G3R: Gradient Guided Generalizable Reconstruction

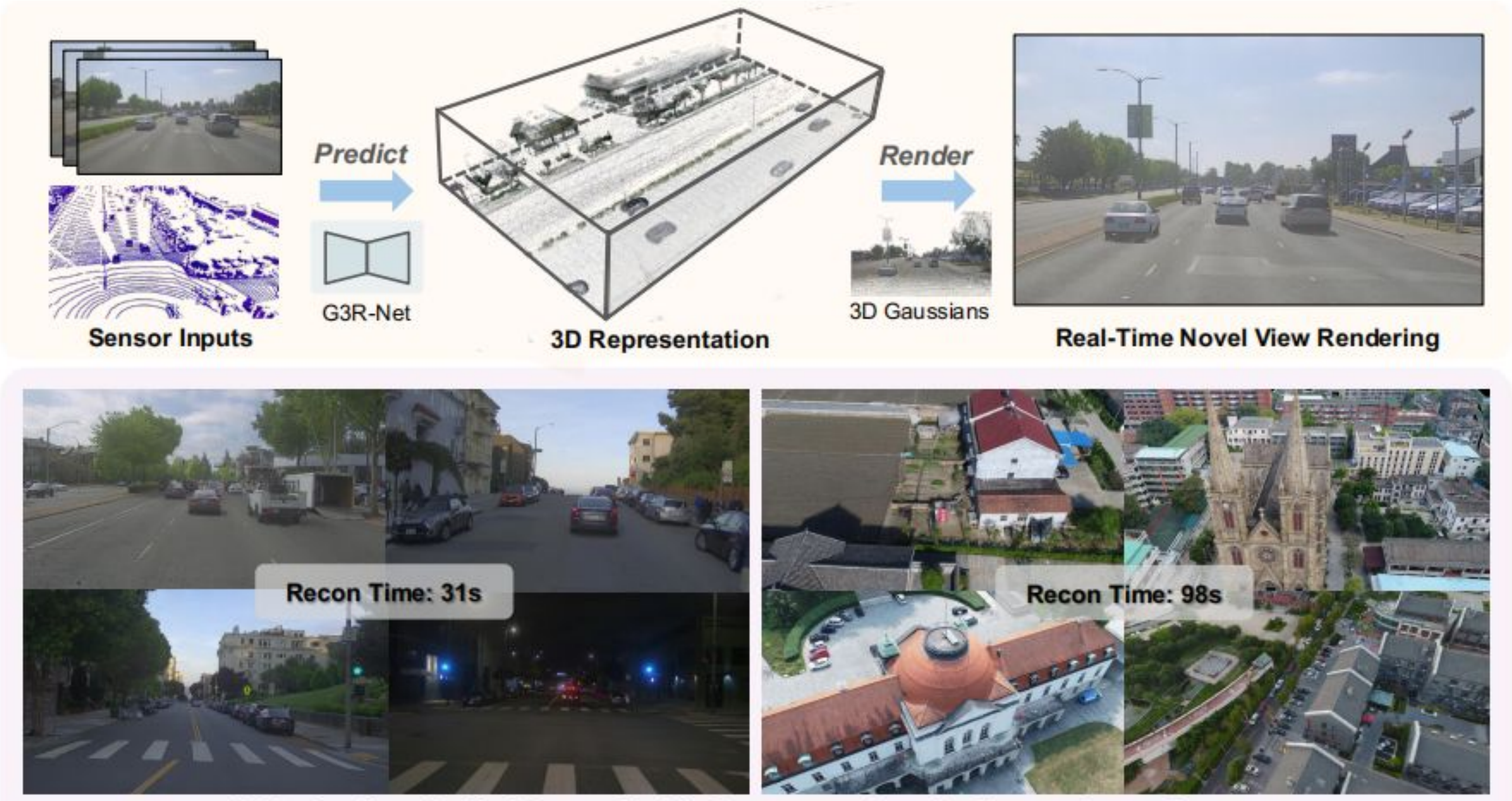
Yun Chen*, Jingkang Wang*, Ze Yang, Sivabalan Manivasagam, Raquel Urtasun

Waabi, University of Toronto

https://waabi.ai/g3r

Motivation: Generalizable Reconstruction

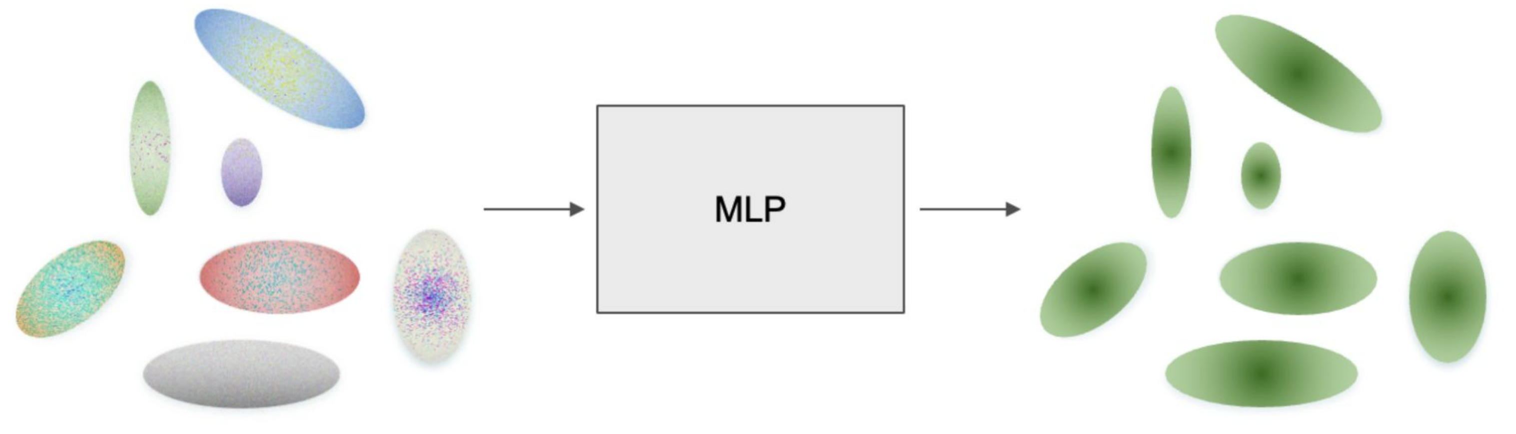
+ **Task**: Scalable reconstruction is important for simulation!
+ **Existing approaches**:
  + **Per-scene optimization (NeRF, 3DGS)** - costly, overfits to source
  + **Generalizable NVS/LRMs** - small scenes/objects, limited input views
+ **G3R**: (1) large dynamic scenes reconstructed in ~30s (2) arbitrary number of input images (3) more robust prediction for large view changes



G3R: Gradient Guided Generalizable Reconstructions for Unseen Large Scenes

Scene Representation

+ **3D Neural Gaussians**
  + Augment each 3D Gaussian with a latent feature vector
  + Provide additional representation capacity and easier prediction
  + MLP decodes 3D Neural Gaussians to 3D Gaussians



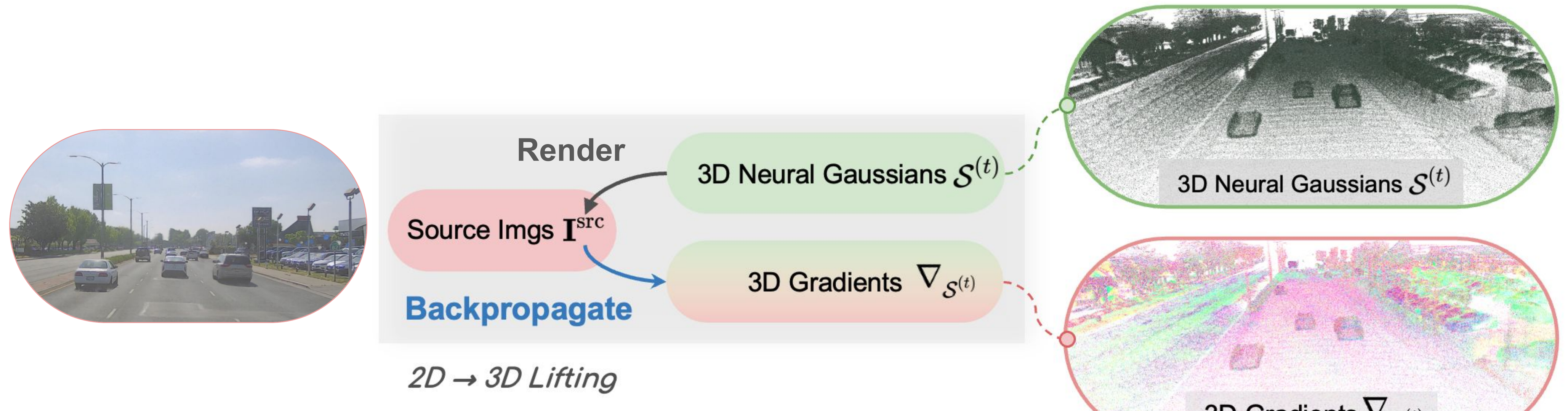
3D Neural Gaussians: *3D Gaussians + latent feature vector*

3D Gaussians: *size, rotation, color, opacity*

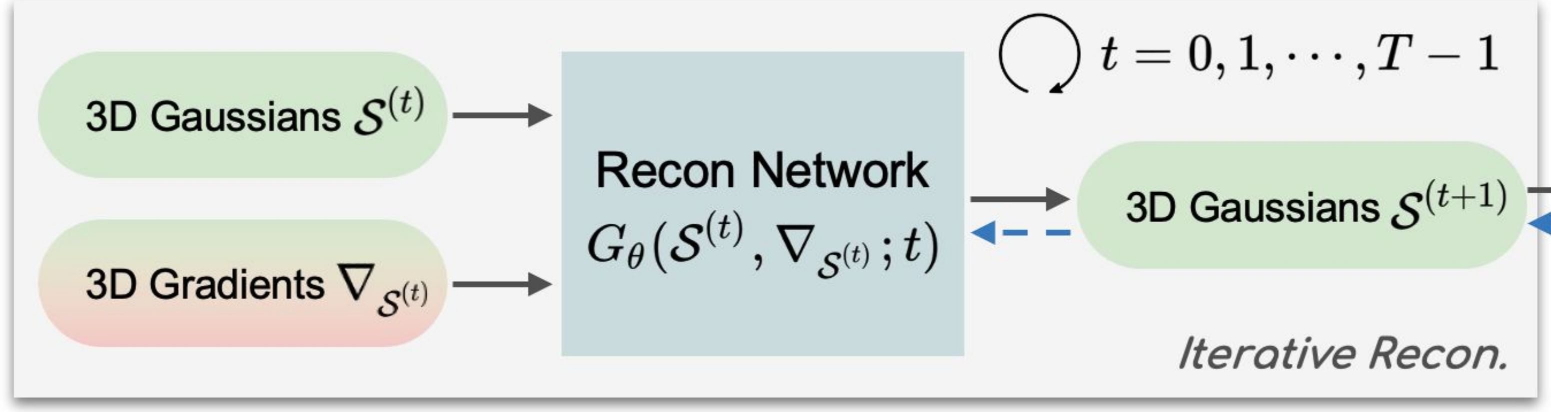
+ **Dynamic unbounded scene decomposition**
  + Static background, a set of dynamic actors, and a distant region for far-away buildings and sky.
  + Initialize 3D Neural Gaussians with LiDAR / multi-view stereo points

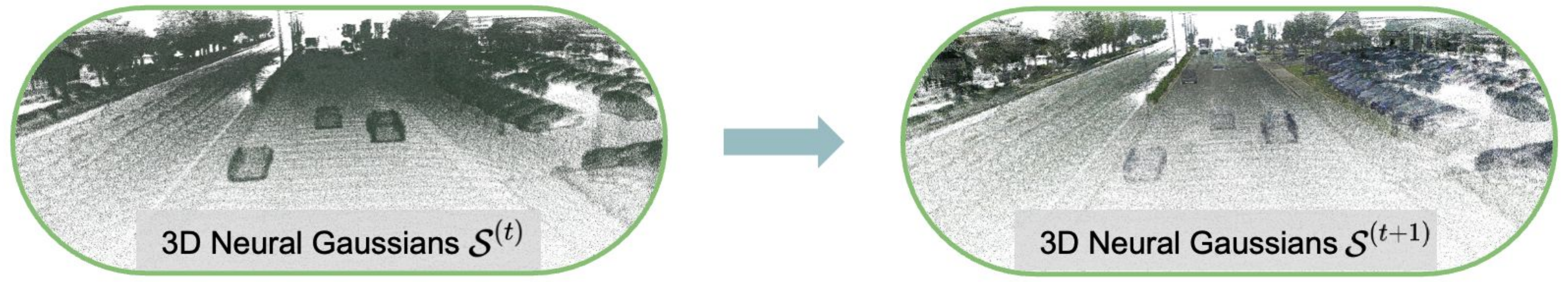
G3R

+ **G3R:** combines the benefits of fast feed-forward prediction methods with the iterative gradient feedback from per-scene optimization approaches
+ **Encode 2D Images in 3D as Gradients: "rendering and backpropagating"**
  + *Motivation*: Differentiable renderer bridges 2D and 3D
  + *Approach*: (1) render 3D representation to source views, (2) compute loss w.r.t. ground-truth images, (3) backpropagate to get 3D gradients, which encodes 2D info
  + *Why?* (1) a unified representation for multi-image aggregation, (3) occlusion-awareness in lifting 2D to 3D, (3) fast computation with 3DGS tile-rasterization



+ **Iterative Reconstruction with a Neural Network**
  + *Key idea*: Neural network as a learned optimizer for reconstruction
  + *Approach*: iteratively refine the 3D neural Gaussians for T steps
  + *Why?* overcome limited network capacity and diverse data distribution
  + Train with mix of source and target images
    + Increases robustness of predicted 3D representation at novel views





Training across many large outdoor scenes with a combination of photometric loss, perceptual loss, and a regularization term to ensure the flatness of 3D Gaussians.

$$\mathcal{L} = \mathcal{L}_{\mathrm{mse}}(\hat{\mathbf{I}}, \mathbf{I}) + \lambda_{\mathrm{lpips}}\mathcal{L}_{\mathrm{lpips}}(\hat{\mathbf{I}}, \mathbf{I}) + \lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}}(\mathcal{G})$$

Results

+ **Qualitative comparison with SOTA approaches**

Ground Truth | **ENeRF** | **PixelSplat** | **G3R (ours)**

+ **Quantitative comparison with SOTA**

		PSNR↑	Recon Time	FPS
Generalizable	ENeRF	24.43	0.057s†	6.93
	PixelSplat	23.21	0.74s†	147
Per-scene Opt.	Instant-NGP	24.34	7min 16s	3.24
	3DGS	25.14	50min 14s	121
Ours	G3R (turbo)	24.76	31s	121
	G3R	25.22	123s	121

+ **Ablation study**

Models	PSNR
Ours	**25.22**
− 3D neural Gaussian representation	24.72
− iterative reconstruction	20.03
− training with novel views	24.59
− update schedule $\gamma(t)$	25.03

+ **More robust results compared to 3DGS**

+ **Cross-dataset generalization (Pandaset→Waymo)**

+ **Limitations:** (a) artifacts in large extrapolations; (b) dense point initialization; (c) limited simulation controllability such as non-rigid motion and lighting