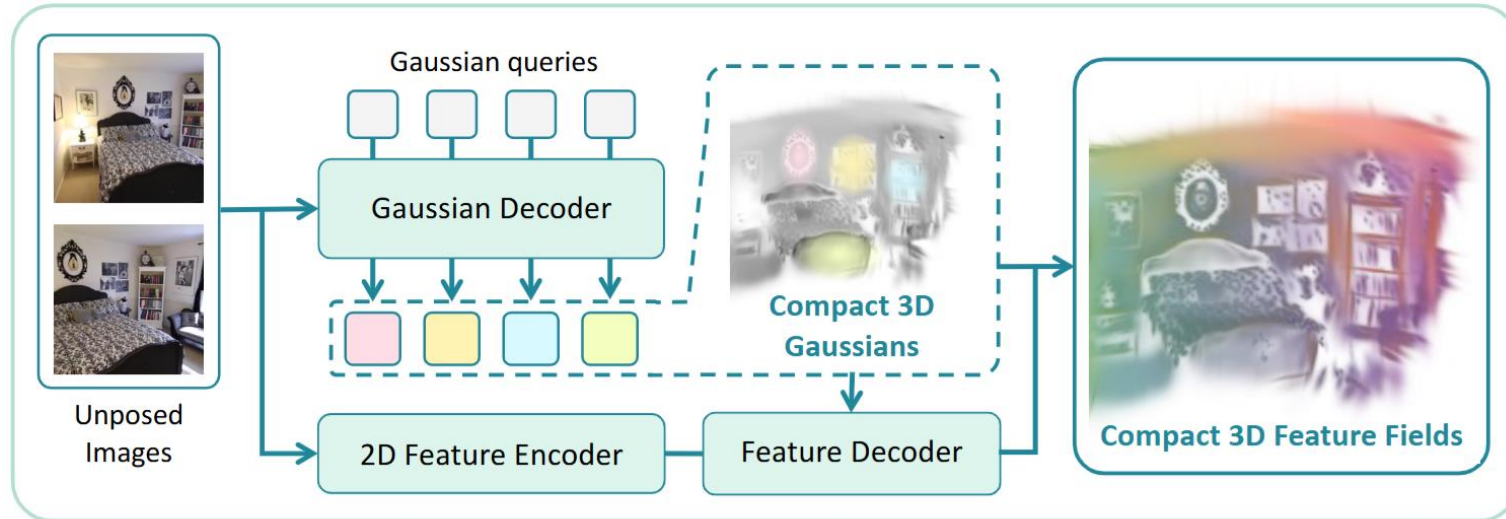
Honggyu An

honggyu@kaist.ac.kr

Computer Vision Laboratory (CVLAB)
Graduate School of Artificial Intelligence, KAIST

Problem Definition

Given *sparse unposed multi-views* as input, we want to estimate *3D Gaussians* that compactly represents the scene for *novel view synthesis* and *3D scene understanding*.



Applications

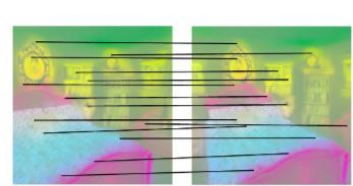
Novel View Synthesis



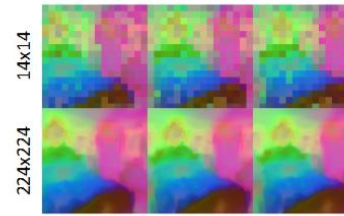
3D Scene Understanding



Multiview Correspondence



Multiview Upsampling



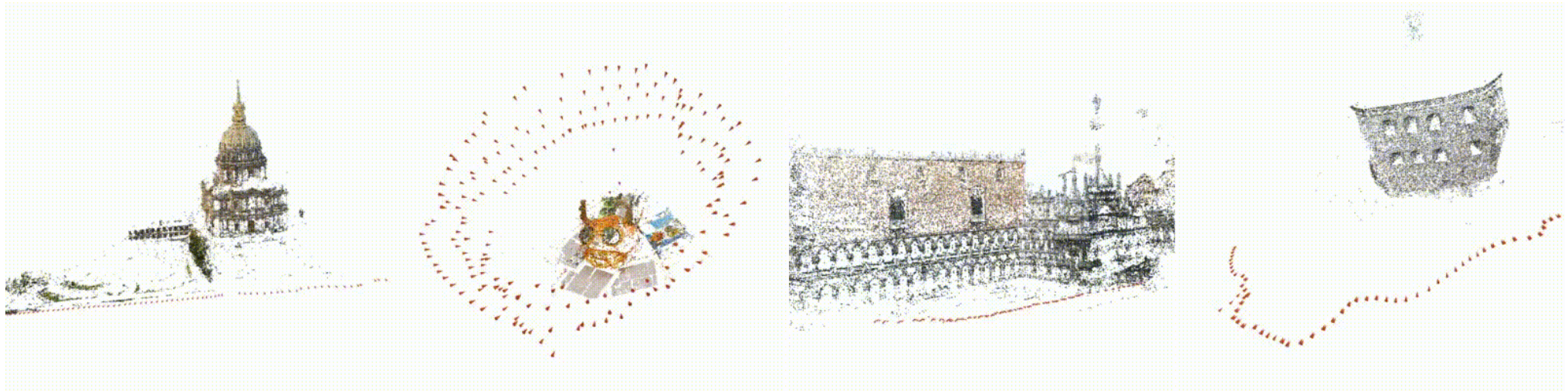
Problem Definition - Input

Given *sparse unposed multi-views* as input, we want to estimate *3DGS* that compactly represents the scene for *novel view synthesis* and *3D scene understanding*.

Q. What do we need to reconstruct 3D scenes?

-> *Dense* number of multiple *posed* images.

-> Requires slow per-scene optimization to obtain camera poses.



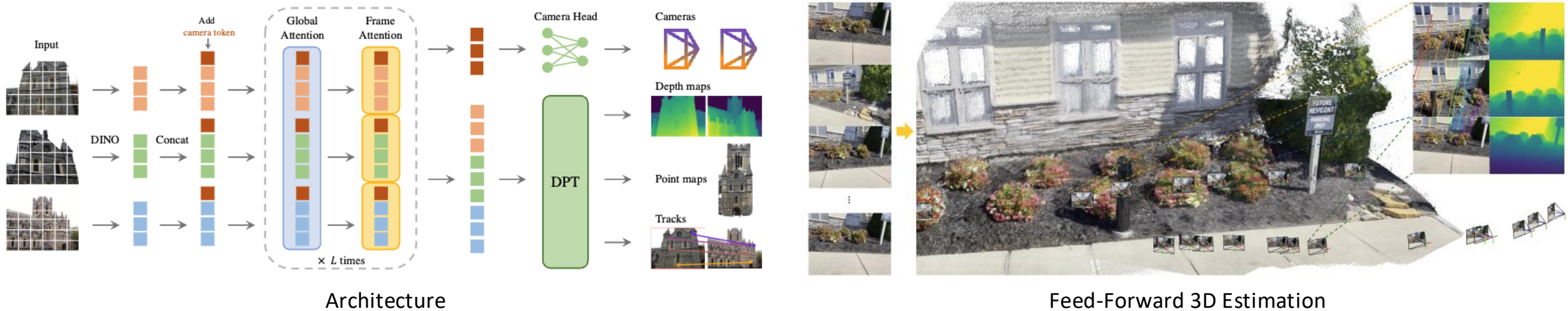
Problem Definition - Input

Given *sparse unposed multi-views* as input, we want to estimate *3DGS* that compactly represents the scene for *novel view synthesis* and *3D scene understanding*.

Q. How can we get rid of the constraint of dense images?

-> Build a model with *priors* learned from *large-scale multi-view data*!

-> Get rid of slow per-scene optimization, and requirement of dense views

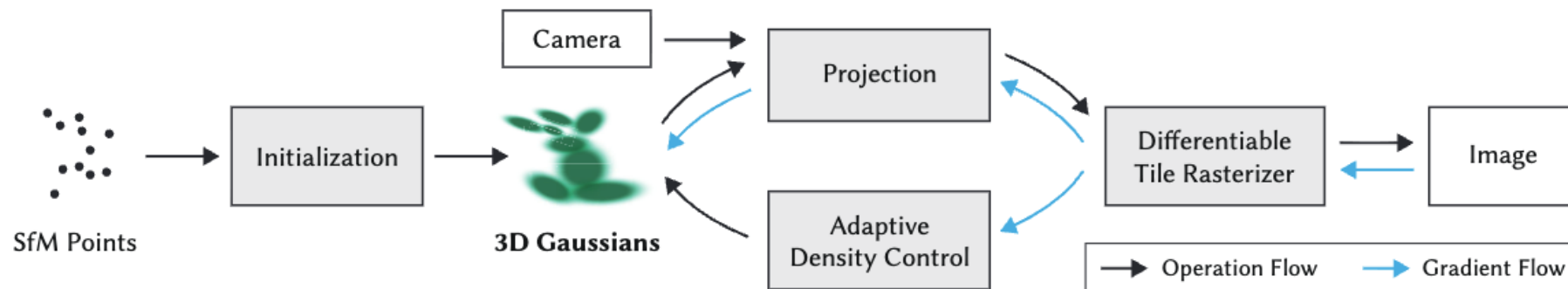


Problem Definition - Representation

Given *sparse unposed multi-views* as input, we want to estimate **3DGS** that compactly represents the scene for *novel view synthesis* and *3D scene understanding*.

Q. How can we represent the 3D scene to solve various downstream tasks?

-> 3D Gaussian Splatting using 3D Gaussian blobs



3D Gaussian Splatting

Problem Definition – Task

Given *sparse unposed multi-views* as input, we want to estimate *3DGS* that compactly represents the scene for *novel view synthesis* and *3D scene understanding*.

Q. What is novel view synthesis?

-> Given a set of images, we want to synthesize an image as it was seen from a novel viewpoint!



Synthesize Images from
Novel Viewpoints

→

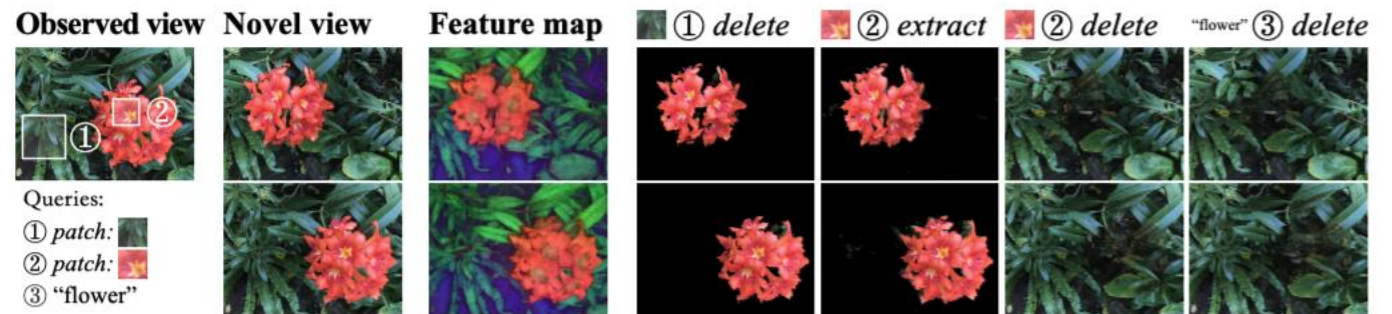
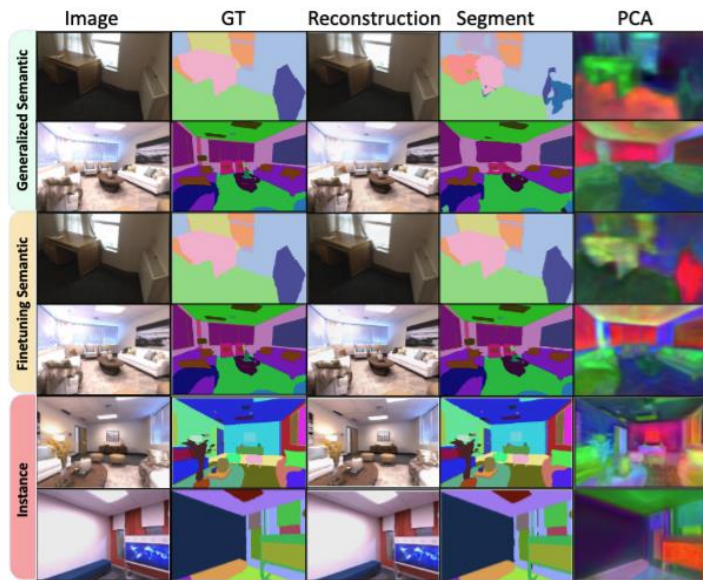


Problem Definition – Task

Given *sparse unposed multi-views* as input, we want to estimate *3DGS* that compactly represents the scene for *novel view synthesis* and *3D scene understanding*.

Q. What is 3D scene understanding?

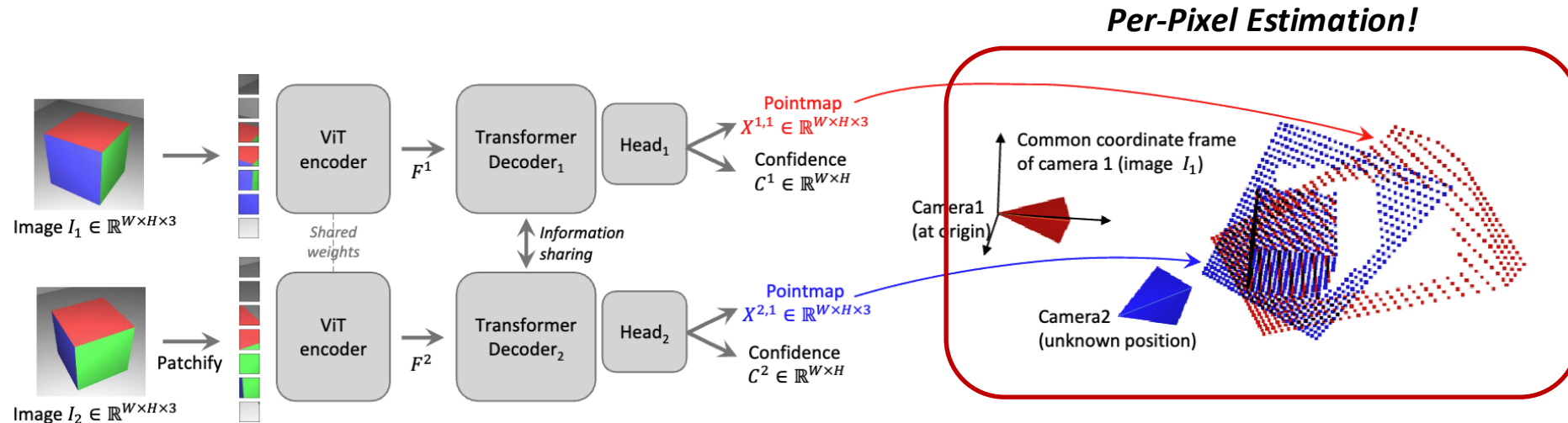
-> Given a set of images, we want to understand (segment) the images in the scene and enable novel-view segmentations!



Baseline

What are the problems of existing models?

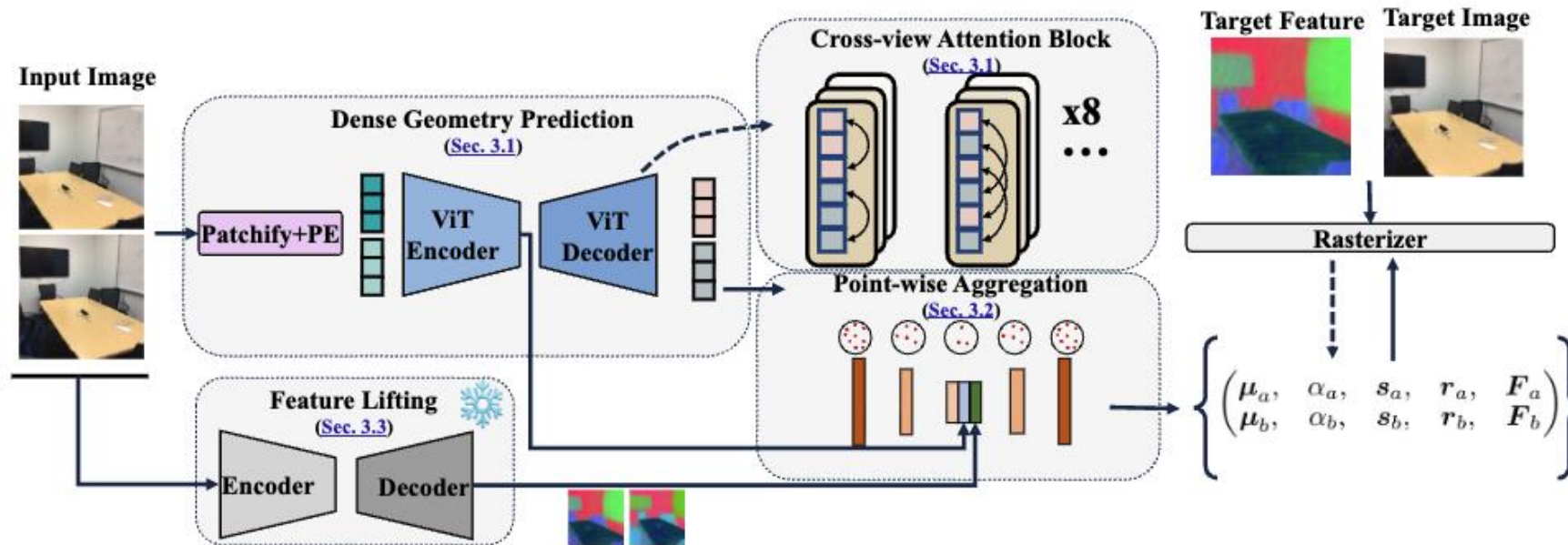
-> They estimated per-pixel Gaussians, which lead to degraded novel view synthesis performance and degraded 3D scene understanding.



Baseline – Large Spatial Model

Given *sparse unposed multi-views* as input, we want to estimate *3DGS* that compactly represents the scene for *novel view synthesis* and *3D scene understanding*.

From the learned 3D Gaussians, lift semantic features for 3D scene understanding

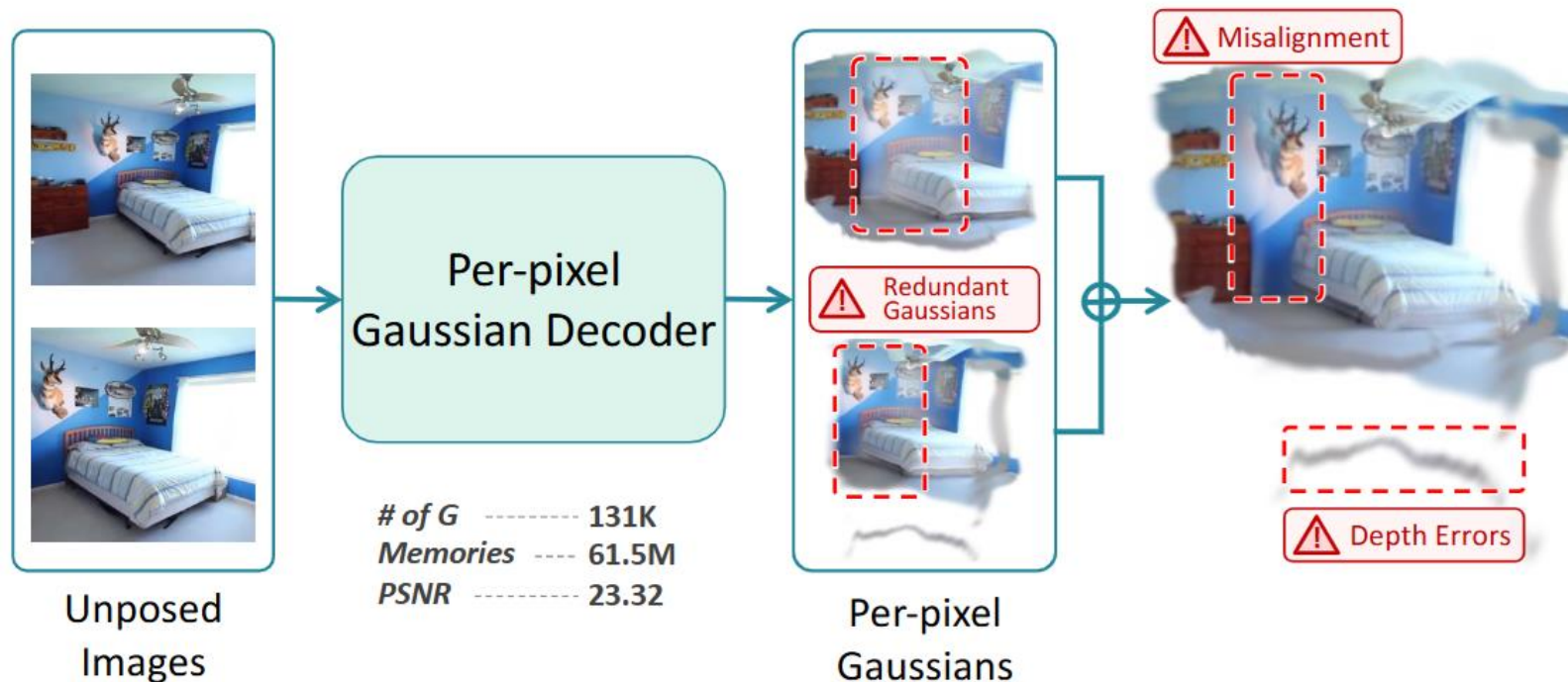


Baseline – Large Spatial Model

What are the problems of existing models?

-> Per-pixel Gaussians often lead to mis-aligned Gaussians, causing artifacts and noisy images when rendering to novel views.

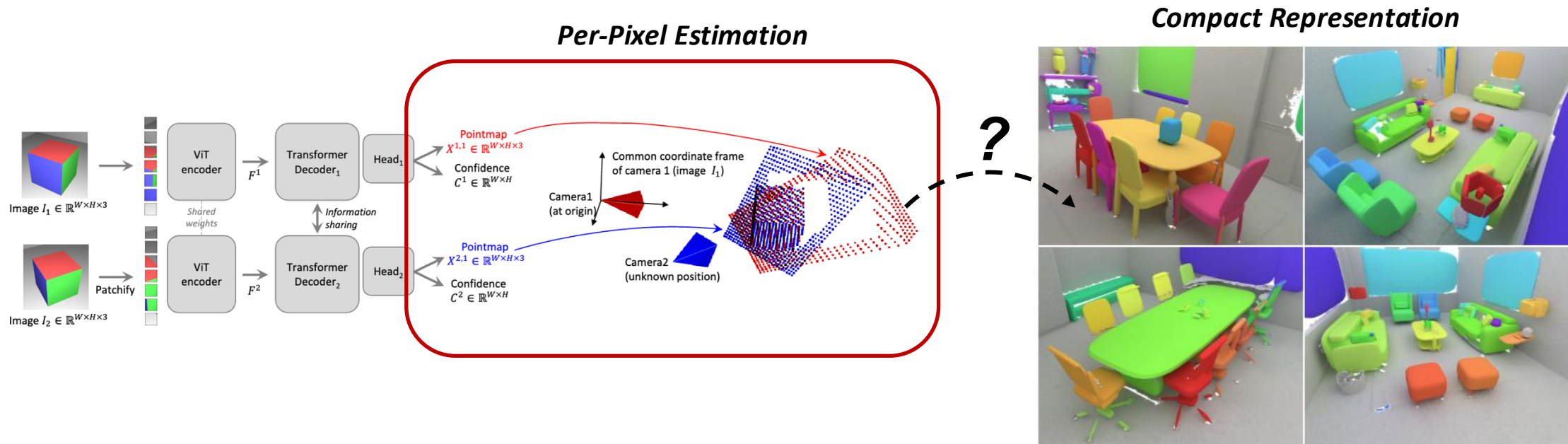
Pixel-Aligned 3D Scene Representation



Baseline – Large Spatial Model

Given sparse unposed multi-views as input, we want to estimate 3DGS that *compactly* represents the scene for novel view synthesis and 3D scene understanding.

Q. Do we really need such per-pixel, dense scene representations?

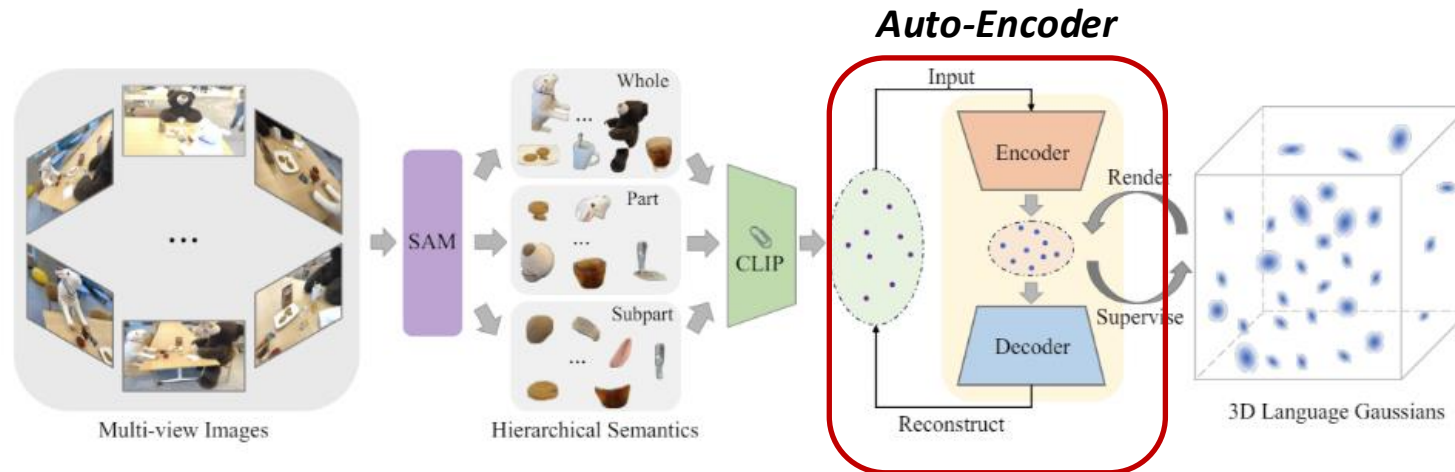


Baseline – Large Spatial Model

What are the problems of existing models?

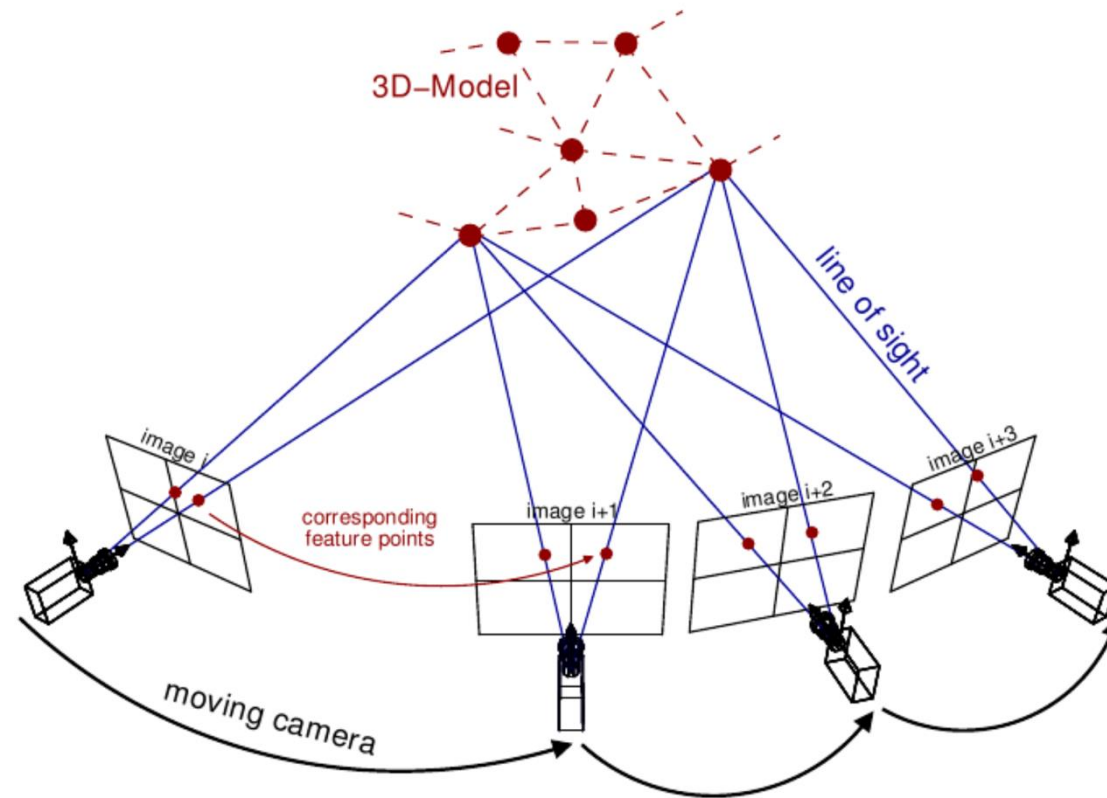
-> Per-pixel Gaussians results with too many Gaussians, which results with large computational overhead. This requires an auxiliary model to compress the feature dimensions.

-> This leads to less computation, but also results in information loss.



Approach

When reconstructing 3D, we can always merge corresponding points or regions across multi-views.



Approach

When reconstructing 3D, we can always merge corresponding points or regions across multi-views.

How can we merge points or Gaussians modeling similar regions?

1. Top-down: First estimate per-pixel Gaussians -> Merge Gaussians post-hoc

2. Bottom-up: First aggregate similar regions -> estimate essential Gaussians only

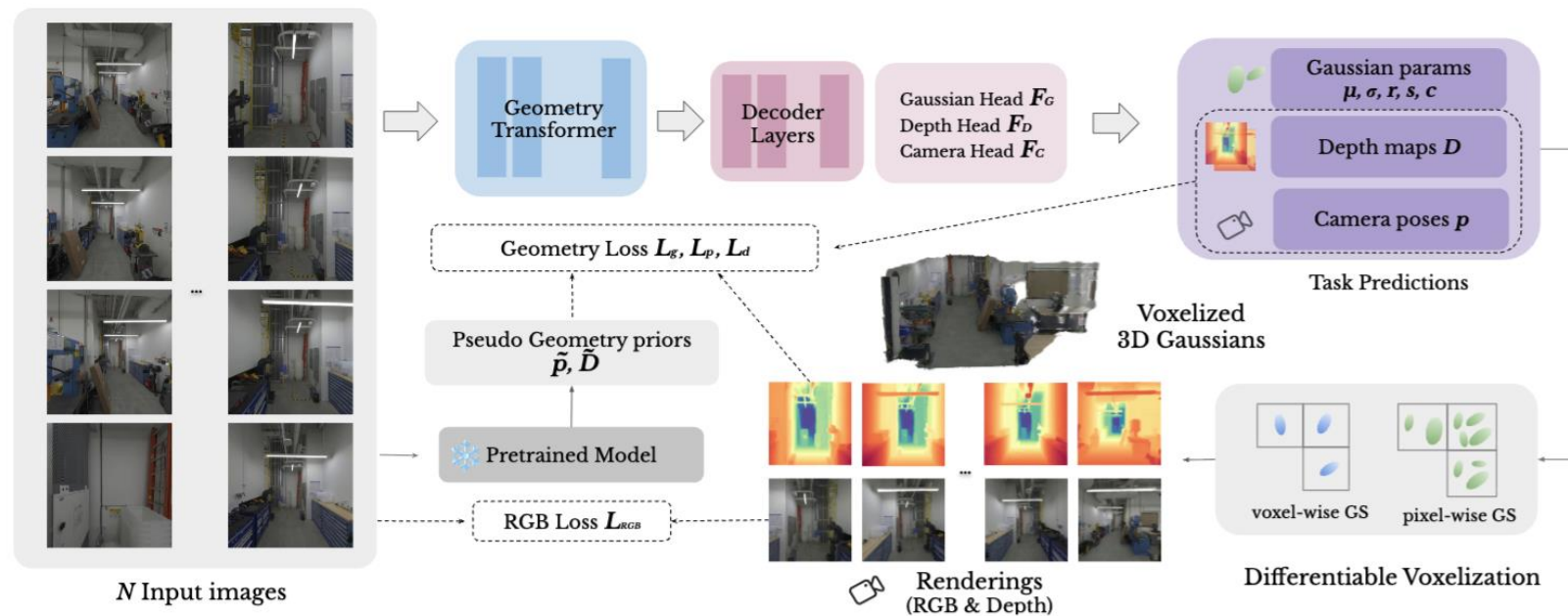
Approach

When reconstructing 3D, we can always merge corresponding points or regions across multi-views.

How can we merge points or Gaussians modeling similar regions?

1. *Top-down*: First estimate per-pixel Gaussians -> Merge Gaussians post-hoc

2. *Bottom-up*: First aggregate similar regions -> estimate essential Gaussians only



Solution

When reconstructing 3D, we can always merge corresponding points or regions across multi-views.

How can we merge points or Gaussians modeling similar regions?

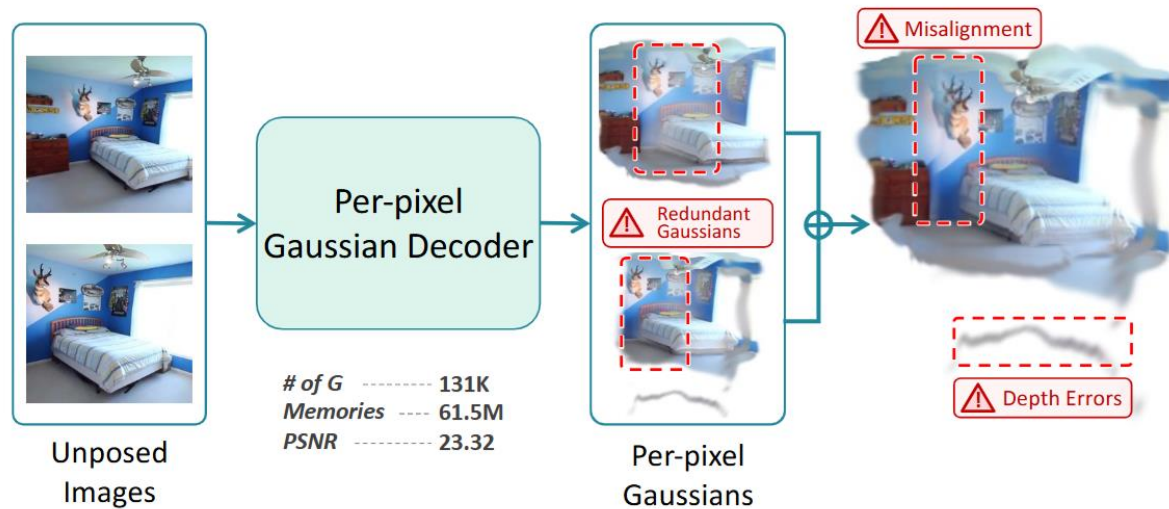
1. *Top-down*: First estimate per-pixel Gaussians -> Merge Gaussians post-hoc
2. *Bottom-up*: First aggregate similar regions -> estimate essential Gaussians only

Methods

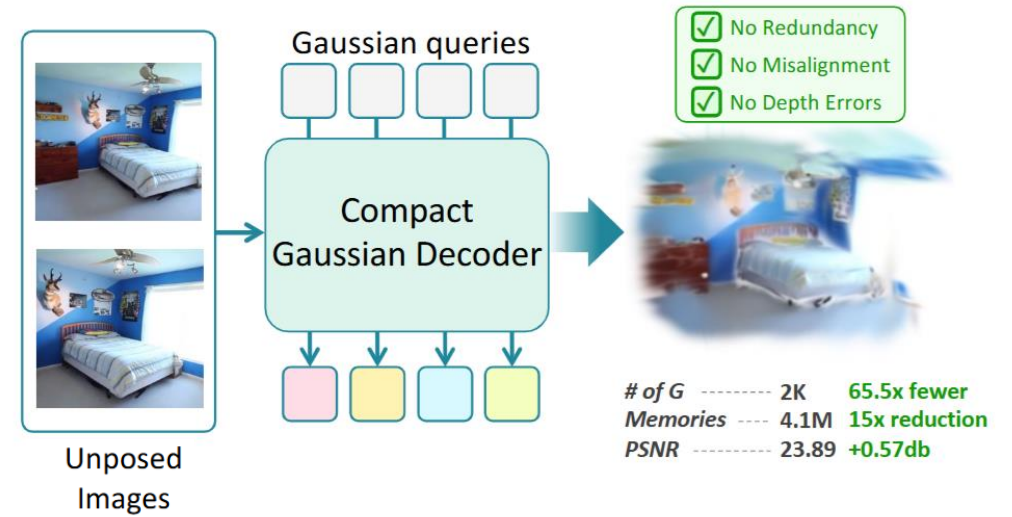
How can we estimate only essential Gaussians to reconstruct the scene?

-> We look into transformer decoder structures, which also decode outputs from learnable queries.

Pixel-Aligned 3D Scene Representation



Compact 3D Scene Representation



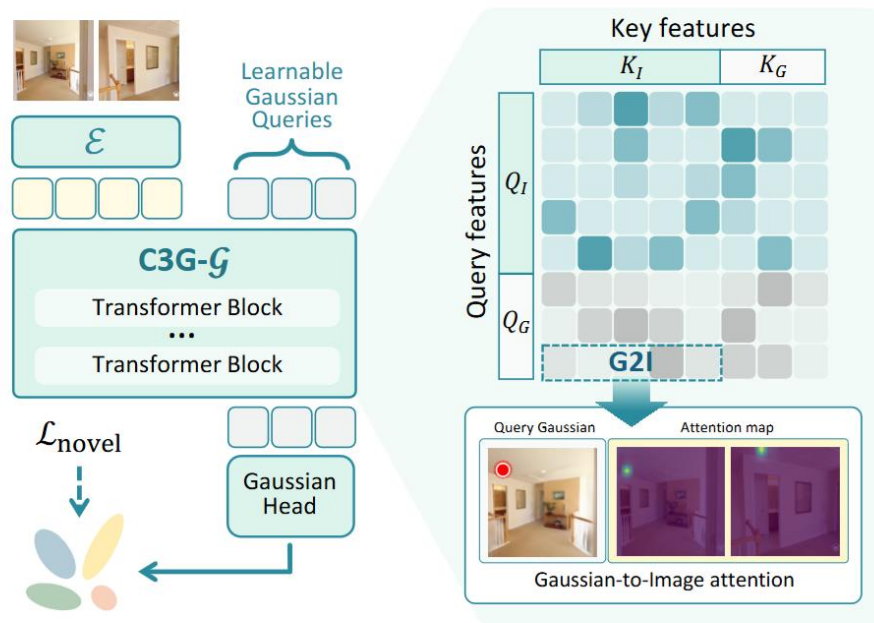
Methods

Compact Gaussian Decoding

-> A significant advantage of this approach is that, we can learn the model to estimate essential Gaussians with only the *novel view synthesis objective*!

-> How are these Gaussians learned?

-> Through *bi-directional attention* of the encoder features and learnable queries!



(a) Overall framework



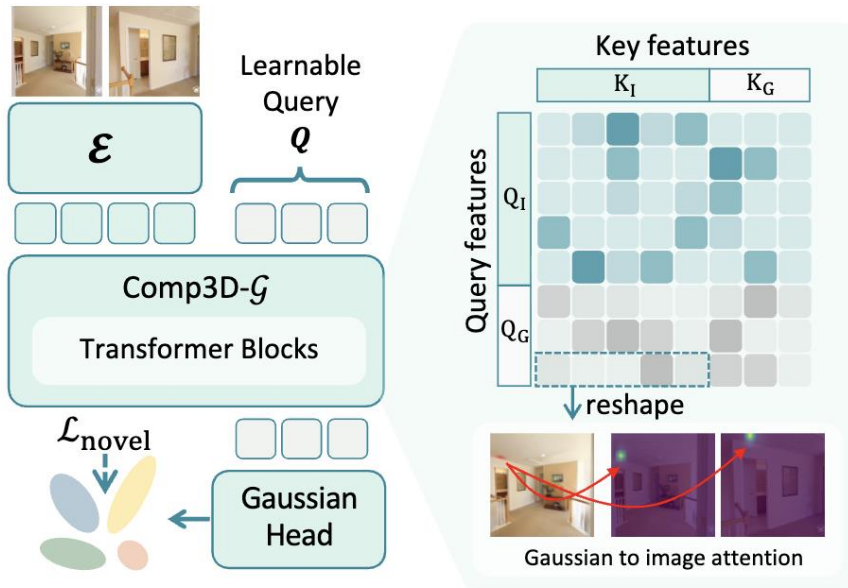
(b) Attention visualization

Methods

Any-Feature Lifting

We introduce another feature aggregator that decodes the *view-invariant feature* to attach to each Gaussian.

We freeze the attention weights, and introduce new query tokens to handle different dimensions.



(a) Overall framework



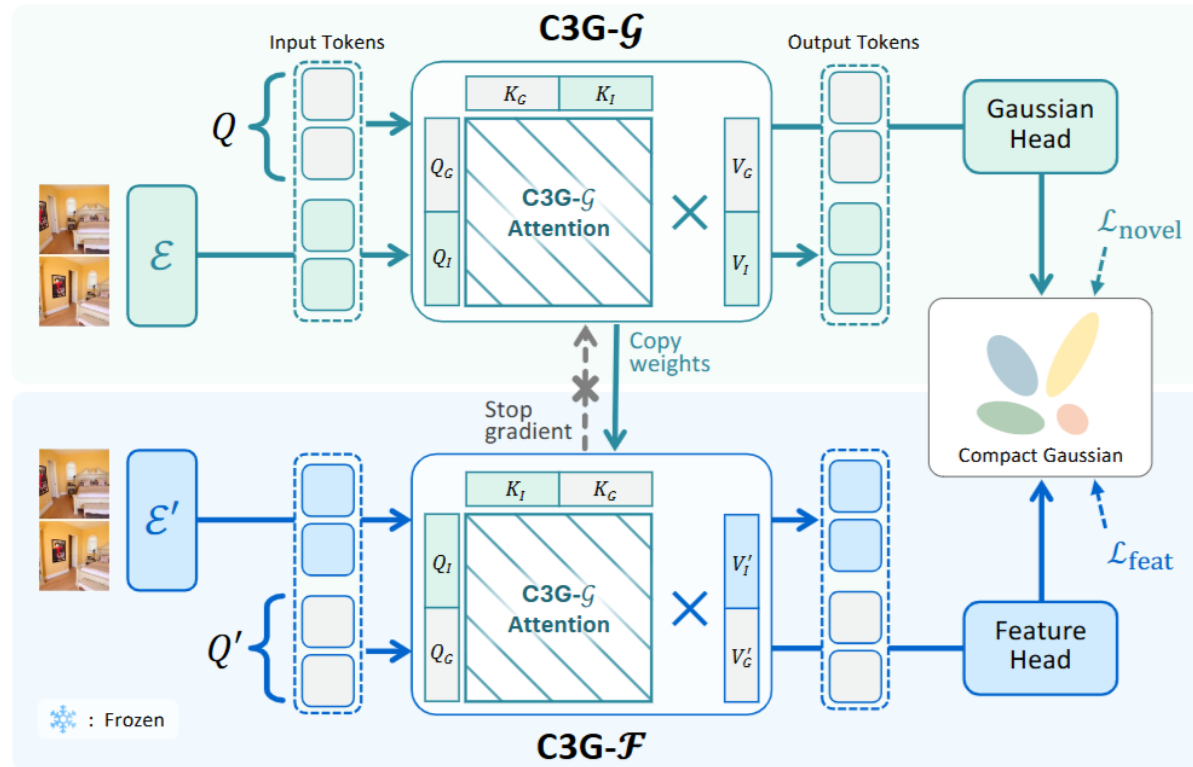
(b) Attention visualization

Analysis

Any-Feature Lifting

We introduce another feature aggregator that decodes the *view-invariant feature* to attach to each Gaussian.

We freeze the attention weights, and introduce new query tokens to handle different dimensions.



Results

Joint Reconstruction and Understanding

- (1) Reconstruction: We evaluate the reconstruction quality by the novel view rendering results
- (2) 3D Scene Understanding : We evaluate the open-vocabulary segmentation

Table 3. **Comparison of 3D scene understanding on Scannet.** We lift LSeg [23] and MaskCLIP[6] features from two input views and evaluate open-vocabulary segmentation on target views. Our method generates fewer Gaussians while outperforming feed-forward and per-scene optimization methods trained with substantially more posed inputs. *: Features directly extracted from target view images.

Target View																
Methods	Feature	Feed Forward	Input Pose	LSeg [23]					MaskCLIP [35]					#G↓	Memories↓	FPS↑
				mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓			
LSeg / MaskCLIP* [23, 35]	✓	✗	-	0.506	0.797	-	-	-	0.341	0.667	-	-	-	-	-	-
Feature-3DGS [59]	✓	✗	✓	0.379	0.644	19.83	0.684	0.357	0.353	0.663	17.47	0.612	0.420	1,185K	845.2MB	19.2
CF ³ [21]	✓	✗	✓	0.376	0.657	20.04	0.691	0.359	0.336	0.634	20.14	0.695	0.354	53K	38.4MB	252.4
NoPoSplat [53]	✗	✓	✗	-	-	24.59	0.792	0.228	-	-	24.59	0.792	0.228	131K	33.6MB	369.8
LSM [9]	✓	✓	✗	0.503	0.793	23.32	0.767	0.250	0.286	0.505	22.87	0.737	0.286	131K	61.5MB	254.5
Ours	✓	✓	✗	0.513	0.783	23.89	0.770	0.285	0.369	0.675	23.75	0.763	0.290	2K	4.1MB	243.4

Source View																
Methods	Feature	Feed Forward	Input Pose	LSeg [23]					MaskCLIP [35]					#G↓	Memories↓	FPS↑
				mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓	mIoU↑	Acc↑	PSNR↑	SSIM↑	LPIPS↓			
LSeg / MaskCLIP* [23, 35]	✓	✗	-	0.521	0.820	-	-	-	0.344	0.665	-	-	-	-	-	-
Feature-3DGS [59]	✓	✗	✓	0.392	0.655	21.73	0.757	0.314	0.353	0.674	22.25	0.777	0.308	1,185K	845.2MB	19.2
CF ³ [21]	✓	✗	✓	0.390	0.668	22.99	0.804	0.272	0.342	0.642	23.16	0.812	0.265	53K	38.4MB	252.4
NoPoSplat [53]	✗	✓	✗	-	-	25.20	0.812	0.217	-	-	25.20	0.812	0.217	131K	33.6MB	369.8
LSM [9]	✓	✓	✗	0.511	0.798	25.44	0.811	0.214	0.251	0.516	25.01	0.824	0.230	131K	61.5MB	254.5
Ours	✓	✓	✗	0.542	0.803	23.92	0.766	0.278	0.361	0.668	23.39	0.759	0.284	2K	4.1MB	243.4

Results

Joint Reconstruction and Understanding

- (1) Reconstruction: We evaluate the reconstruction quality by the novel view rendering results
- (2) 3D Scene Understanding : We evaluate the open-vocabulary segmentation

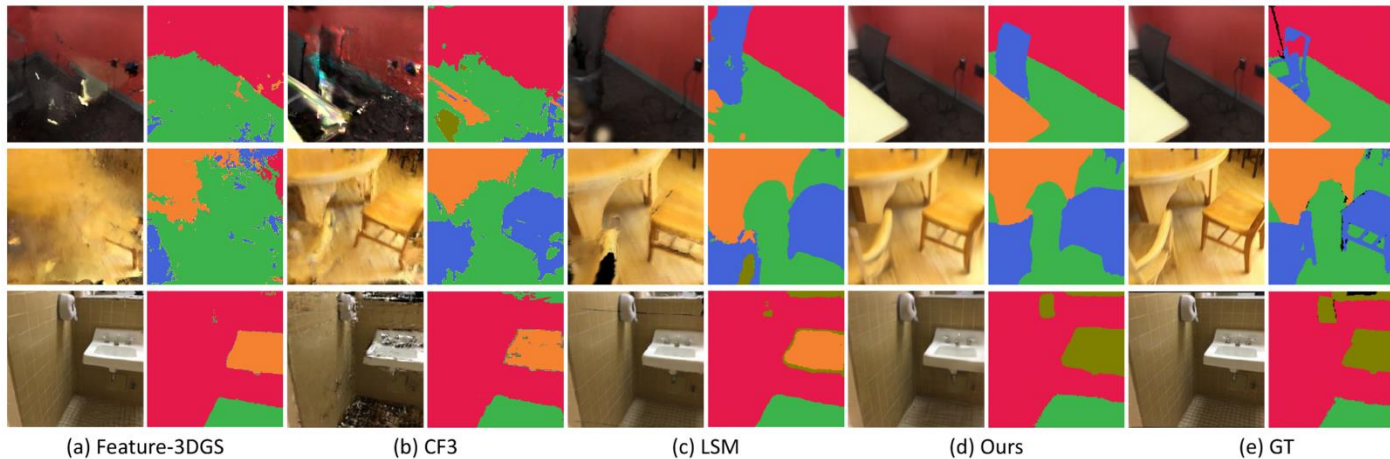


Figure 6. **Qualitative results of 3D scene understanding on ScanNet [10].** We conduct qualitative comparison for 3D scene understanding via novel view synthesis and open-vocabulary segmentation. When compared to both per-scene optimization ((a), (b)) and feed-forward ((c), (d)) methods, ours show the most high-fidelity renderings and accurate segmentation maps compared to the ground-truth.

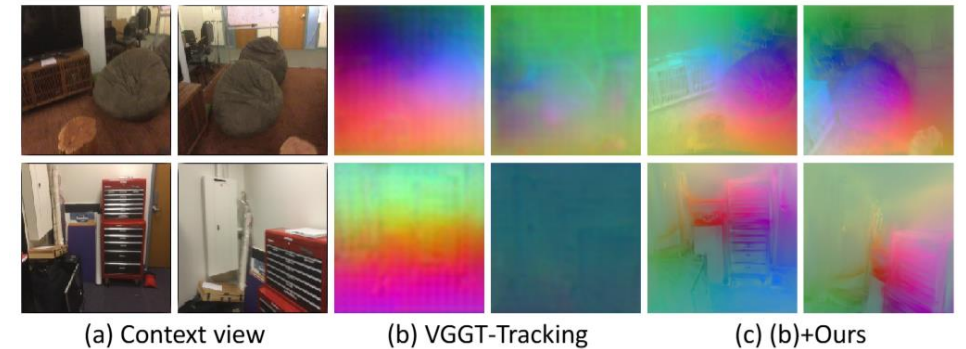
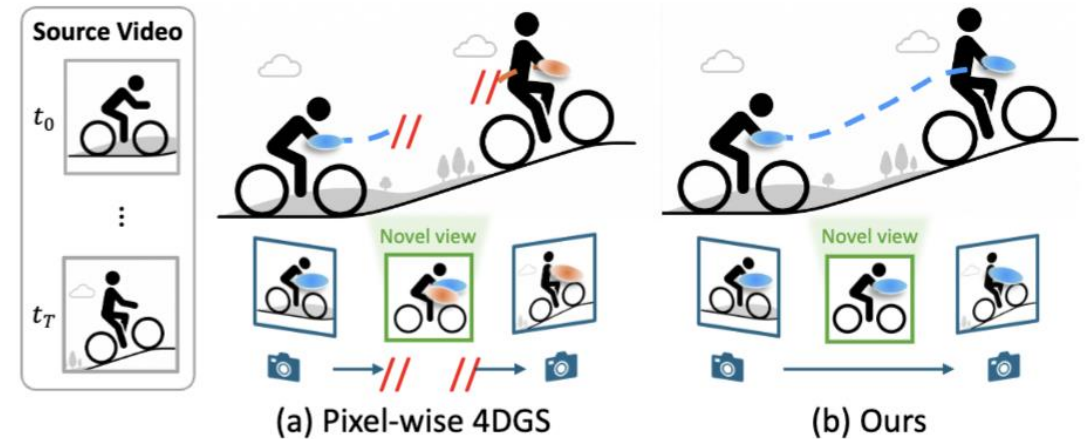
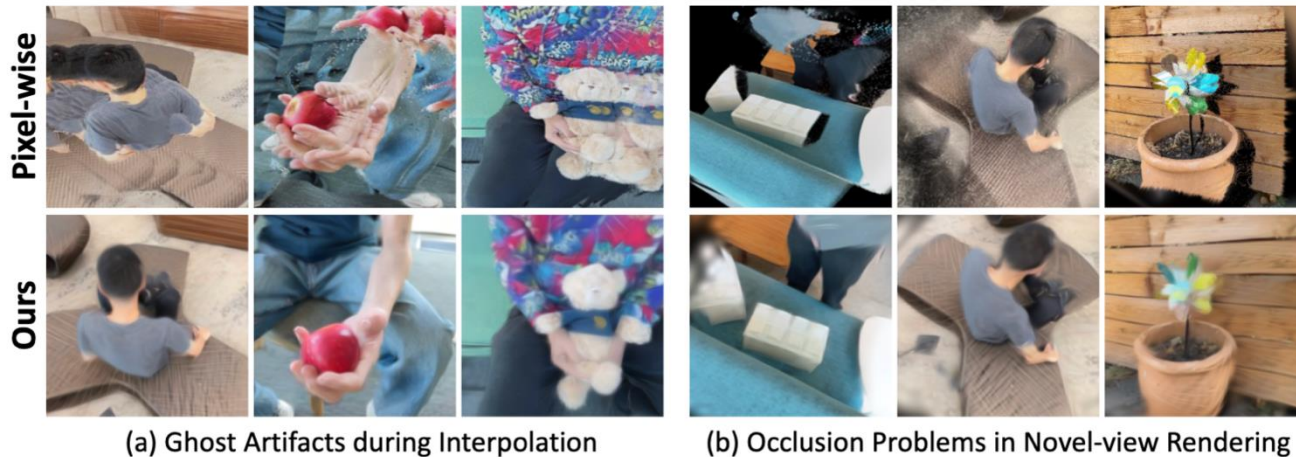


Figure 7. **PCA visualization of multi-view features on ScanNet [10].** We visualize the PCA results of encoded multi-view features. Our method improves multi-view consistency compared to the original visual features [63].

Extending to 4D

We now extending 3D reconstruction to 4D with handling dynamic

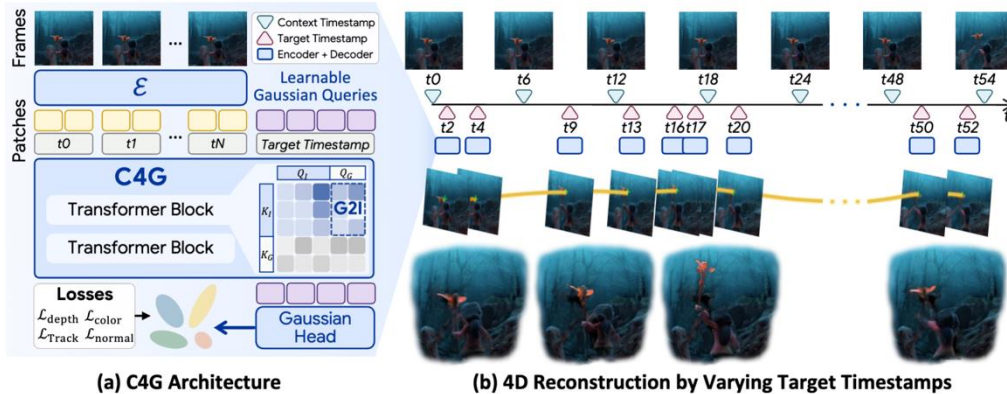
- Per-pixel 4DGS methods does not fully understand motion of dynamic objects.
- They are struggle to ghost artifacts due to the duplicated Gaussians and occlusion issues due to input-view bias.



Extending to 4D

We now extending 3D reconstruction to 4D with handling dynamic

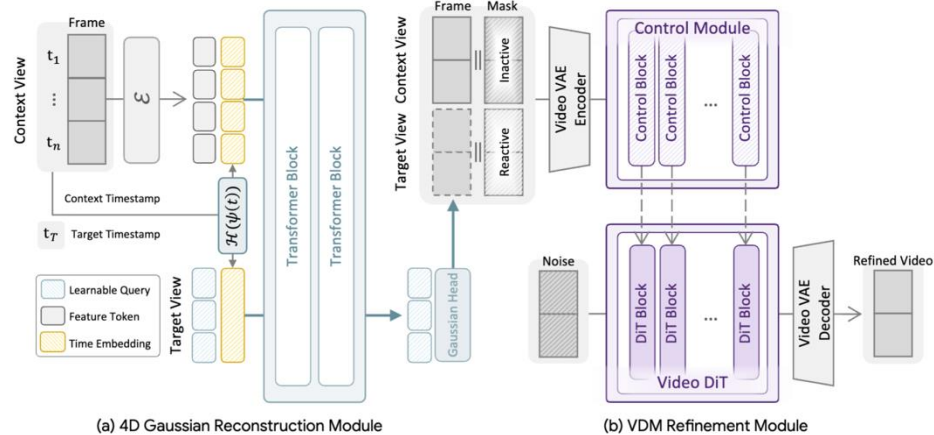
- Incorporating timestamp embedding for temporal understanding
- We additionally make it refinement module with using diffusion model.



C4G rendering

Ground truth

Diffusion results



Thank you!

honggyu@kaist.ac.kr
Computer Vision Laboratory (CVLAB)
Graduate School of Artificial Intelligence, KAIST