GSNeRF: Generalizable Semantic Neural Radiance Fields with Enhanced 3D Scene Understanding

[1]National Taiwan University, [2]NVIDIA


Zi-Ting Chou[1]


Sheng-Yu Huang[1]


I-Jieh Liu[1]


Yu-Chiang Frank Wang[1,2]

Neural Radiance Field (NeRF) (1/2)

- NeRF: Synthesizes novel views of complex 3D scenes from 2D images by representing the scene as neural networks.
- Input: Multi-view images of a scene
- Output: Novel-view image of the scene

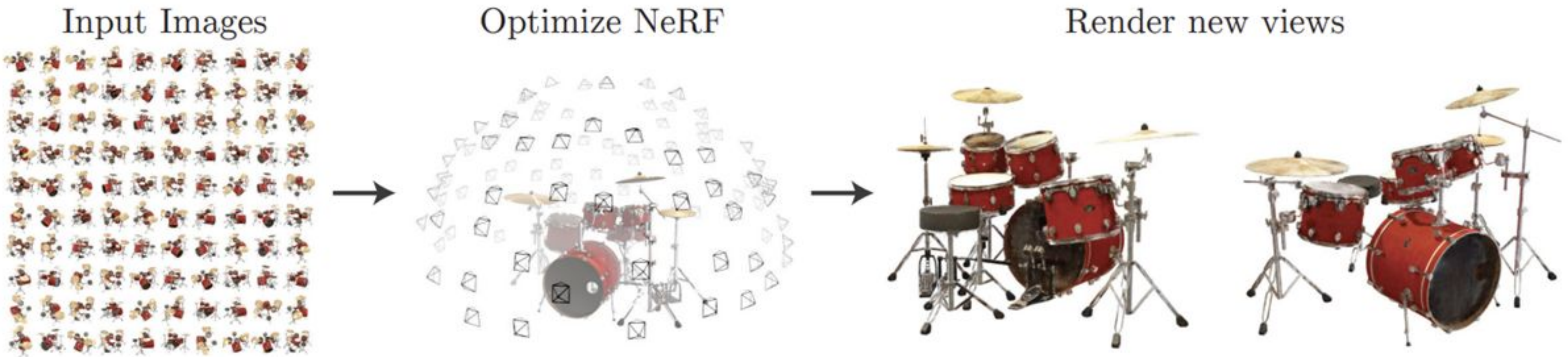
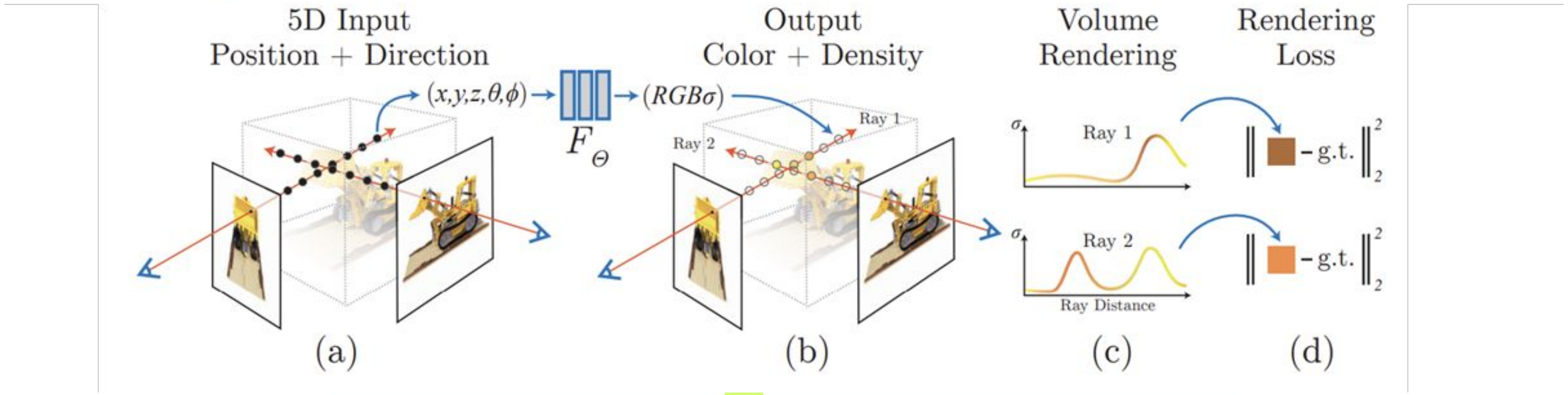


Figure from [link]

Neural Radiance Field (NeRF) (2/2)

- Core Process: Encodes spatial coordinates and viewing directions, outputs color and density, and applies volume rendering to produce images.



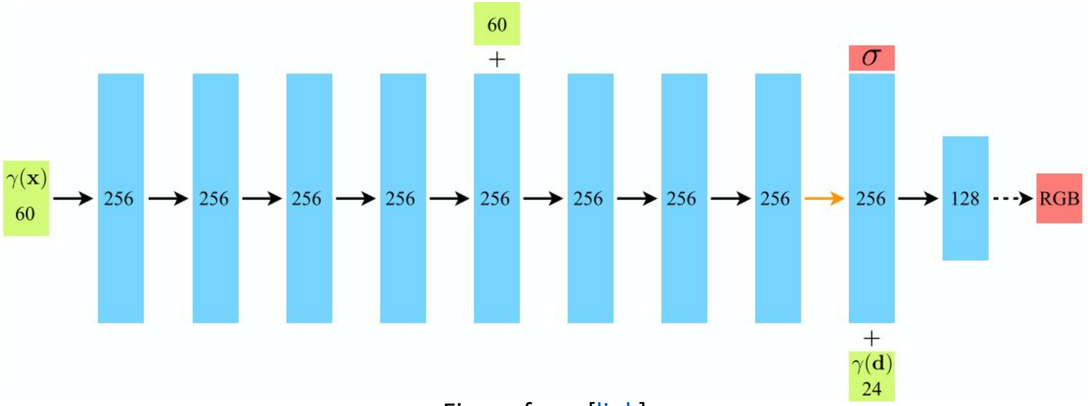


Figure from [link]

What is *Generalizable NeRF*?

Generalizable: one model weight for every scene

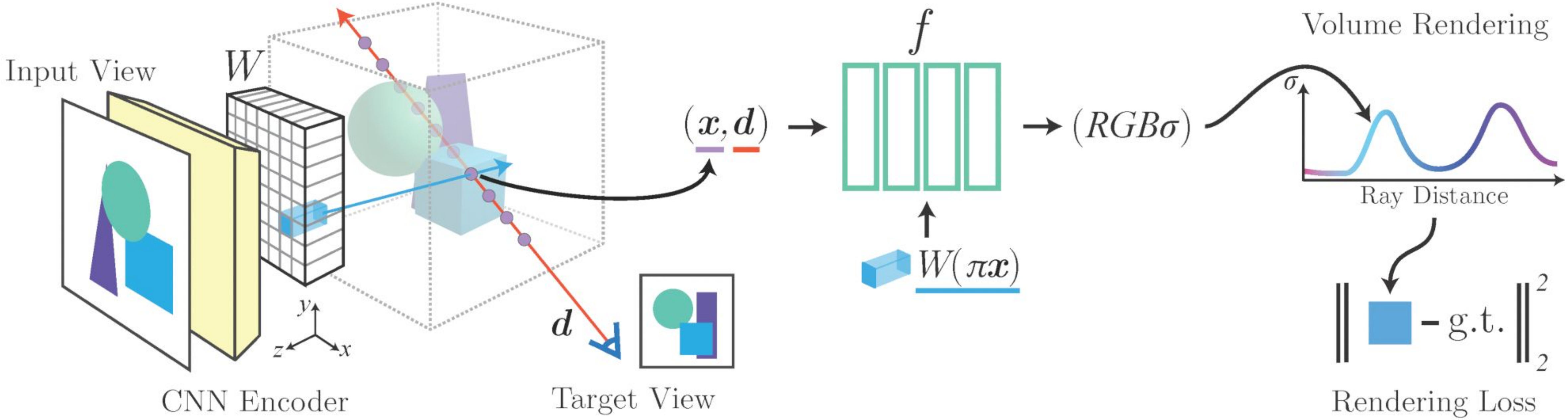
During Training (seen scene)

During Testing (unseen scene)

Input Output

How Generalizable NeRF works?

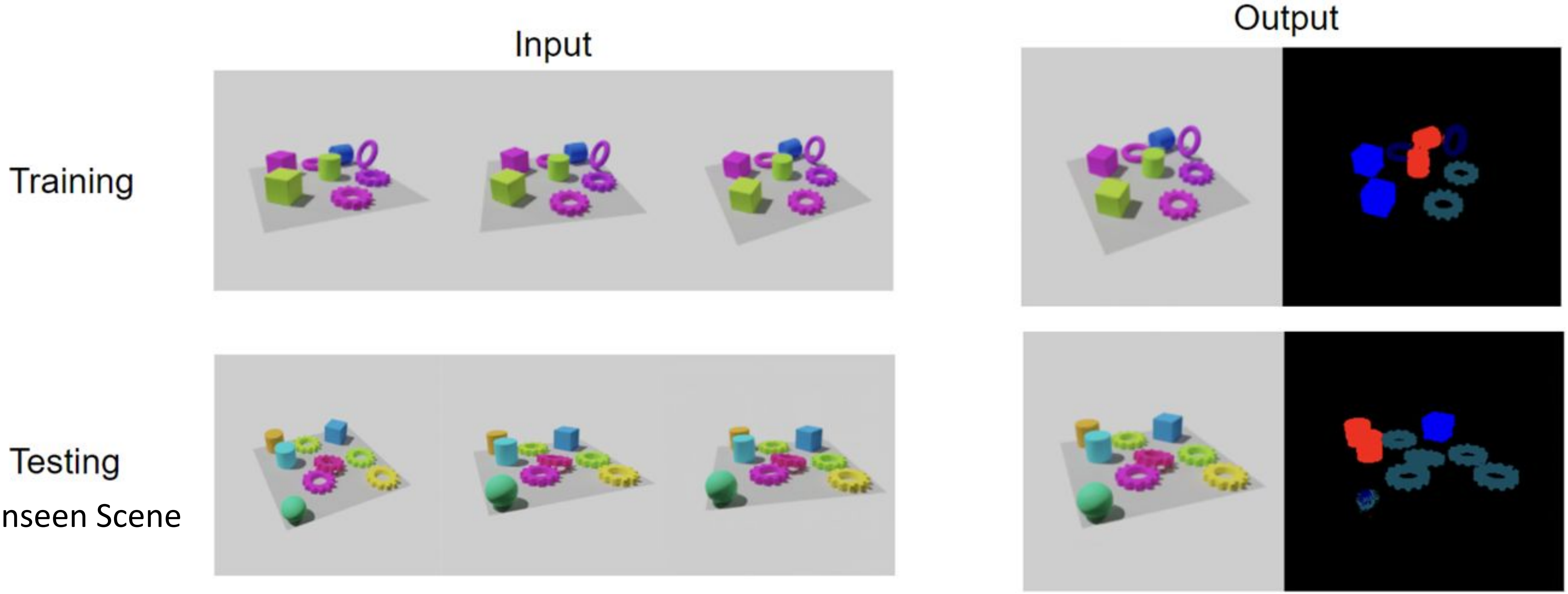
- pixel NeRF (CVPR'21)
  - Infers novel view of unseen scene from input images using pixel-aligned features.



[pixel NeRF website]

Our Task

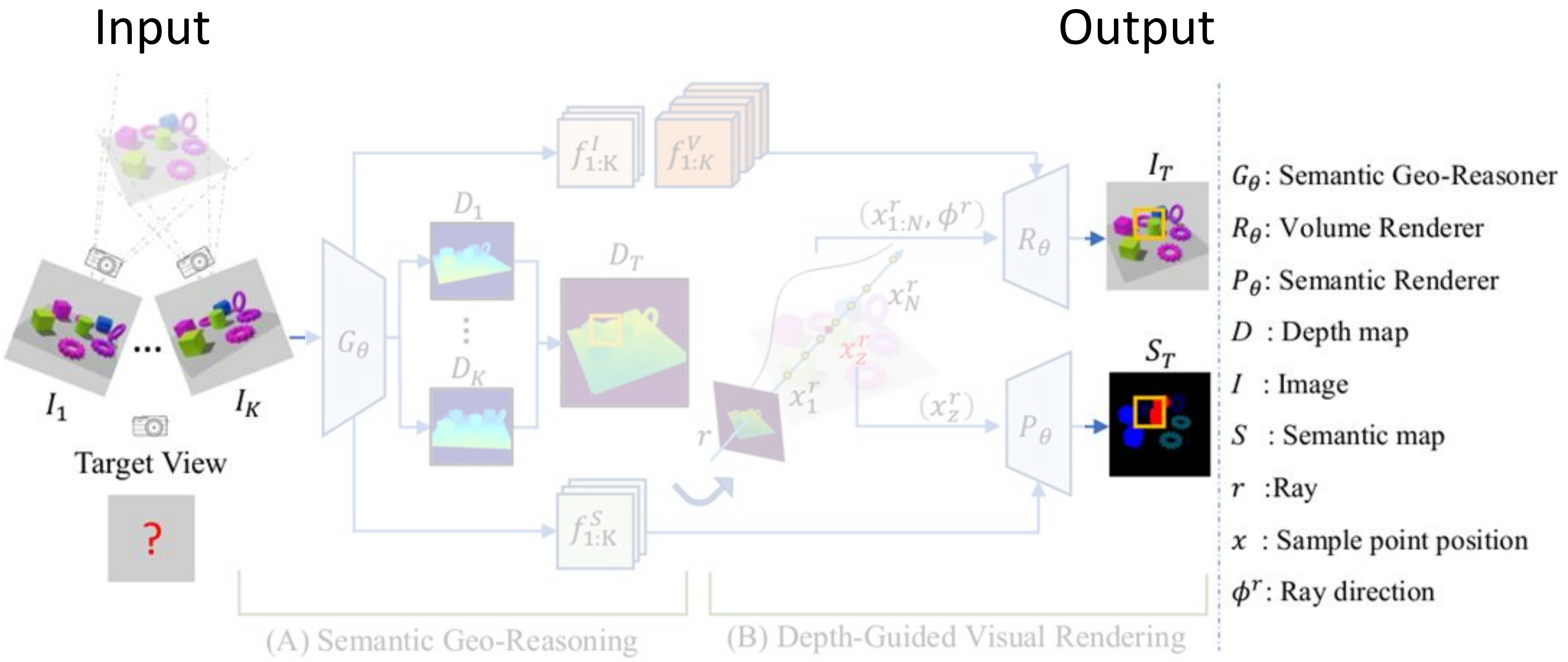
- Enable the generalizable NeRF with novel view semantic segmentation ability.



Unseen Scene

Method

– Generalizable Semantic Neural Radiance Fields with Enhanced 3D Scene Understanding



Method

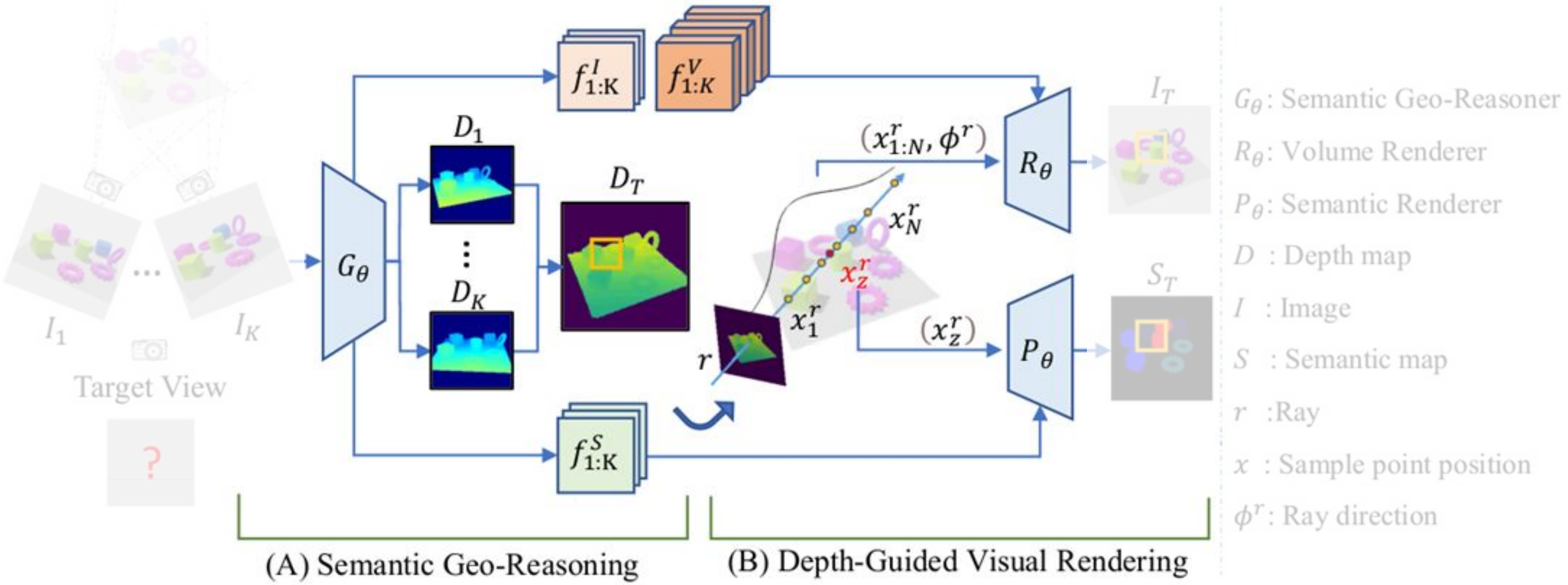
– Generalizable Semantic Neural Radiance Fields with Enhanced 3D Scene Understanding

A. Semantic Geo-Reasoning

- Extract semantic and geometry features from a scene.

B. Depth-Guided Visual Rendering

- Utilize the extracted geometric information to perform depth-guided image and semantic rendering.

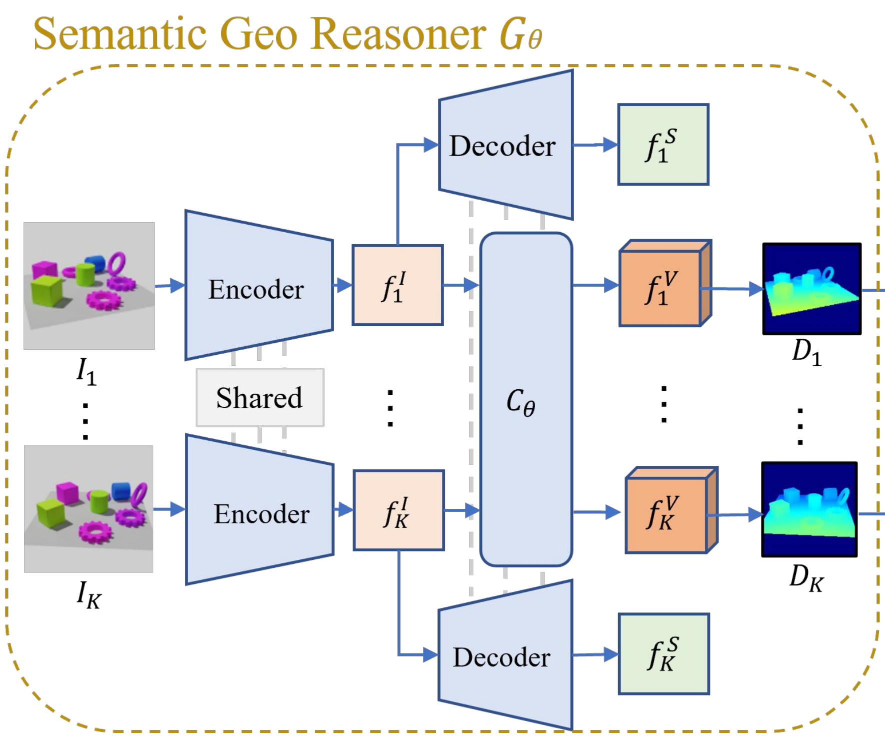


Method

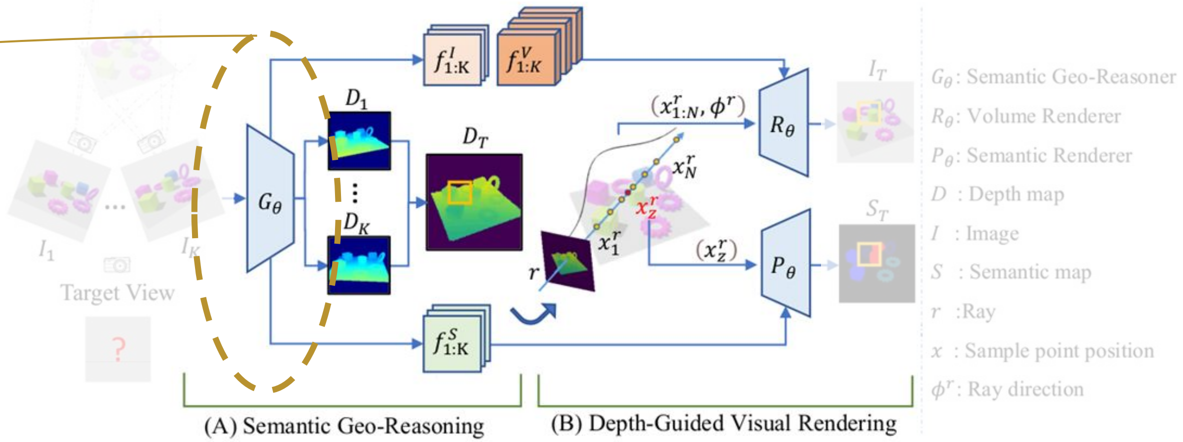
– Generalizable Semantic Neural Radiance Fields with Enhanced 3D Scene Understanding

A. Semantic Geo-Reasoning

- Extract semantic and geometry features from a scene.



Novel-view depth estimation



Depth Regularization:

1. Supervised: $\mathcal{L}_D = \frac{1}{K}(\sum_{k=1}^{K} \|D_k - \hat{D}_k\|_{s1})$

2. Self-Supervised: $\mathcal{L}_{ssl} = \lambda_1 \mathcal{L}_{RC} + \lambda_2 \mathcal{L}_{SSIM} + \lambda_3 \mathcal{L}_{Smooth}$

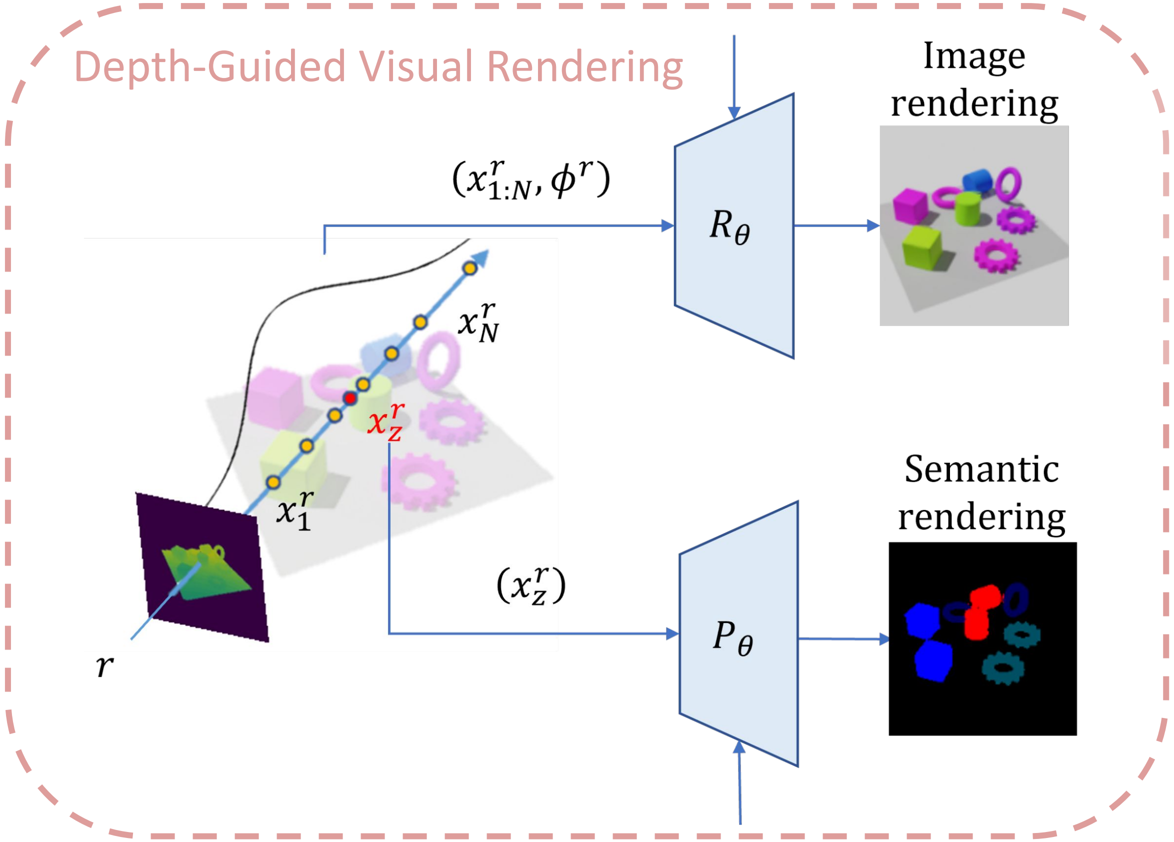
ref: RCMVSNet

Method

– Generalizable Semantic Neural Radiance Fields with Enhanced 3D Scene Understanding

B. Depth-Guided Visual Rendering

- Utilize the extracted geometric information to perform depth-guided image and semantic rendering.



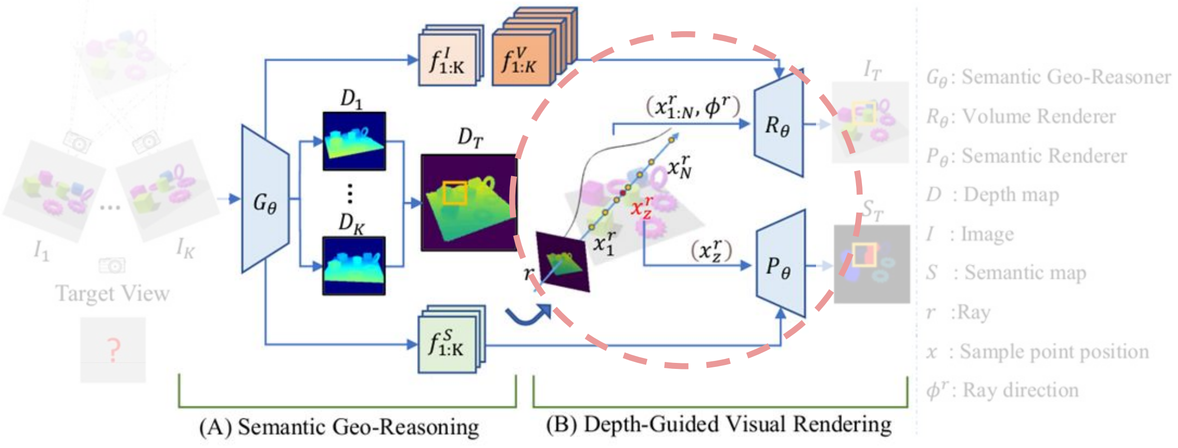


Image rendering loss: L2 loss $\mathcal{L}_{image} = \sum_{r \in R} \|\mathbf{C}(r) - \hat{\mathbf{C}}(r)\|_2^2$

Semantic loss: Cross-entropy loss $\mathcal{L}_{sem} = \sum_{r \in R} (\mathbf{S}(r) \log \hat{\mathbf{S}}(r))$

Experiment

– Quantitative Evaluation

fig: ScanNet

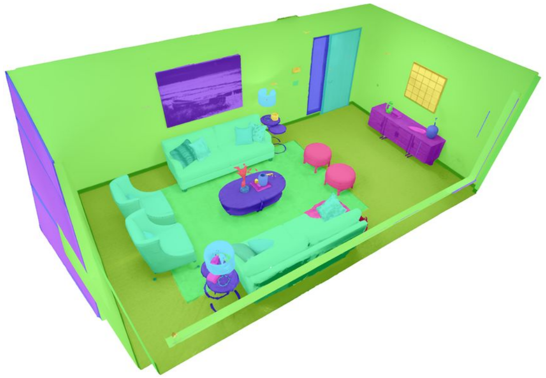
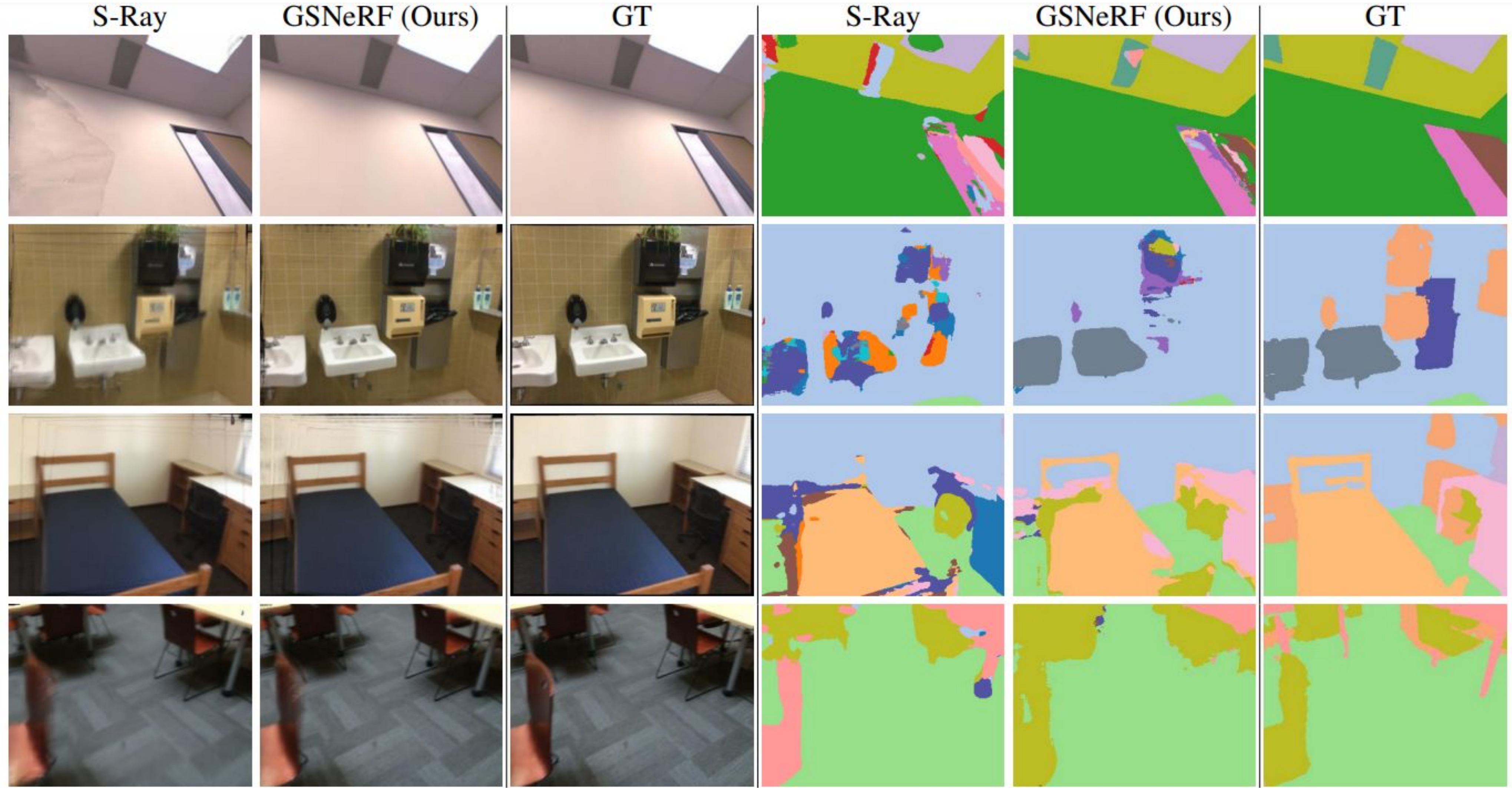


fig: Replica

- ScanNet & Replica Datasets
  - ScanNet: Real-world 3D indoor scene dataset.
  - Replica: Synthetic 3D indoor scene dataset.
- Experimentation
  - S-Ray (CVPR '23) uses multi-view GT depth as input. Therefore, we conduct experiments on our method with and without depth supervision.

Generalized method	GT Depth	ScanNet [5]					Replica [30]				
	Train / Test	mIoU	acc. / class acc.	PSNR↑	SSIM↑	LPIPS↓	mIoU	acc. / class acc.	PSNR↑	SSIM↑	LPIPS↓
Neuray [21] + semhead	✔ / ✔	52.09	67.81 / 61.98	25.01	83.07	31.63	44.37	79.93 / 54.25	26.21	87.37	30.93
GeoNeRF [16] + semhead	✔ /	53.78	76.18 / 61.90	**32.55**	**90.88**	12.69	45.12	81.67 / 52.36	28.70	88.94	20.42
S-Ray [20]	✔ / ✔	55.53	77.79 / 60.92	25.19	83.66	30.98	45.30	80.48 / 53.72	26.38	88.13	30.04
GSNeRF (Ours)	✔ /	**58.30**	**79.79 / 65.93**	31.33	90.73	**12.53**	**51.52**	**83.41 / 61.29**	**31.16**	**92.44**	**12.54**
MVSNeRF [3] + semhead		43.06	66.90 / 53.63	24.14	80.36	34.63	30.21	69.35 / 39.75	23.68	84.37	28.08
GeoNeRF [16] + semhead		45.11	67.12 / 53.44	30.75	88.27	16.48	40.35	74.63 / 49.18	29.92	91.14	17.60
GNT [36] + semhead		43.49	62.06 / 51.84	24.39	82.37	28.36	38.14	71.44 / 47.46	24.56	87.31	20.97
Neuray [21] + semhead		46.09	66.39 / 53.79	25.24	84.39	31.33	40.91	76.23 / 50.15	27.80	89.55	23.68
S-Ray [20]		47.69	64.90 / 54.47	25.13	84.18	30.44	43.27	77.63 / 52.85	26.77	88.54	22.81
GSNeRF (Ours)		**52.21**	**74.71 / 60.14**	**31.49**	**90.39**	**13.87**	**51.23**	**83.06 / 61.10**	**31.71**	**92.89**	**12.93**

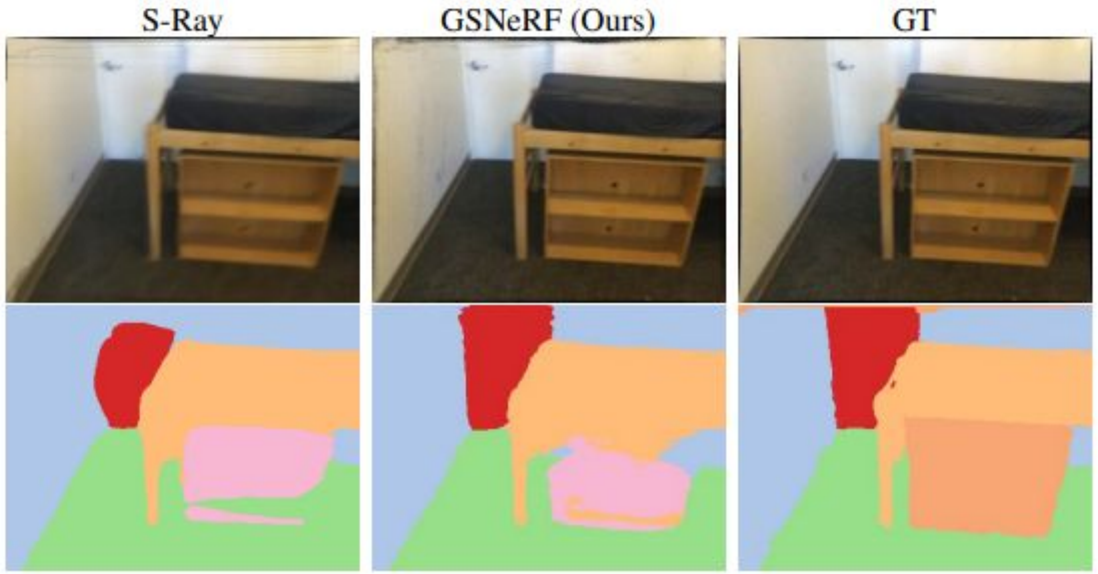
Experiment
– Qualitative Evaluation (generalized setting)



Experiment
– Test time fine-tuning

- Although our primary focus is on generalizability, we also conduct fine-tuning for both qualitative and quantitative experiments on the ScanNet dataset.
  - Generalized Setting: Testing on novel scenes that were not seen during training.
  - Fine-tune Setting: Fine-tuning on test scenes for 5k steps (~ 20 minutes) before evaluation.

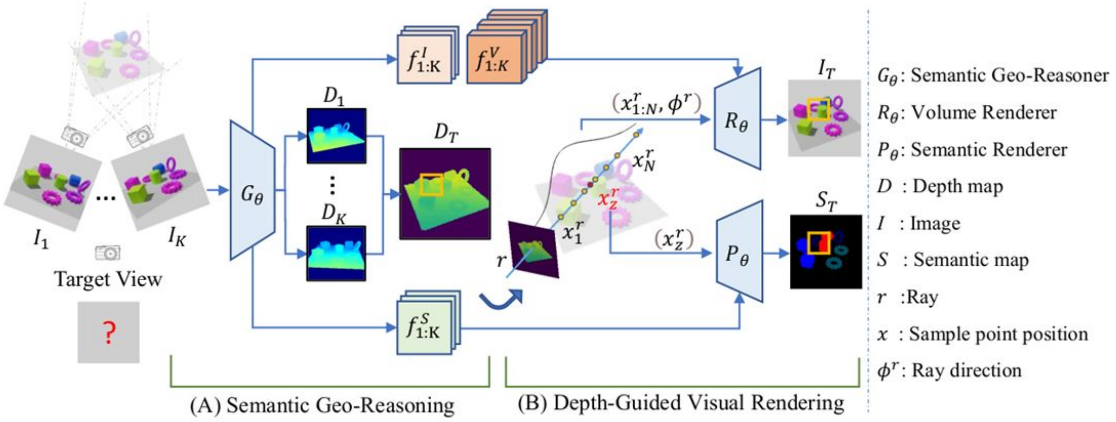
Finetuned Method	GT Depth	ScanNet		
	Train / Test	mIoU	acc. / class acc.	PSNR
S-Ray	✔ / ✔	92.4	98.2 / 93.8	27.67
Ours	✔ /	**93.9**	**99.1 / 98.4**	**31.70**
S-Ray	/	91.6	97.3 / 92.2	27.31
Ours	/	**93.2**	**98.2 / 96.8**	**30.89**

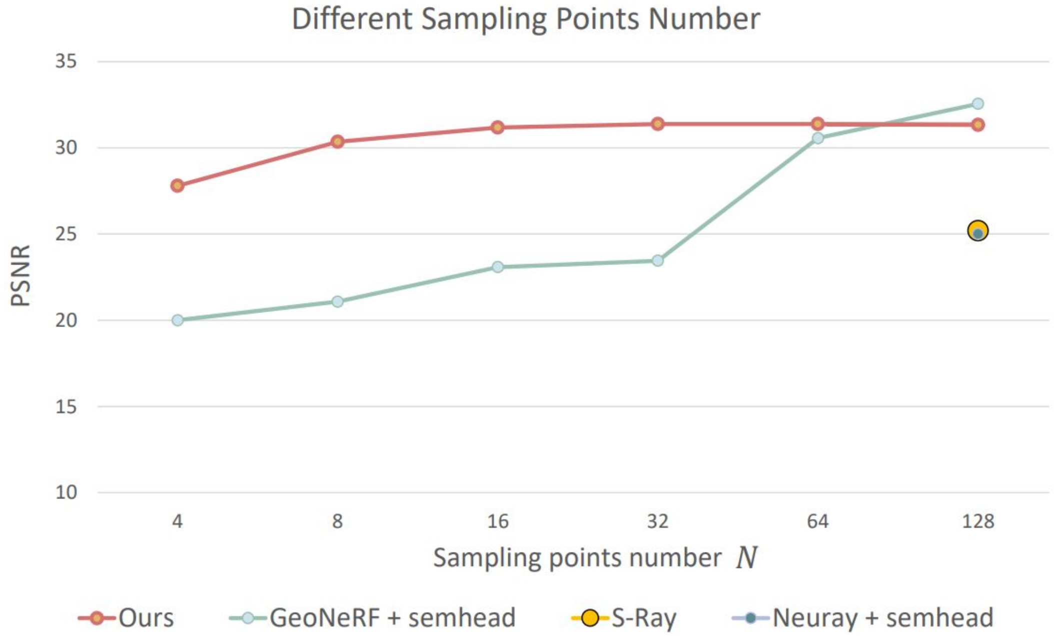


Experiment

– Analysis of GSNeRF

- Sampling Efficiency (on **ScanNet dataset**)
  - Thanks to our depth-guided sampling strategy, the number of sampling points (for image rendering) can be reduced during inference, without compromising segmentation performance.
  - 4x rendering speed with better image and segmentation quality.

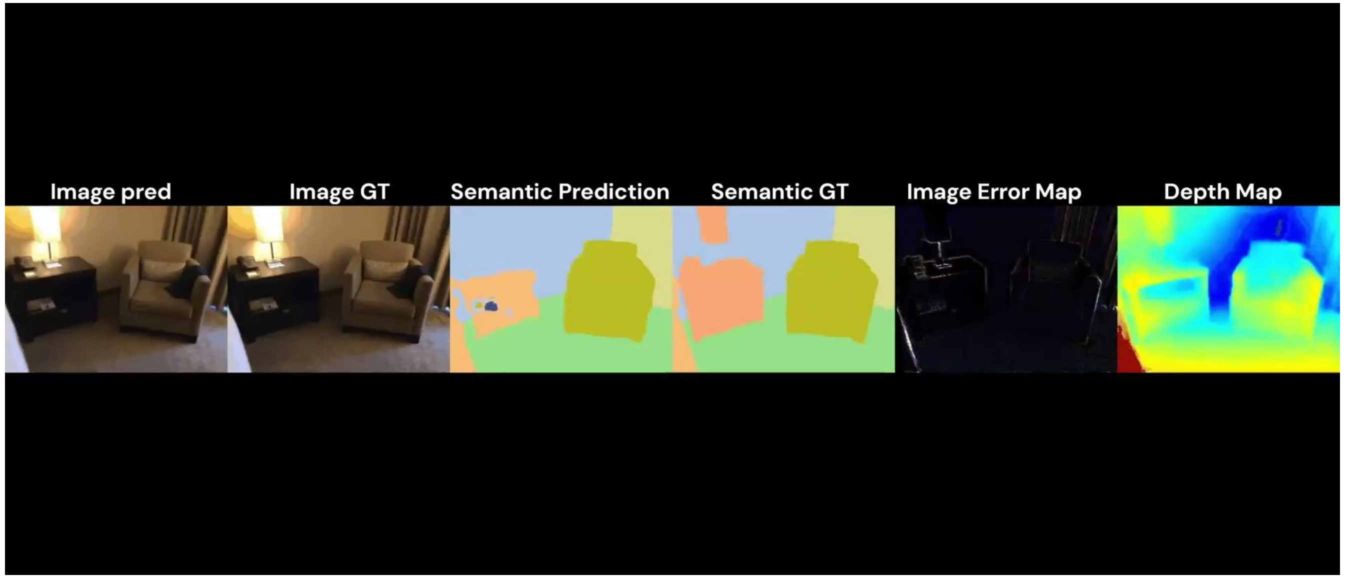


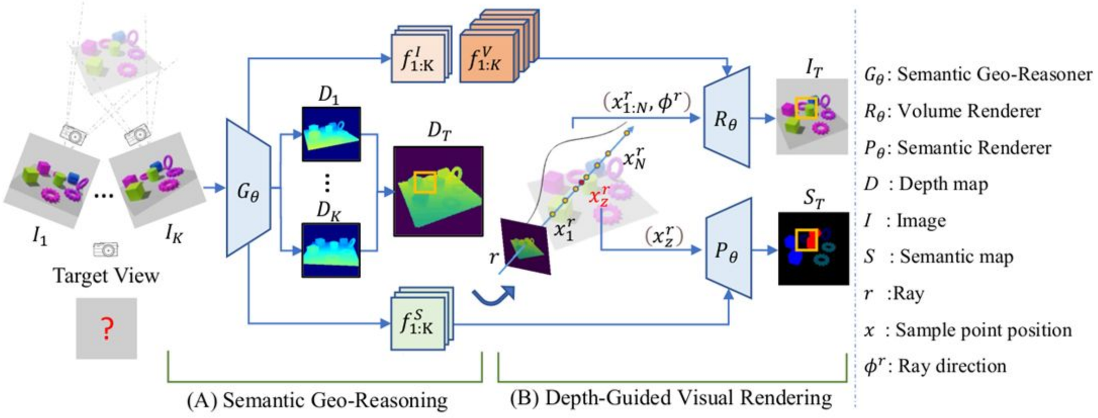


	N	FPS↑	PSNR↑	mIoU↑
S-Ray	128	0.16	25.13	47.69
Ours	128	0.11	31.49	52.21
Ours	4	0.84	27.80	52.21

Conclusion

- Introducing Generalizable Semantic Neural Radiance Fields (GSNeRF) for **simultaneously novel view synthesis and semantic segmentation**.
- Propose innovative depth estimation and **depth-guided visual rendering**, outperforms existing methods on real-world and synthetic datasets.



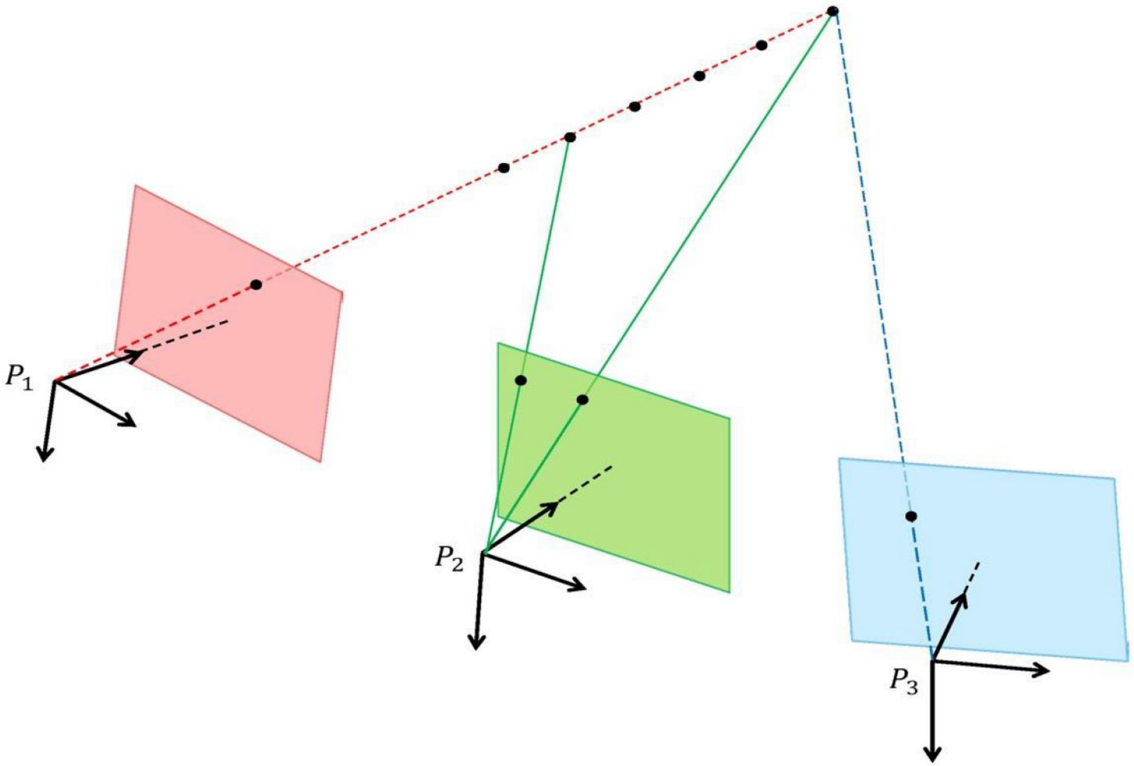


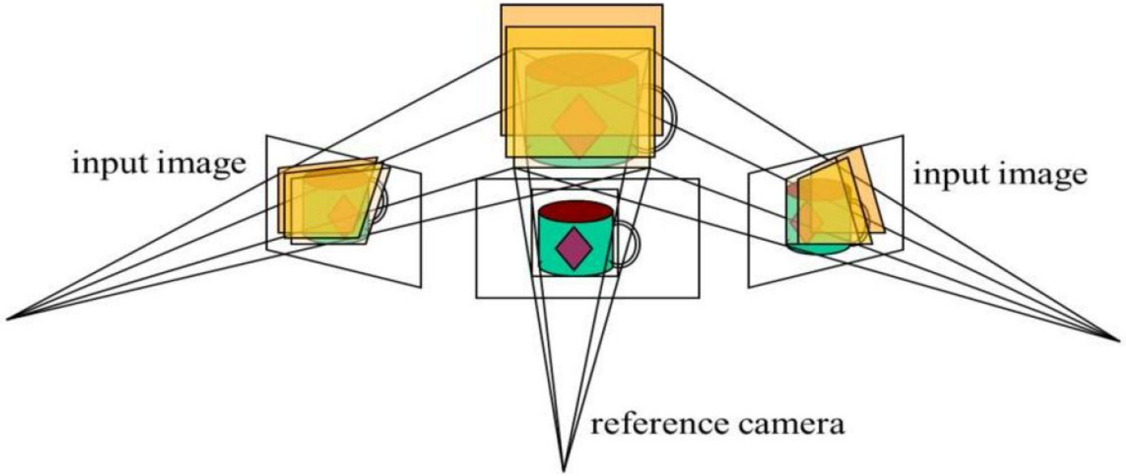
Thanks for your attention!

Backup Slides

MVS – Cost Volume

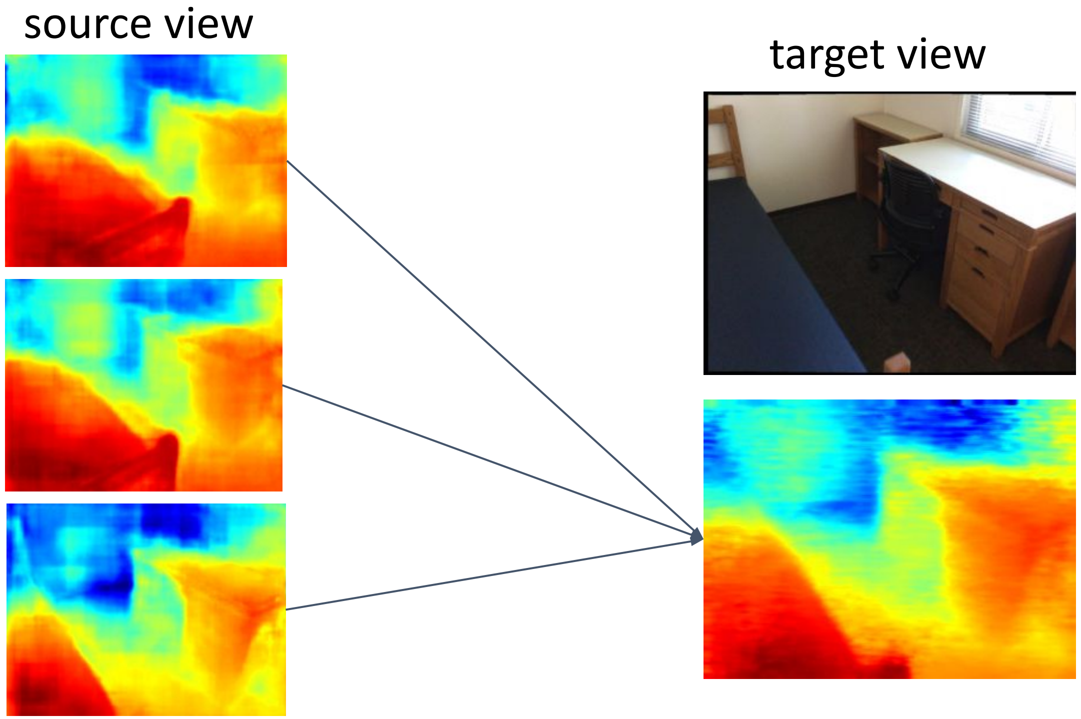
- Cost volume is constructed by variance across pixels (of different images)





Algorithm

- Target (Novel) view depth estimation
  - With multi-view depth estimation
  - Projecting all depth predictions into 3D space
  - Reprojecting onto the target camera



Algorithm 1 Estimation of Depth in Target View

Input: Depth predictions of each source view $D_{1:K}$, camera pose of each source view $\xi_{1:K}$, target camera pose ξ_T
Data: Image size: (H, W), camera pose of the world coordinate ξ_w

Output: Target view depth estimation D_T

$A \leftarrow$ empty array()
for k = 1, ..., K **do**
  $g \leftarrow$ meshgrid(H, W)
  Project g into the coordinate system defined by ξ_k
  Multiply g by the corresponding depth prediction D_k

  $g \leftarrow$ **Transform**(g, ξ_k, ξ_w)
  Append g to the array A
end for
$A \leftarrow$ **Transform**(A, ξ_w, ξ_T)
Reproject A onto the ξ_T image plane
$Z \leftarrow$ the third element (Z-axis) of points A
$A' \leftarrow$ round the first two elements of A to integer values

$W \leftarrow$ The first two elements of (A' - A)
Weight and normalize Z using weight W
Set the depth of target view D_T to Z based on the index of the first two elements of A'
return Estimated depth of target view D_T

/ Function */*
Transform(point, ξ_1, ξ_2):
return transform point from coordinate ξ_1 to ξ_2

Training Loss

- Image rendering loss: L2 loss
- Semantic loss: Cross-entropy loss
- depth loss:
  - supervised:
  - self-supervised:

$$\mathcal{L}_{image} = \sum_{r \in R} \|\mathbf{C}(r) - \hat{\mathbf{C}}(r)\|_2^2$$

$$\mathcal{L}_{sem} = \sum_{r \in R} (\mathbf{S}(r) \log \hat{\mathbf{S}}(r))$$

$$\mathcal{L}_D = \frac{1}{K} (\sum_{k=1}^{K} \|D_k - \hat{D}_k\|_{s1})$$

$$\mathcal{L}_{ssl} = \lambda_1 \mathcal{L}_{RC} + \lambda_2 \mathcal{L}_{SSIM} + \lambda_3 \mathcal{L}_{Smooth}$$

ref: [RCMVSNet]

With GT depth supervision: $\mathcal{L} = \mathcal{L}_{image} + \mathcal{L}_D + \lambda \mathcal{L}_{sem}$

Without GT depth supervision: $\mathcal{L} = \mathcal{L}_{image} + \mathcal{L}_{ssl} + \lambda \mathcal{L}_{sem}$

Metrics

- PSNR: $$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$
- SSIM: $$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$
- LPIPS: $$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} ||w_l \odot (\hat{y}_{hw}^l - \hat{y}_{0hw}^l)||_2^2$$

ref: [Link]()