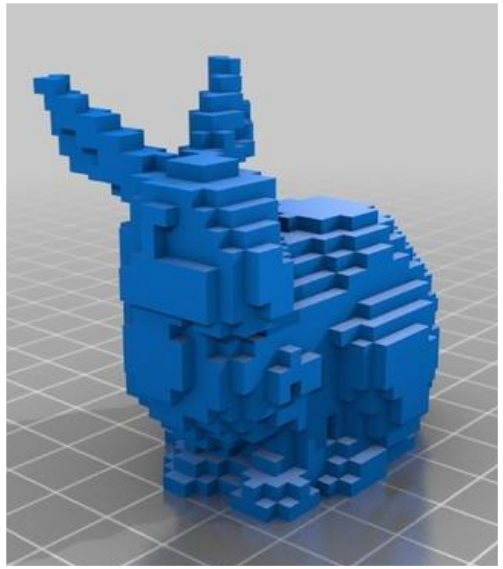
3D Editing and Online Perception with Neural Radiance Fields

Zhaopeng Cui

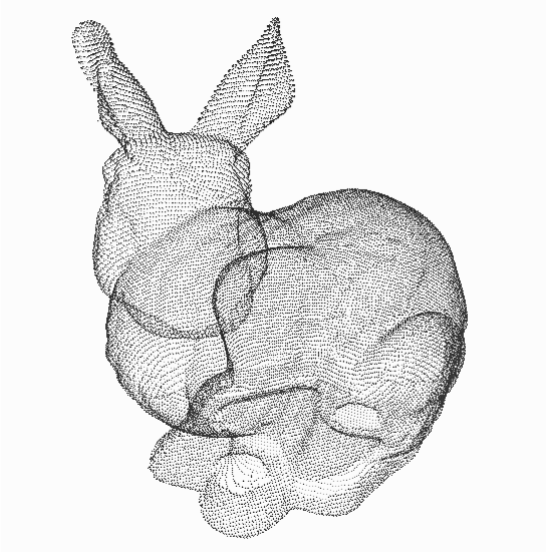
Zhejiang University

Explicit 3D Representation

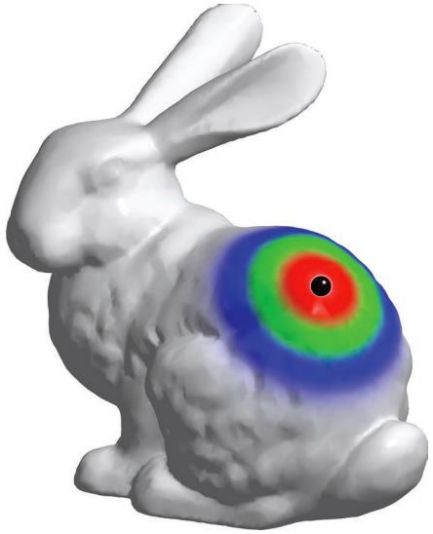
- ✓Structured geometry
- ✓Discrete and well-defined
- ✓Visually intuitive



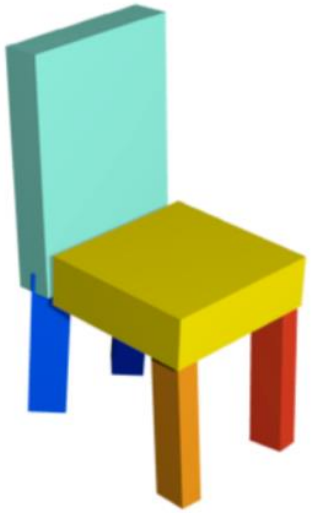
Volumetric



Point Cloud



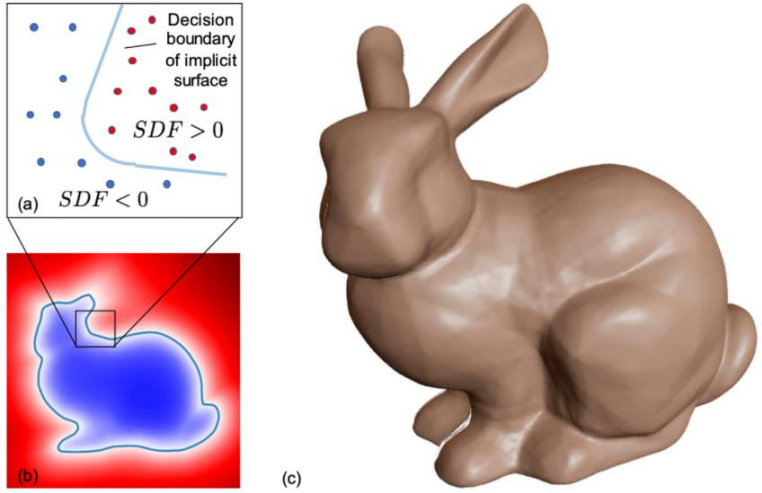
Mesh



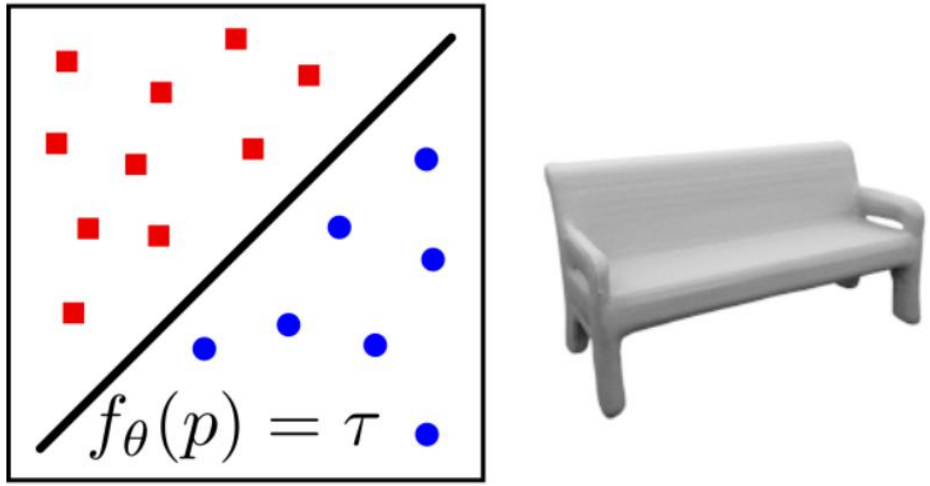
Part Assembly

Implicit Neural Representation

- ✓Infinite-resolution
- ✓Lightweight
- ✓Continuity and differentiability



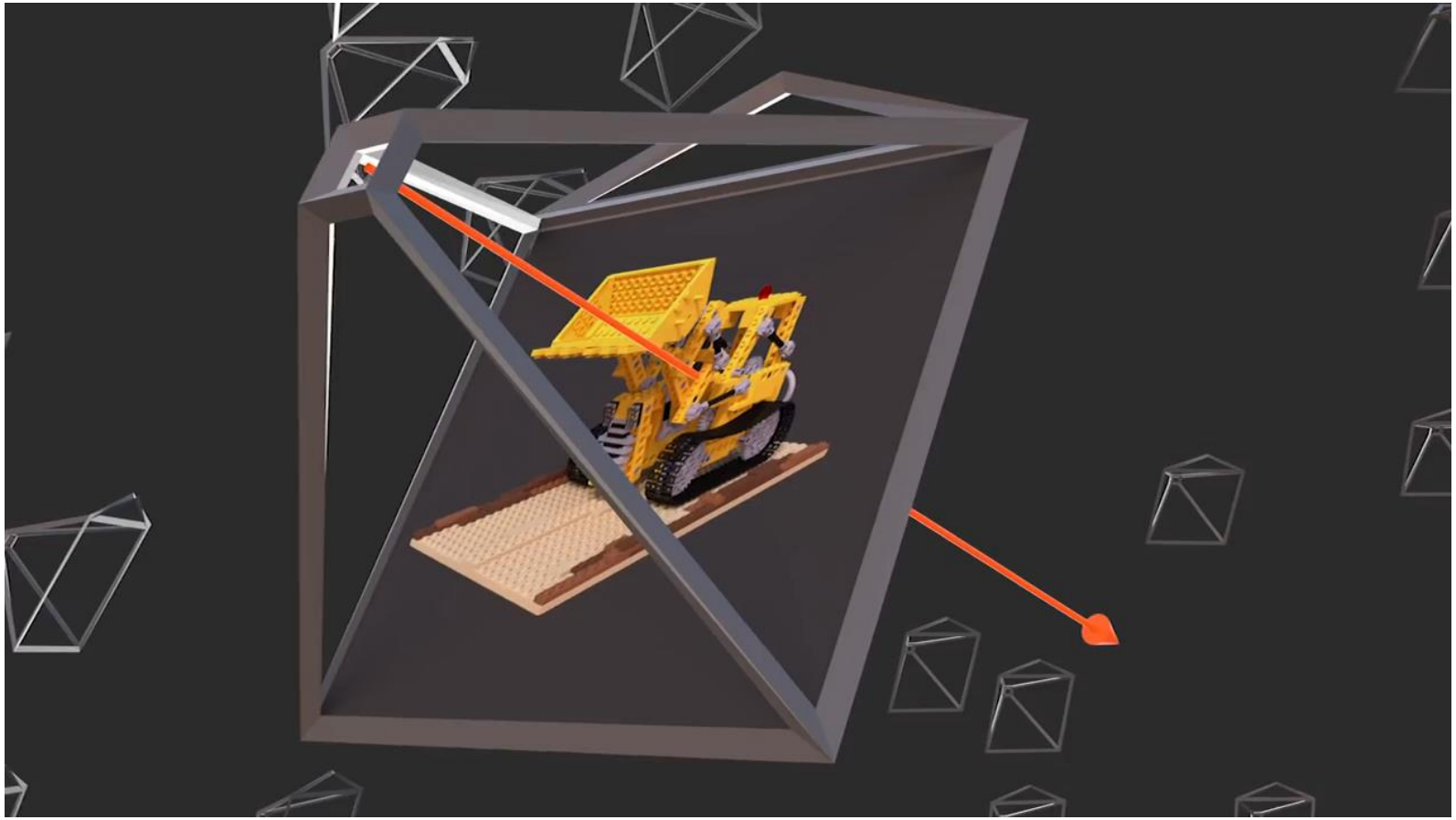
DeepSDF
[Park et al., 2019]



Occupancy Networks
[Mescheder et al., 2019]

Neural Radiance Fields

- Neural Radiance Fields (NeRF) can represent a scene as the weights of an MLP, trained on many images with known camera poses.



Courtesy of Ben Mildenhall, Pratul P. Srinivasan and Matthew Tancik

NeRF: Representing Scenes as Neural Radiance Fields, Mildenhall et al. ECCV 2020

Neural Radiance Fields

- NeRF has attracted extensive attention in both academy and industry.

Nerf: Representing scenes as neural radiance fields for view synthesis

B Mildenhall, PP Srinivasan, M Tancik… - Communications of the …, 2021 - dl.acm.org

We present a method that achieves state-of-the-art results for synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a …

☆ Save 99 Cite Cited by 3416 Related articles

NeRF Explosion 2020

21 minute rea

Publish

NeRF at ICCV 2021

12 minute read

Published: (

In anticipation c
use Neural Radi
Many of the pa
sheer number o
keep up. The to
to cut down on
program. I shar

NeRF at CVPR 2022

13 minute read

Published

There are mor
conference. W
we rounded u
reading pleasu

Below are all t
associated pa
any mis-chara
twitter.

NeRF at CVPR 2023

Mark Boss, Frank Dellaert

May 1, 2023 · 49 min read · NeRF, Literature Review

It is now my third time writing a summary of NeRFy things at a conference. This time it is the big one: CVPR. The list of accepted papers is massive again, with 2359 papers.

What is even more astounding is that the number of NeRF papers has grown significantly. I scanned the provisional program for potential NeRF titles and manually confirmed a relationship to the NeRF field.

NeRF: Neural Radiance Field in 3D Vision, Introduction and Review

Kyle (Yilin) Gao, *Graduate Student Member, IEEE*, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, *Member, IEEE*, Jonathan Li, *Fellow, IEEE*

Abstract—Neural Radiance Field (NeRF) has recently become a significant development in the field of Computer Vision, allowing for implicit, neural network-based scene representation and novel view synthesis. NeRF models have found diverse applications in robotics, urban mapping, autonomous navigation, virtual reality/augmented reality, and more. Due to the growing popularity of NeRF and its expanding research area, we present a comprehensive survey of NeRF papers from the past two years. Our survey is organized into architecture and application-based taxonomies and provides an introduction to the theory of NeRF and its training via differentiable volume rendering. We also present a benchmark comparison of the performance and speed of key NeRF models. By creating this survey, we hope to introduce new researchers to NeRF, provide a helpful reference for influential works in this field, as well as motivate future research directions with our discussion section.

Index Terms—Neural Radiance Field, NeRF, Computer Vision Survey, Novel View Synthesis, Neural Rendering, Volume Rendering, 3D Reconstruction

1 Introduction

Neural Radiance Field (NeRF) models are novel view synthesis methods which use volume rendering with (typically) implicit neural scene representation via Multi Layer Perceptrons (MLPs) to learn the geometry and lighting of a 3D scene. Mildenhall et al. first introduced NeRF at ECCV 2020 [1], and since then, it has achieved state-of-the-art visual quality, produced impressive demonstrations, and inspired many subsequent works. Recently, NeRF models have found applications in photo-editing, 3D surface extraction, human avatar modelling, and large/city-scale 3D representation and view synthesis.

NeRF models have important advantages over other methods of novel view synthesis and scene representation.

- NeRF models are self supervised. They can be trained using only multi-view images of a scene. Unlike many other neural representations of 3D scenes, NeRF models require only images and poses to learn a scene, and do not require 3D/depth supervision. The poses can also be estimated using Structure from Motion (SfM) packages such as COLMAP [2], as was done in certain scenes in the original NeRF paper.
- NeRF models are photo-realistic. Compared to classical techniques such as [3] [4], as well as earlier novel view synthesis methods such as [5][6][7], neural 3D representation methods [8][9][10], the original NeRF model converged to better results in terms of visual quality, with more recent models performing even better.

NeRF models have attracted much attention in the computer vision community in the recent past, with hundreds of papers and preprints appearing on popular code aggregation website [1] with many eventually appearing in top-tier computer vision conference. In 2022, the impact of NeRF is large and ever increasing, with the original NeRF paper by Mildenhall et al. receiving more than 2000 citations (as of May 2023), and growing interest year-over-year. Given current interest, we believe it necessary to organize a survey paper to help computer vision practitioners with this new topic. We also introduce some of the more recent literature missed out by previous surveys.

The rest of this manuscript is organized as follows.

- Section 2 introduces existing NeRF surveys preprints (2.1), explains the theory behind NeRF volume rendering (2.2), introduces the commonly used datasets (2.3) and quality assessment metrics (2.4).
- Section 3 forms the main body of the paper, and introduces the influential NeRF publications, and contains the taxonomy we created to organize these works. Its subsections detail the different families of NeRF innovations proposed in the past two years, as well as recent applications of NeRF models to various computer vision tasks.
- Sections 5 and 6 discuss potential future research directions and applications, and summarize the survey.

- *Corresponding authors: Jonathan Li, Linlin Xu.*
- *This research was partially funded by the Natural Science and Engineering Research Council of Canada under Grant RGPIN-2022-03741 and the Mitacs Accelerate Program under Project IT32340.*
- *Kyle Gao, Dening Lu, Linlin Xu, and Jonathan Li are with the Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontrao N2L 3G1, Canada (e-mail: y56gao, d62lu, l44xu, junli@uwaterloo.ca).*
- *Hongjie He and Jonathan Li are with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (e-mail: h69he@uwaterloo.ca , junli@uwaterloo.ca).*
- *Yina Gao is with the Faculty of Engineering, University of Toronto (e-mail: Yina.gao@mail.utoronto.ca).*

1. https://paperswithcode.com/method/nerf

[Gao et al., 2023]

Neural Radiance Fields

- NeRF has attracted extensive attention in both academy and industry.

Instant NeRFs [NVIDIA, 2022]

Immersive Map [Google, 2022]

Challenges

- As a new 3D representation, NeRF faces the following challenges:
  - Efficiency: Faster training and inference
  - Scalability: From object-scale to city-scale
  - Robustness: Robust to input images and camera poses
  - Generalization: Scene agnostic without training on the test data
  - Dynamics: Modeling of dynamic objects
  - Editability: Editable as traditional representations
  - Application: Applied to facilitate other 3D tasks
  - …

Challenges

- As a new 3D representation, NeRF faces the following challenges:
  - Efficiency: Faster training and inference
  - Scalability: From object-scale to city-scale
  - Robustness: Robust to input images and camera poses
  - Generalization: Scene agnostic without training on the test data
  - Dynamics: Modeling of dynamic objects
  - **Editability: Editable as traditional representations**
  - **Application: Applied to facilitate other 3D tasks**
  - …

Outline

3D Editing with NeRF
[ICCV'21, ECCV'22, CVPR'23]

Online Perception with NeRF
[CVPR'22, ICCV'23, Arxiv'23]

3D Editing with NeRF

[ICCV'21, ECCV'22, CVPR'23]

Q1.1

Can we edit the objects in the scene?

Object-Compositional Neural Radiance Field

Input Images

Direct

Standard Novel View Synthesis

Ours

Decompose to object level

Editable Scene Rendering

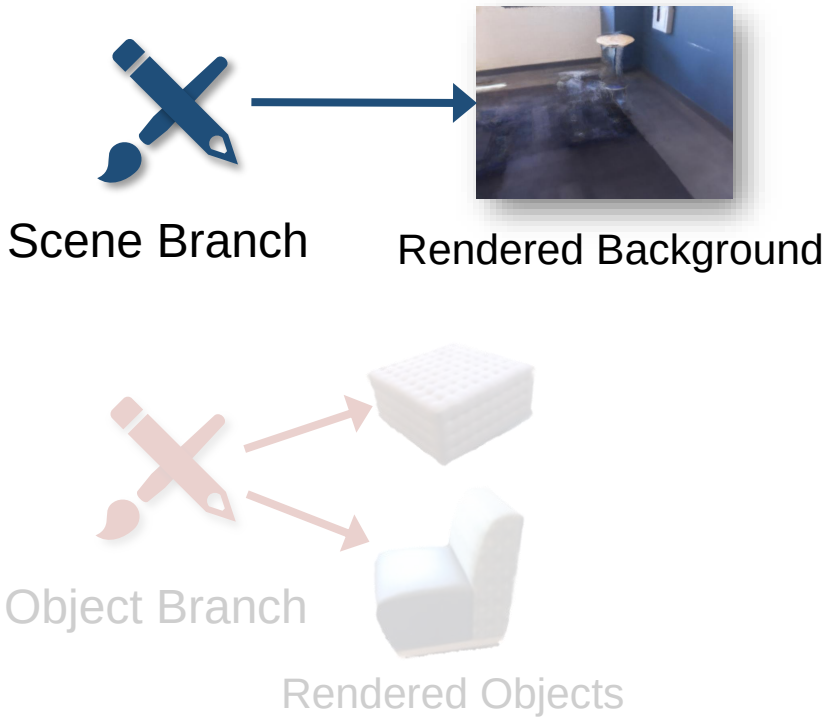
Object-Compositional Neural Radiance Field

We design a two-pathway architecture.

Learning Object-Compositional Neural Radiance Field for Editable Scene Rendering, Yang et al., ICCV 2021

Object-Compositional Neural Radiance Field

Scene Branch: renders the entire view of the scene.

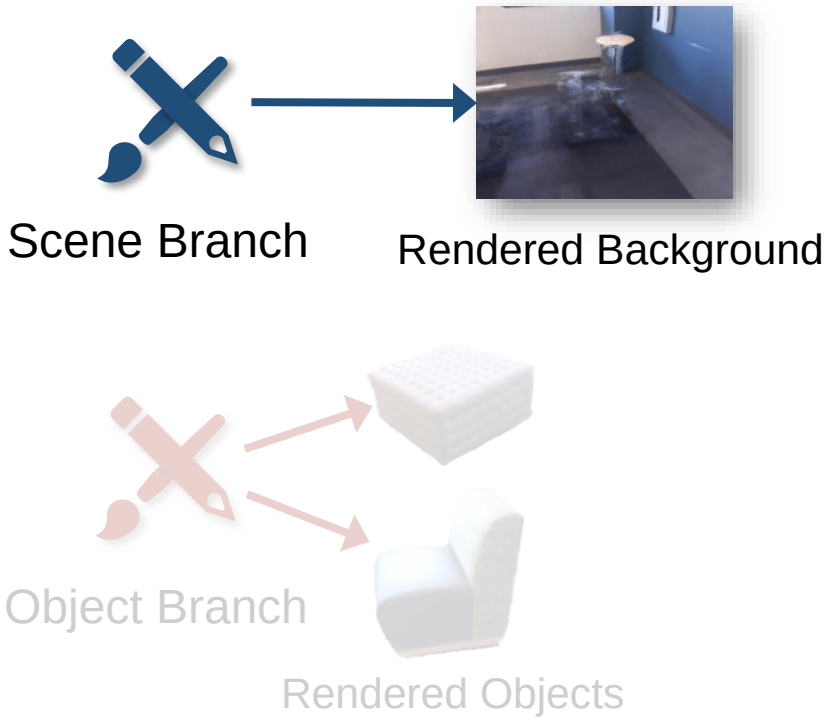


Scene Branch

Full Scene Rendering

Object-Compositional Neural Radiance Field

Scene Branch: renders the background for editable scene rendering.



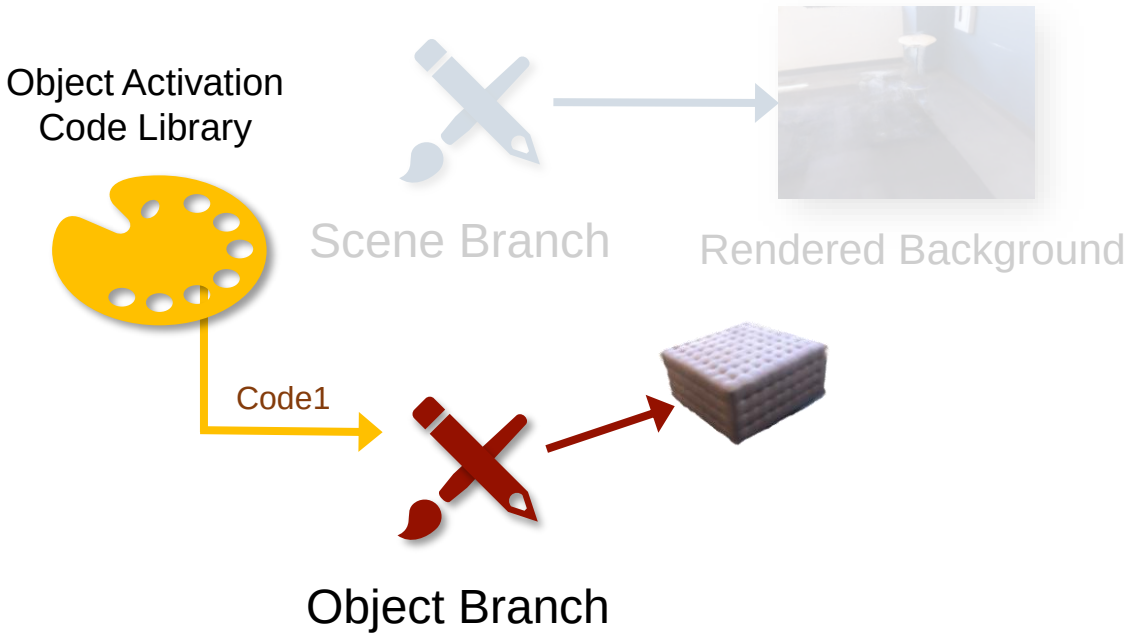
Scene Branch

Background Rendering

(pruning ray samples user-defined region)

Object-Compositional Neural Radiance Field

Object branch: renders each standalone object conditioned on the object activation code.

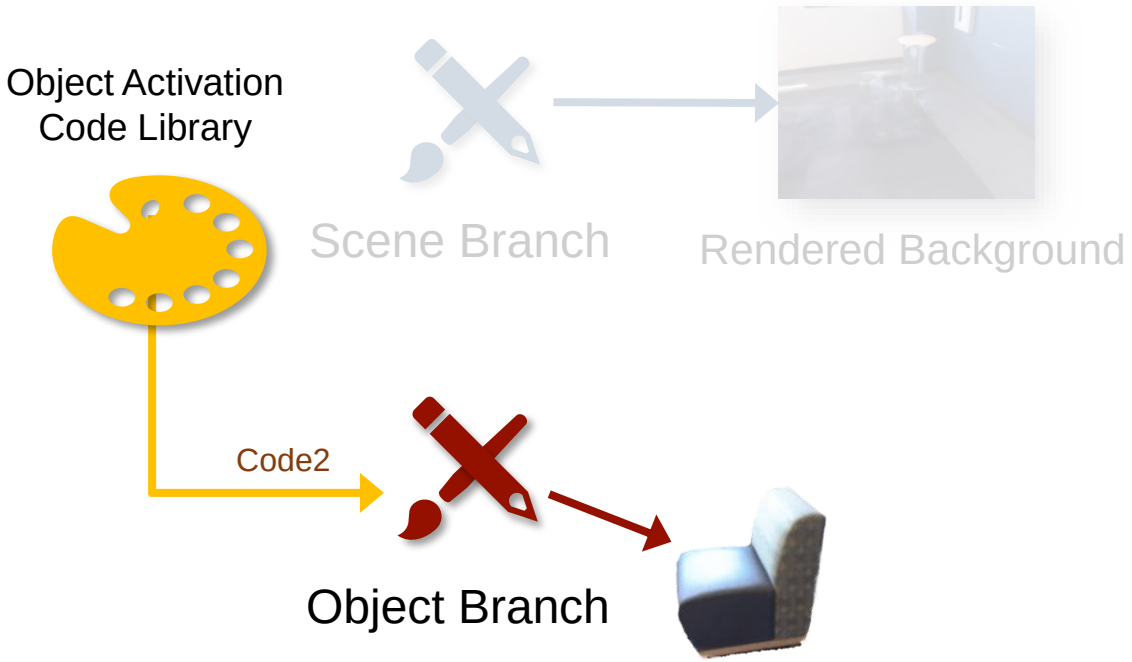


Object Branch

Conditioned on Code1

Object-Compositional Neural Radiance Field

Object branch: renders each standalone object conditioned on the object activation code.

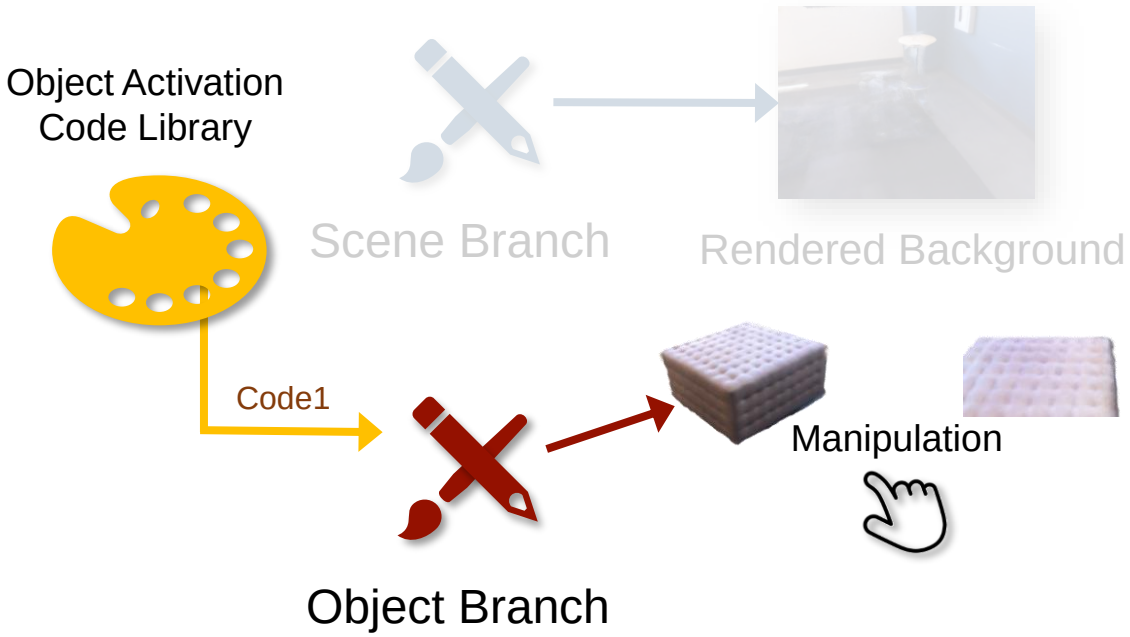


Object Branch

Conditioned on Code2

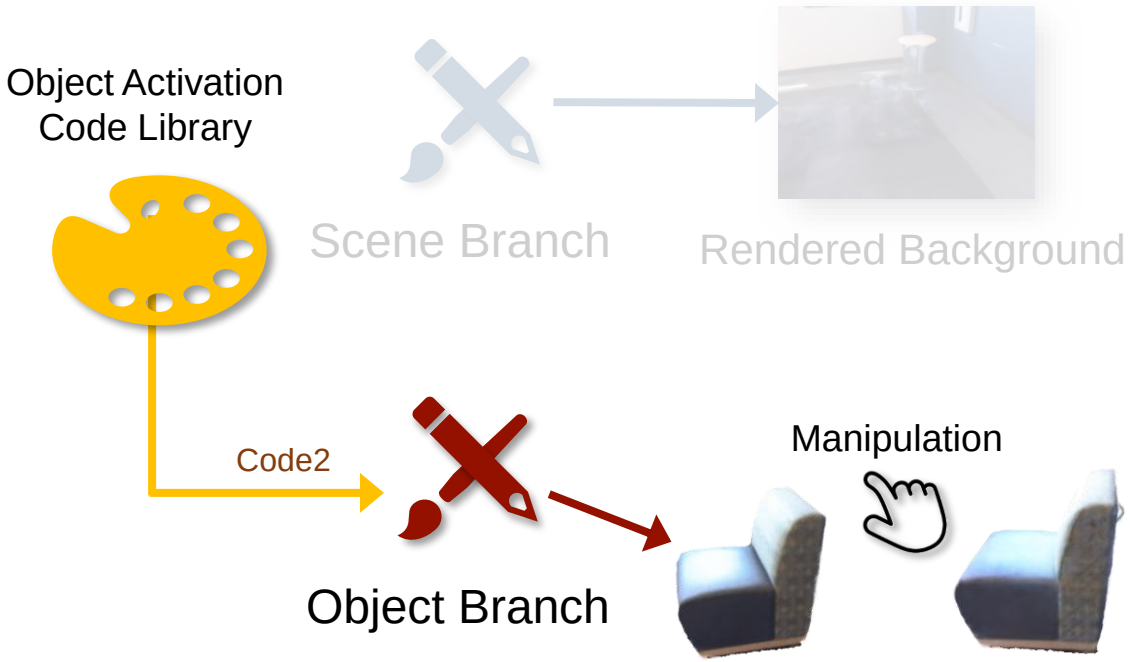
Object-Compositional Neural Radiance Field

We can render the manipulated objects by transforming the shooting rays.



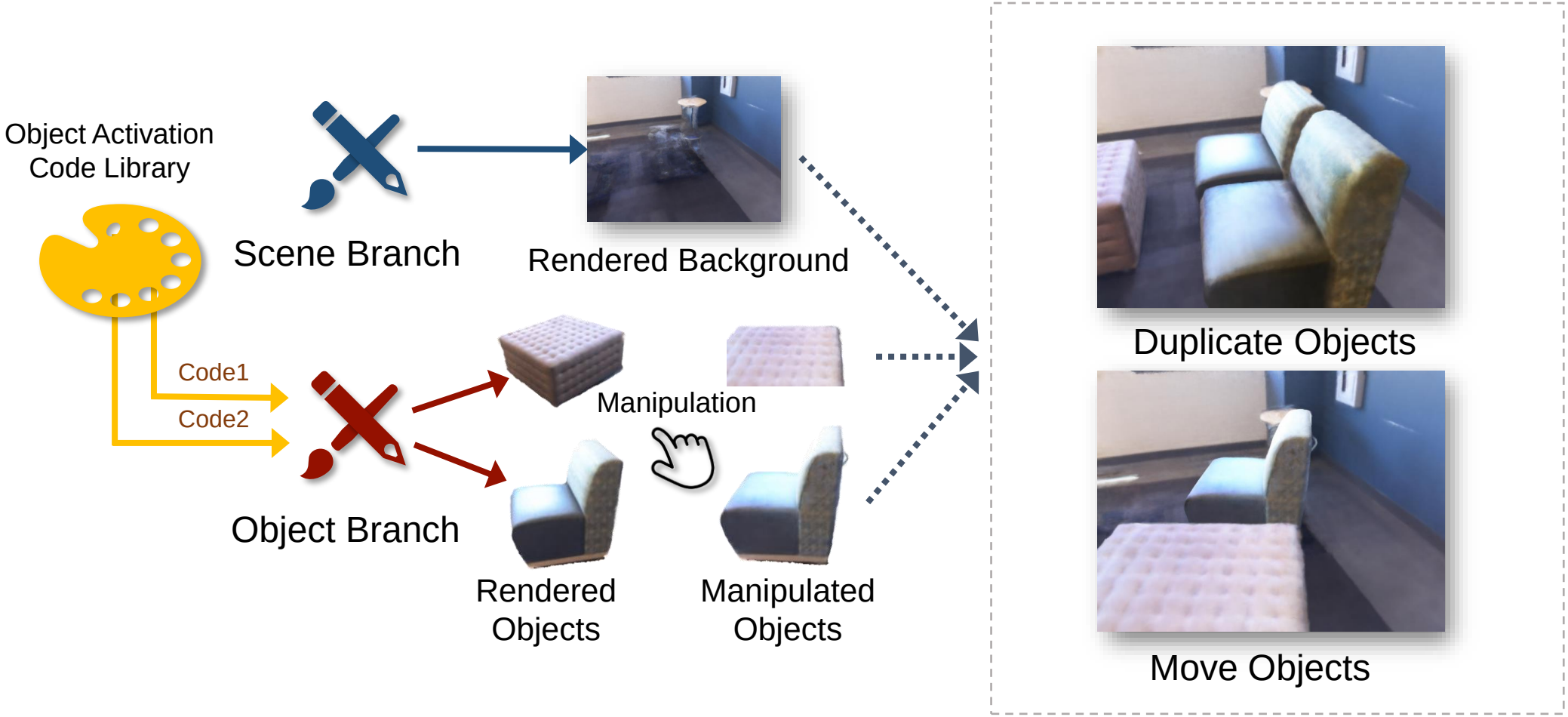
Object-Compositional Neural Radiance Field

We can render the manipulated objects by transforming the shooting rays.

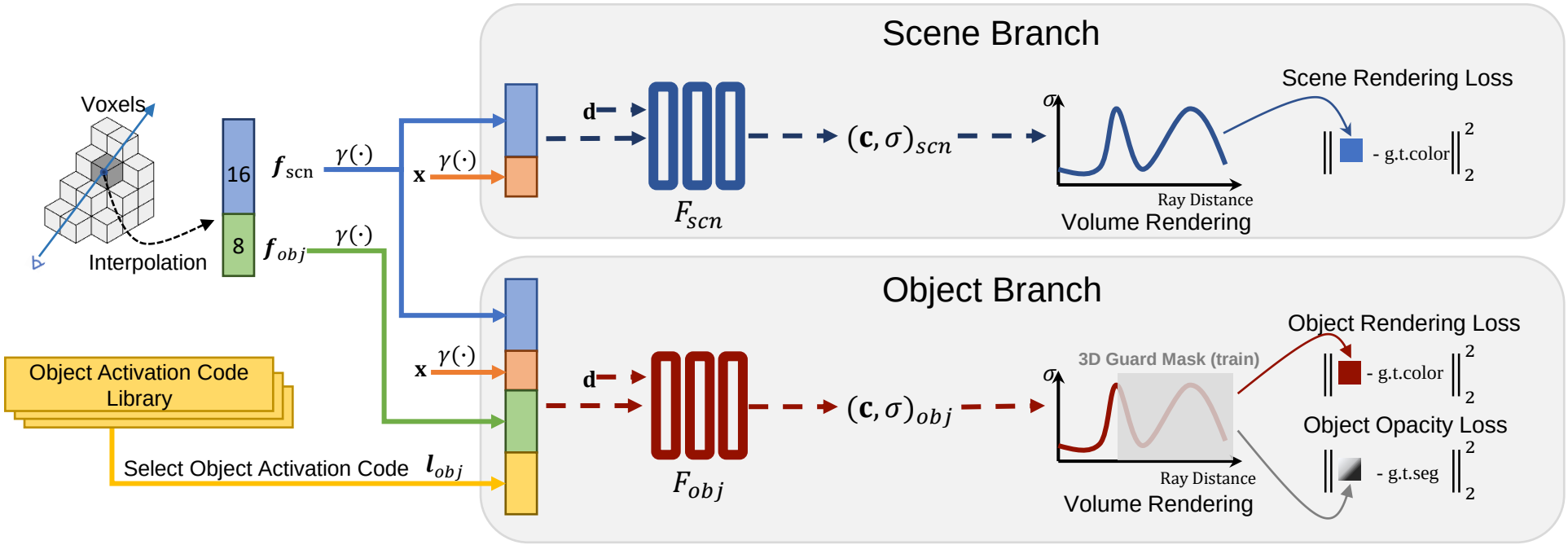


Editable Scene Rendering

We jointly render the objects and the background.



Framework Details



Object-level Supervision

- Objective Rendering

$$\hat{C}(\boldsymbol{r})_{obj} = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_{obj\,i}$$

Object Color

$$\hat{O}(\boldsymbol{r})_{obj} = \sum_{i=1}^{N} T_i \alpha_i$$

Object Opacity

$$\alpha_i = 1 - \exp\left(-\sigma_{obj\,i}\delta_i\right)$$

Sum of the product of transmittance and alpha of points along the ray.

Object-level Supervision

- Objective Rendering

$$\hat{C}(\boldsymbol{r})_{obj} = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_{obj\,i}$$

Object Color

$$\hat{O}(\boldsymbol{r})_{obj} = \sum_{i=1}^{N} T_i \alpha_i$$

Object Opacity

$$\alpha_i = 1 - \exp\left(-\sigma_{obj\,i}\delta_i\right)$$

Sum of the product of transmittance and alpha of points along the ray.

- Loss Definition

$M(\boldsymbol{r})^k$ Rough Segmentation Mask

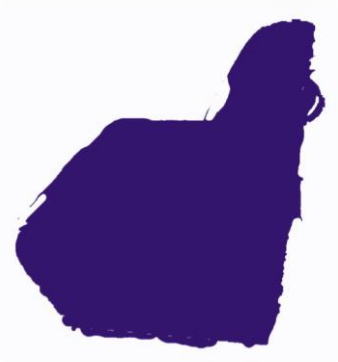
$$\mathcal{L}_{obj} = \sum_{\boldsymbol{r}\in N_r} \sum_{k\in[\![1..K]\!]} \lambda_1 M(\boldsymbol{r})^k ||\hat{C}(\boldsymbol{r})^k_{obj} - C(\boldsymbol{r})||_2^2 + \lambda_2 w(\boldsymbol{r})^k ||\hat{O}(\boldsymbol{r})^k_{obj} - M(\boldsymbol{r})^k||_2^2$$

Rendered objects should be close to the observations.

Only opaque at the object area and transparent elsewhere.

Object-level Supervision

Image View



Input Segmentation

Rendered Object

Rendered Opacity
(Segmentation)

Image View

Annotated Segmentation

Rendered Object

Rendered Opacity
(Segmentation)

😄 Robust to segmentation noise

😄 Even works for complex shapes

Handling Occlusion in Clustered Scenes

Clustered Real-world Scene

Directly apply

object supervision

Incomplete Reconstruction

Handling Occlusion in Clustered Scenes

Clustered Real-world Scene

Directly apply

object supervision

Incomplete Reconstruction

2D instance masks is facing the 3D space ambiguity in the occluded region.

Handling Occlusion in Clustered Scenes

Clustered Real-world Scene

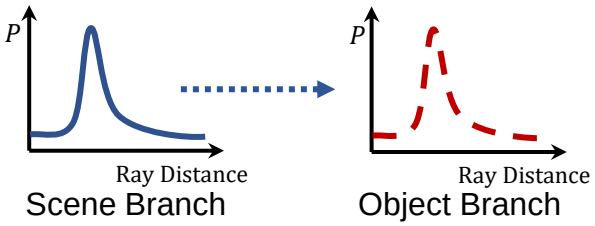
Directly apply

object supervision

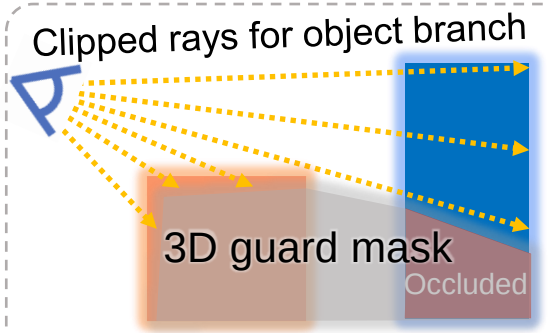
Incomplete Reconstruction

2D instance masks is facing the 3D space ambiguity in the occluded region.

Our Solution



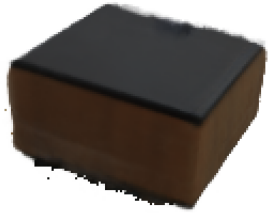
Sampling Distribution Guidance



Block Gradient to the Occluded Space

Biased and masked

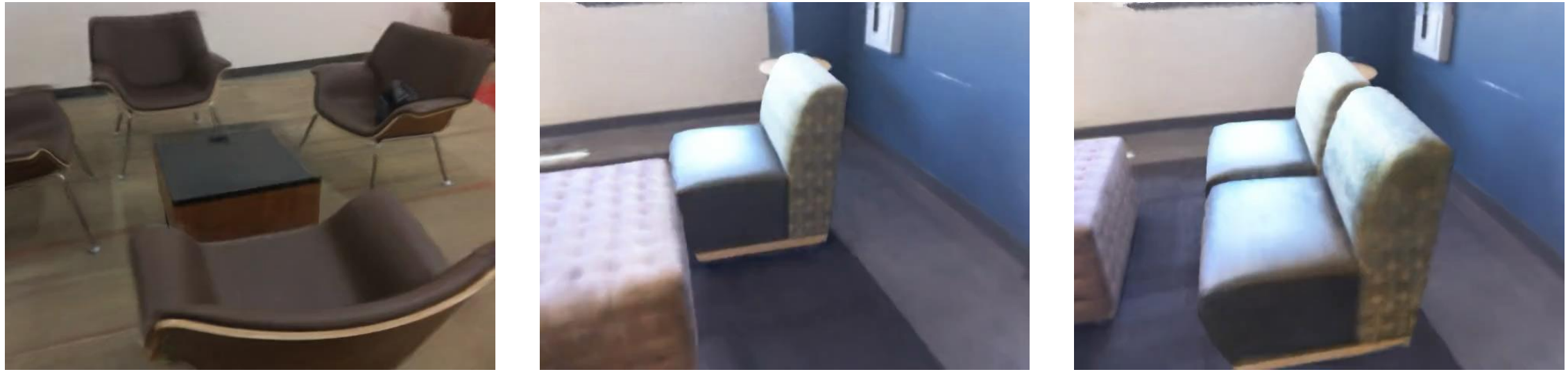
supervision



Complete Reconstruction

Examples on the ScanNet Dataset

Novel View Synthesis



Editable Scene Rendering

Examples on the ToyDesk Dataset

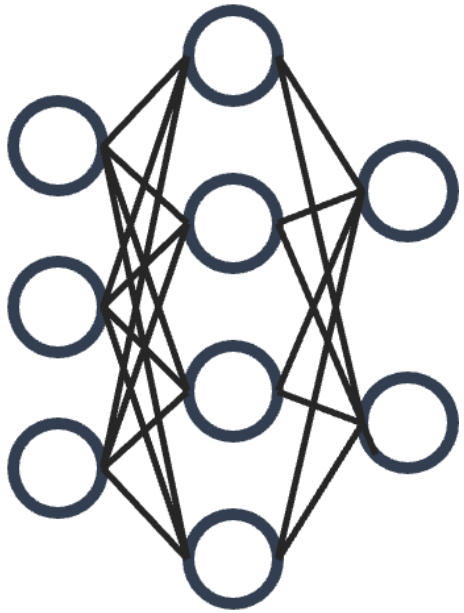
Novel View Synthesis

Editable Scene Rendering

Q1.2

Can we edit both the geometry and texture of neural radiance fields?

NeuMesh: Geometry&Texture Disentangled NeRF



Transfer

Pure MLP Model

Mesh-based Representation

❌ Baked → Not Editable

✅ Disentangled → Editable

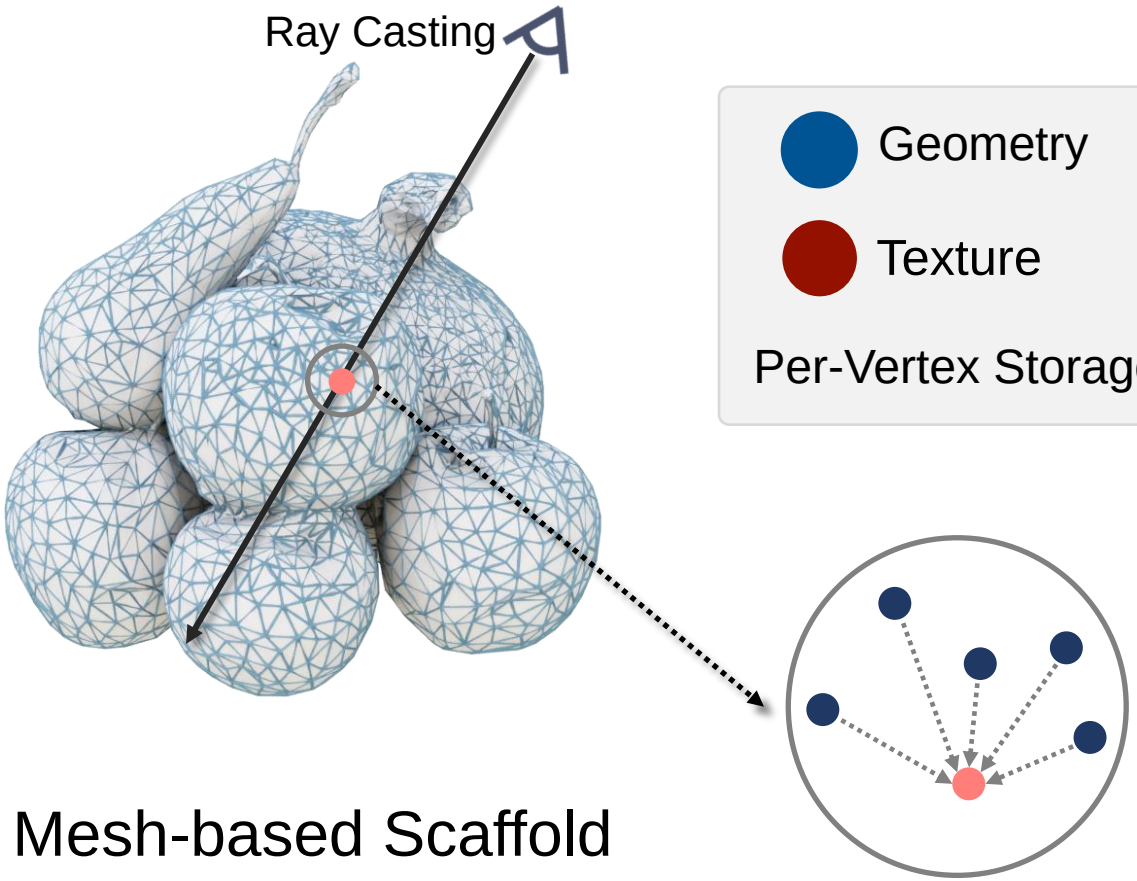
Transfer a pure MLP model into a mesh-based editable representation

NeuMesh: Learning Disentangled Neural Mesh-based Implicit Field for Geometry and Texture Editing, Yang et al., ECCV 2022

NeuMesh: Geometry&Texture Disentangled NeRF

Volume Rendering of the Object

NeuMesh: Geometry&Texture Disentangled NeRF



Mesh-based Scaffold

Geometry

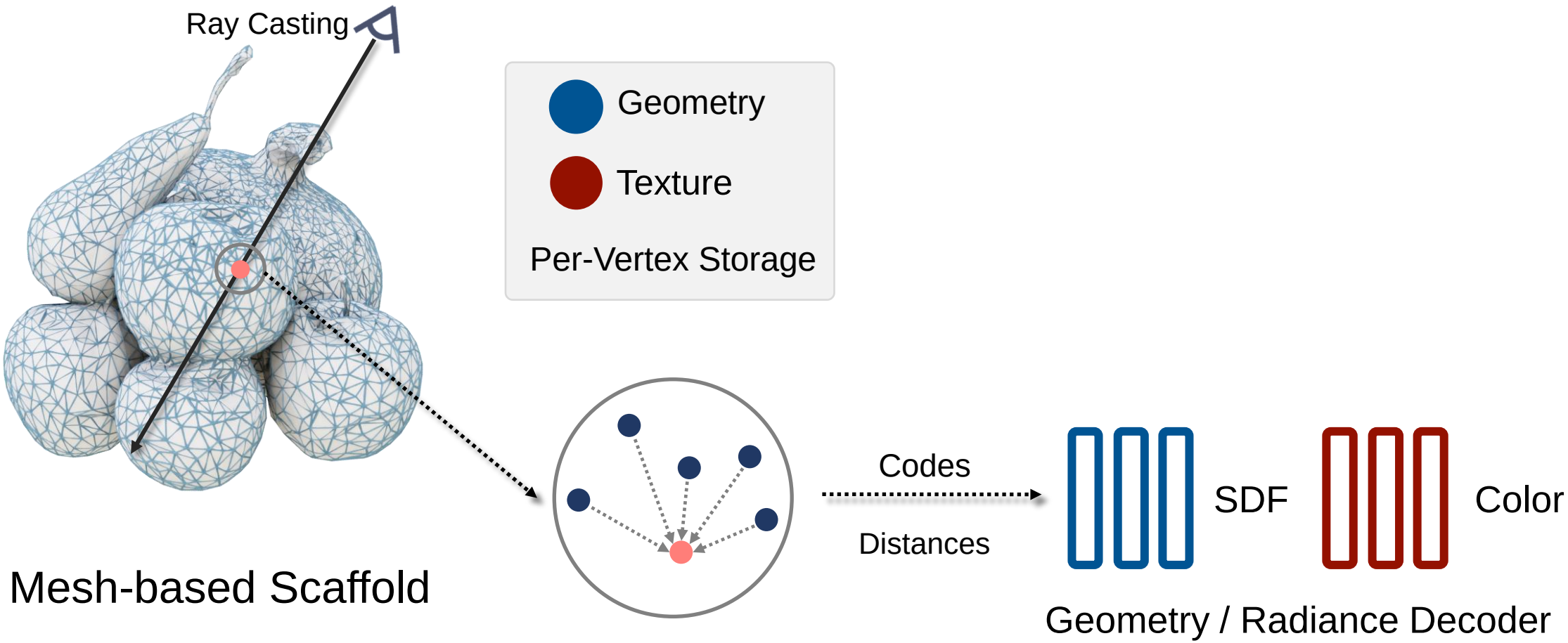
Texture

Per-Vertex Storage

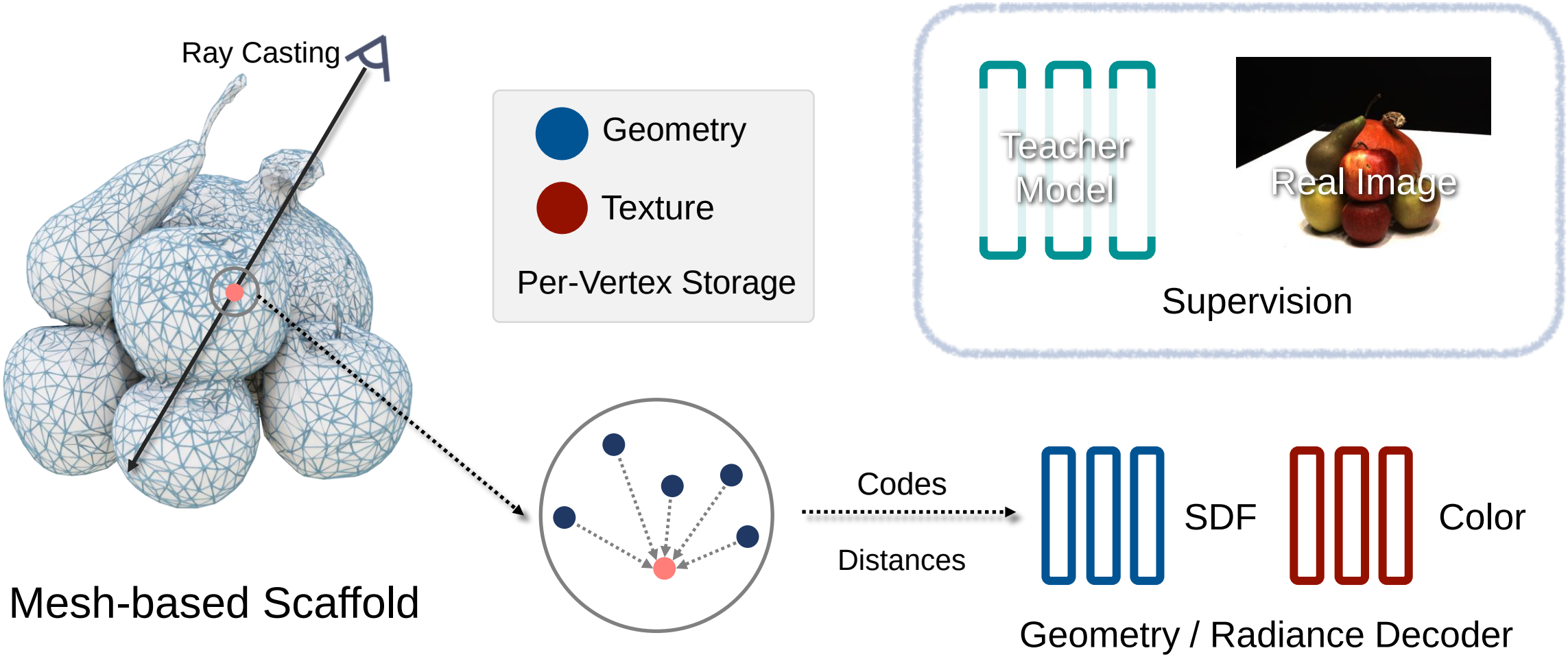
Disentangled Geometry / Texture Codes in Mesh Vertices

Query Points to Obtain Locally Interpolated Codes

NeuMesh: Geometry&Texture Disentangled NeRF



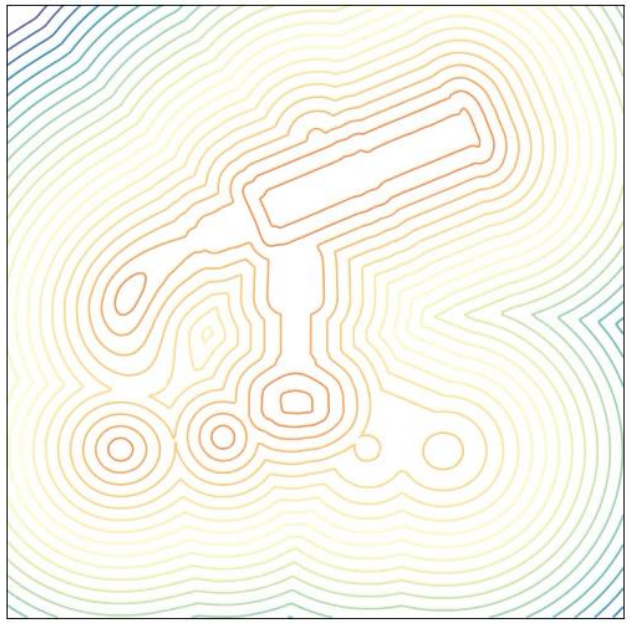
NeuMesh: Geometry&Texture Disentangled NeRF



Now, we will show how we enable editing.

How we enable editing - Geometry Editing

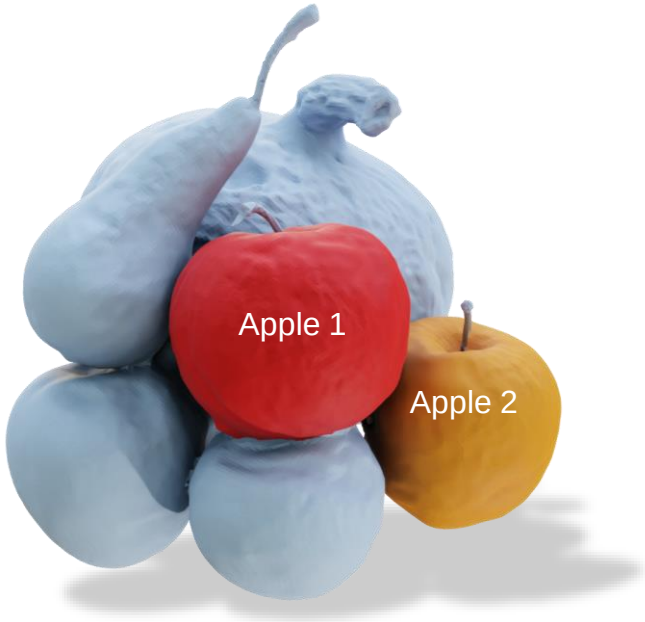
User Geometry Edit



Synchronized Field Deformation

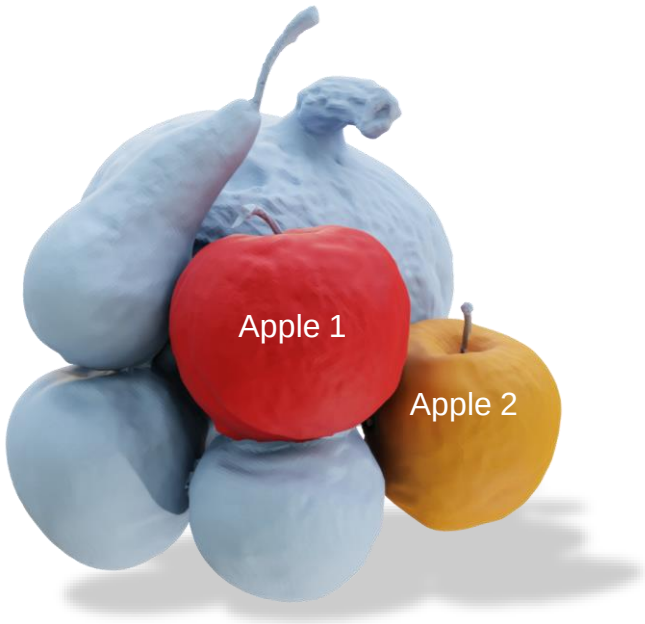
Rendering Result

How we enable editing - Texture Swapping



User-Selected Area

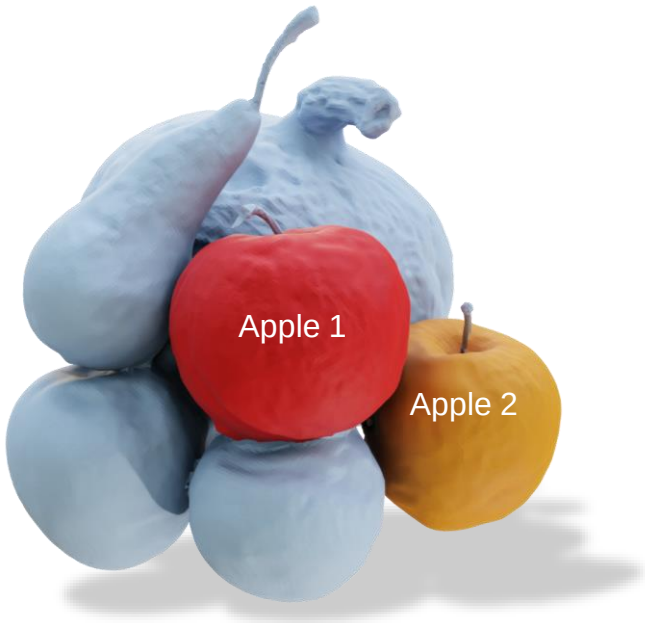
How we enable editing - Texture Swapping



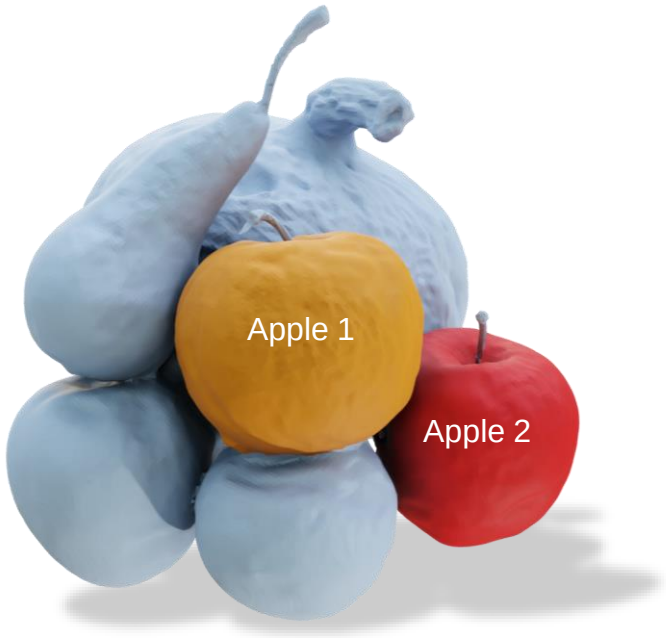
User-Selected Area

Non-rigid Alignment

How we enable editing - Texture Swapping

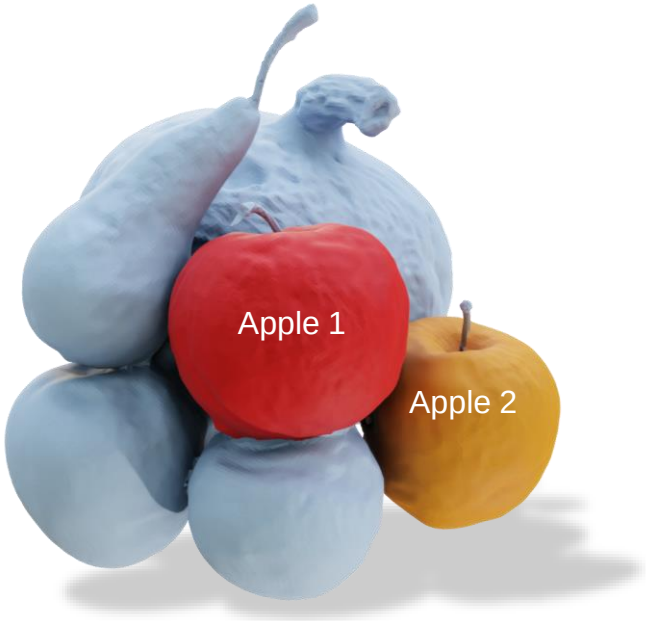


User-Selected Area

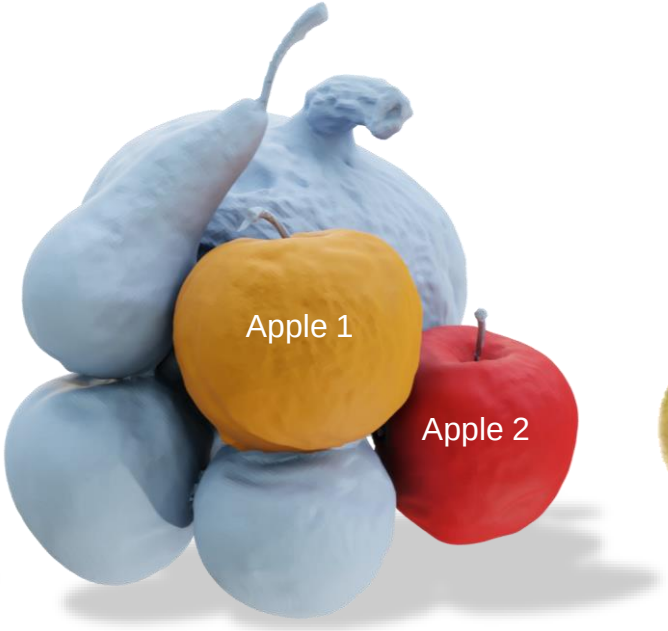


Texture Code Swapped

How we enable editing - Texture Swapping



Area selection



Texture Code Swapping

Before

After

Rendering Results

How we enable editing - Texture Filling

Original Model

User-Selection

User Selected Area

How we enable editing - Texture Filling

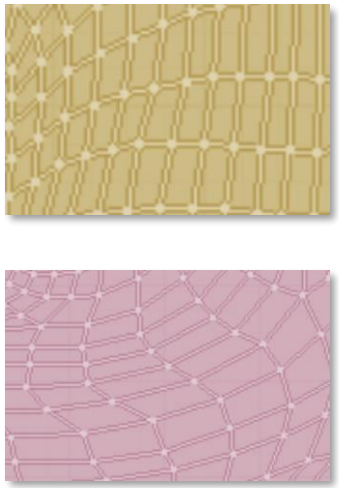
User Selected Area

How we enable editing - Texture Filling

User Selected Area

Unwrap

UV-Map

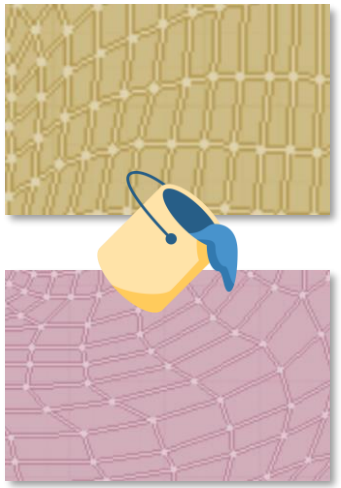


How we enable editing - Texture Filling

User Selected Area

Unwrap

UV-Map



Filling

Latent Texture Templates

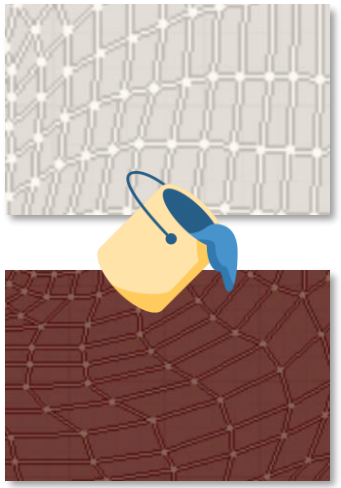
Pre-captured Model

How we enable editing - Texture Filling

User Selected Area

Unwrap

UV-Map



Filling

Latent Texture Templates

Pre-captured Model

How we enable editing - Texture Filling

Original Rendering

Filling

Latent Texture Templates

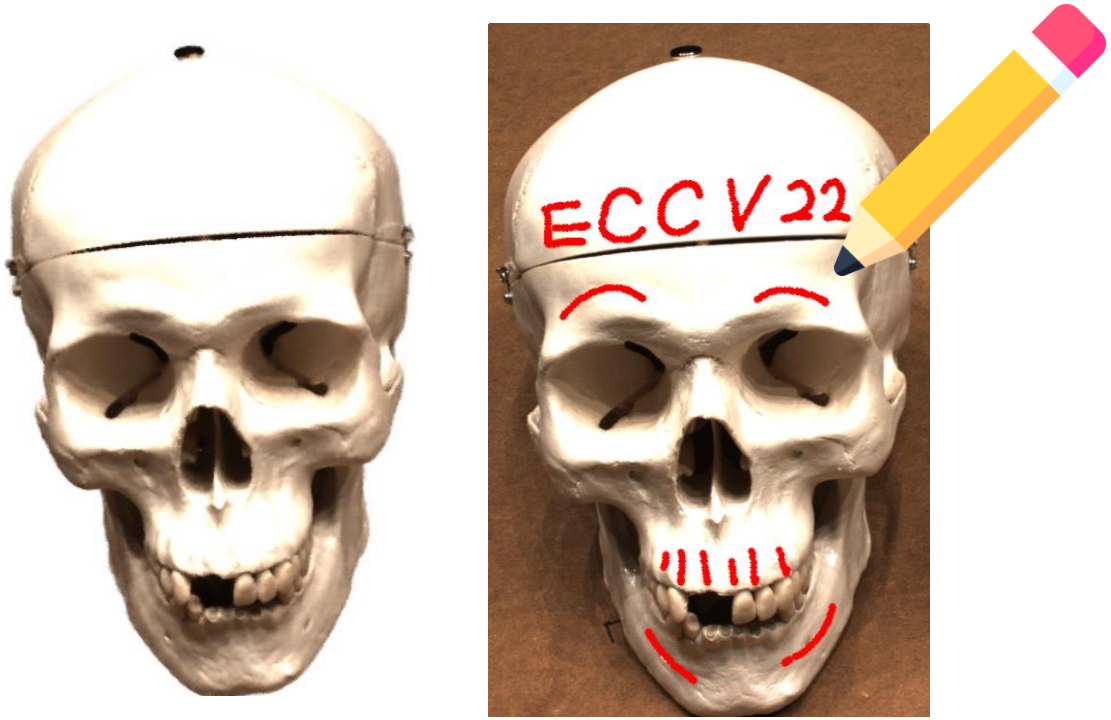
Pre-captured Model

Edited Rendering

How we enable editing - Texture Painting

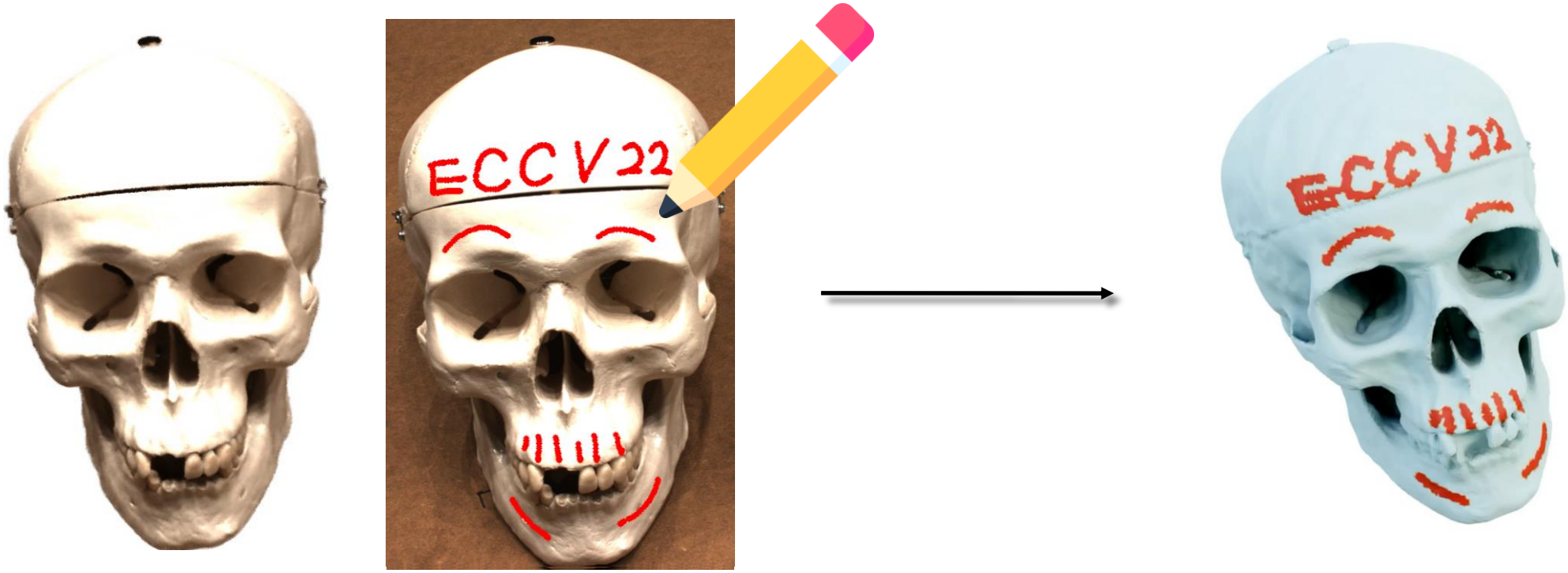
Original Object

How we enable editing - Texture Painting



User Paint on a Single 2D View

How we enable editing - Texture Painting

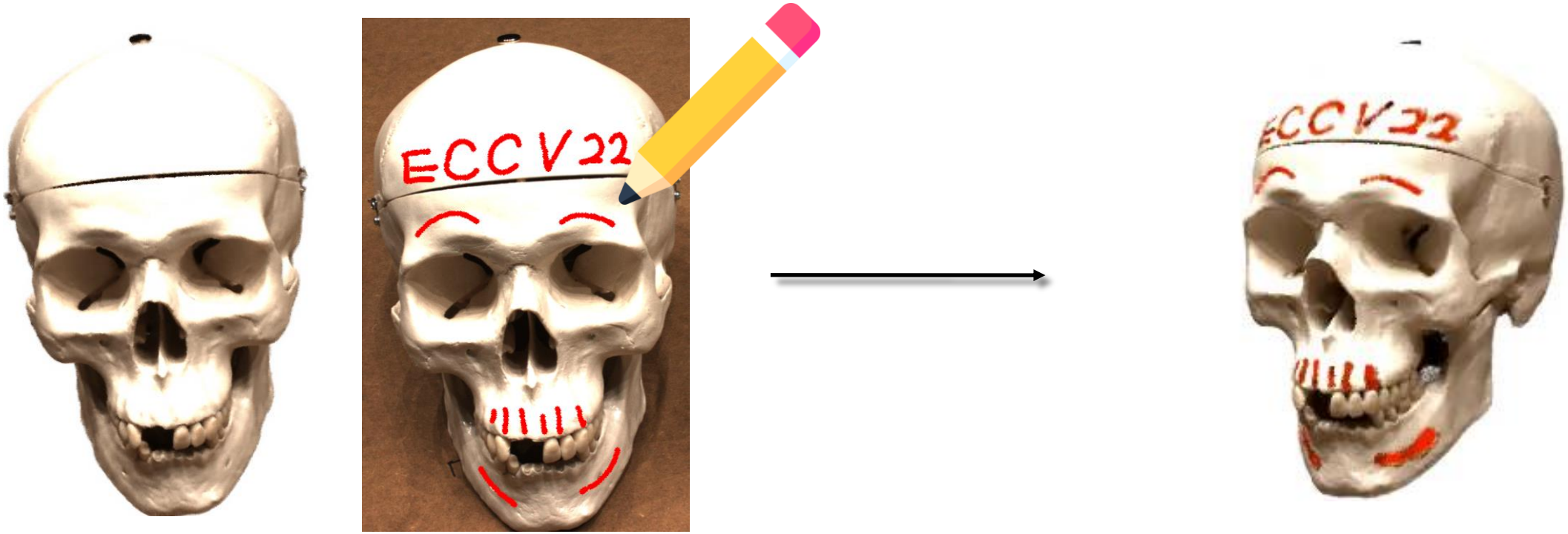


Only Update Affected Codes (highlighted)

User Paint on a Single 2D View

Spatial-Aware Optimization

How we enable editing - Texture Painting



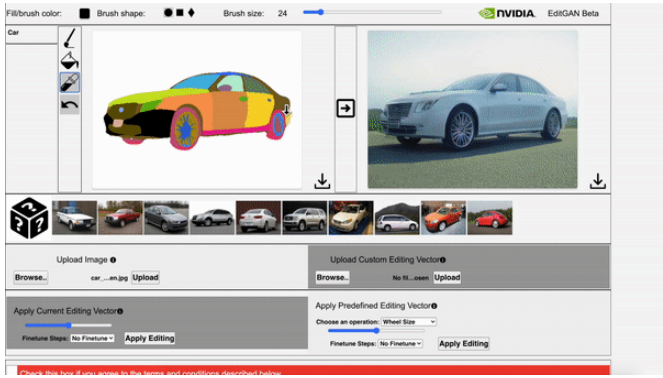
User Paint on a Single 2D View

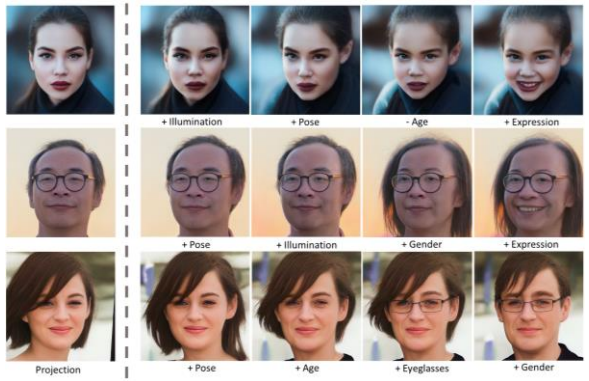
Rendered Object with Transferred Painting

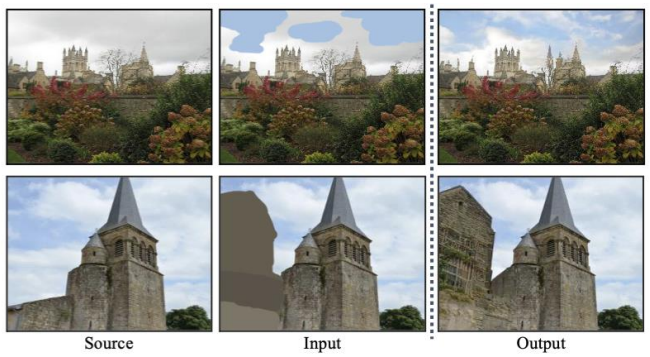
Q1.3

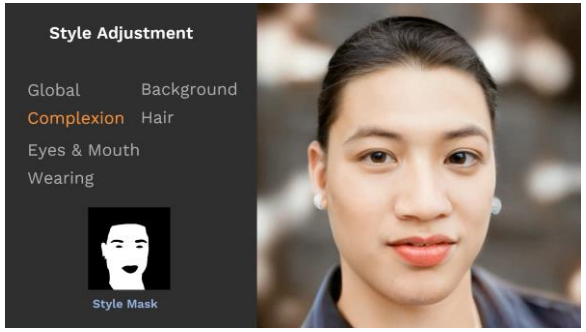
Can we edit neural radiance fields as easy as 2D editing?

Semantic 2D Editing









Text-driven Synthesis & Editing

Stable Diffusion [CVPR' 22]

Text2LIVE [ECCV' 22]

Stroke-based Editing

EditGAN [NeurIPS' 21]

SDEdit [ICLR' 22]

Attribute-based Editing

StyleFlow [ACM ToG' 21]

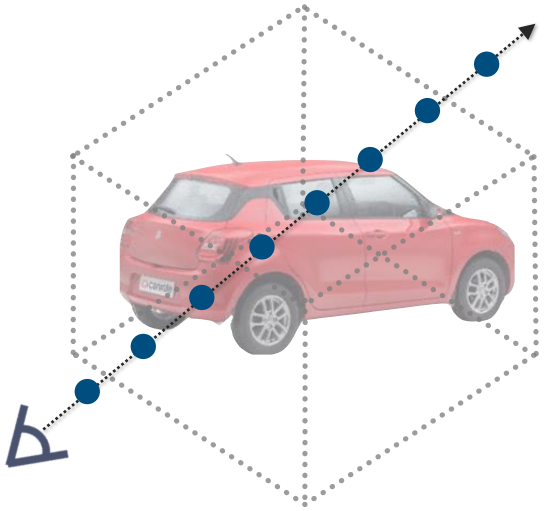
SofGAN [ACM ToG' 22]

We aim at realizing the semantic-driven editing of…

Editing 3D from a single perspective.

Supporting real-world objects and scenes.

SINE: Semantic-driven Image-based NeRF Editing

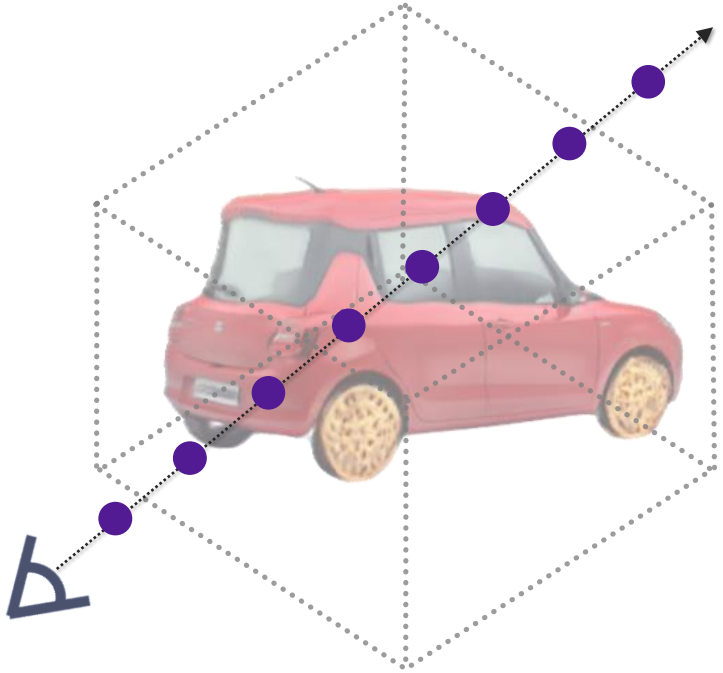


(1) Pre-trained NeRF

(2) Single-View 2D Editing

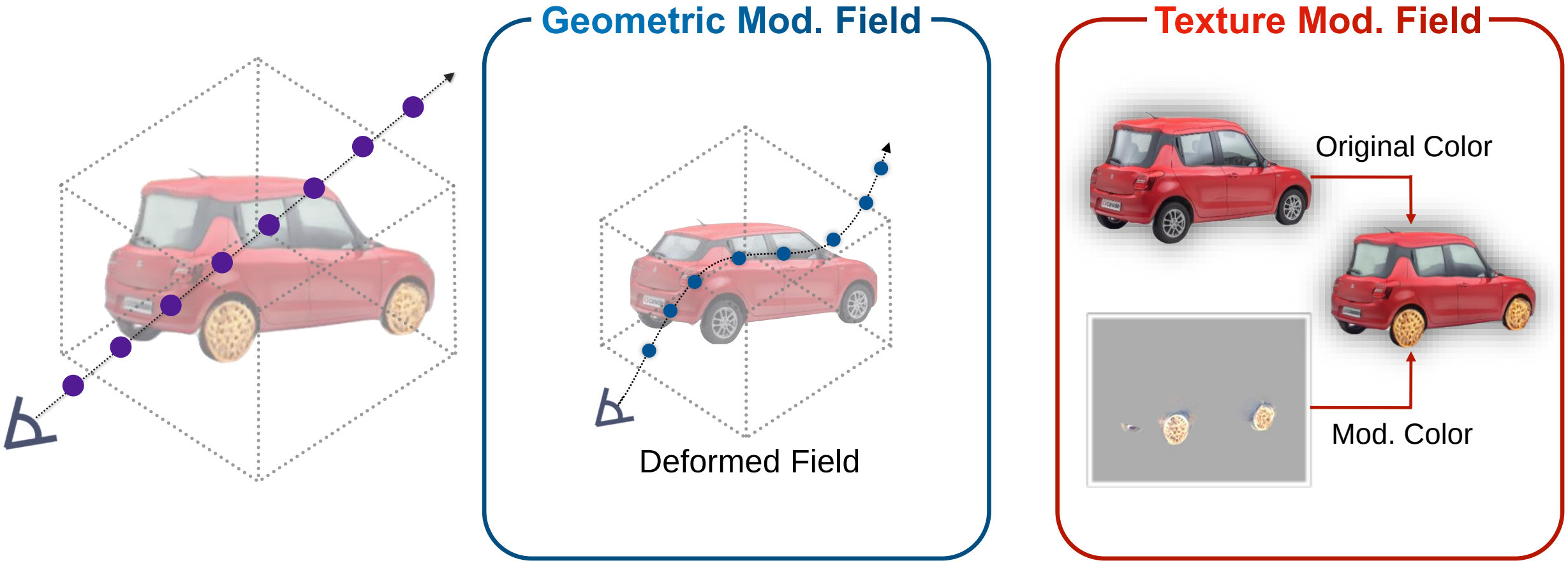
SINE

Semantic-driven Editing



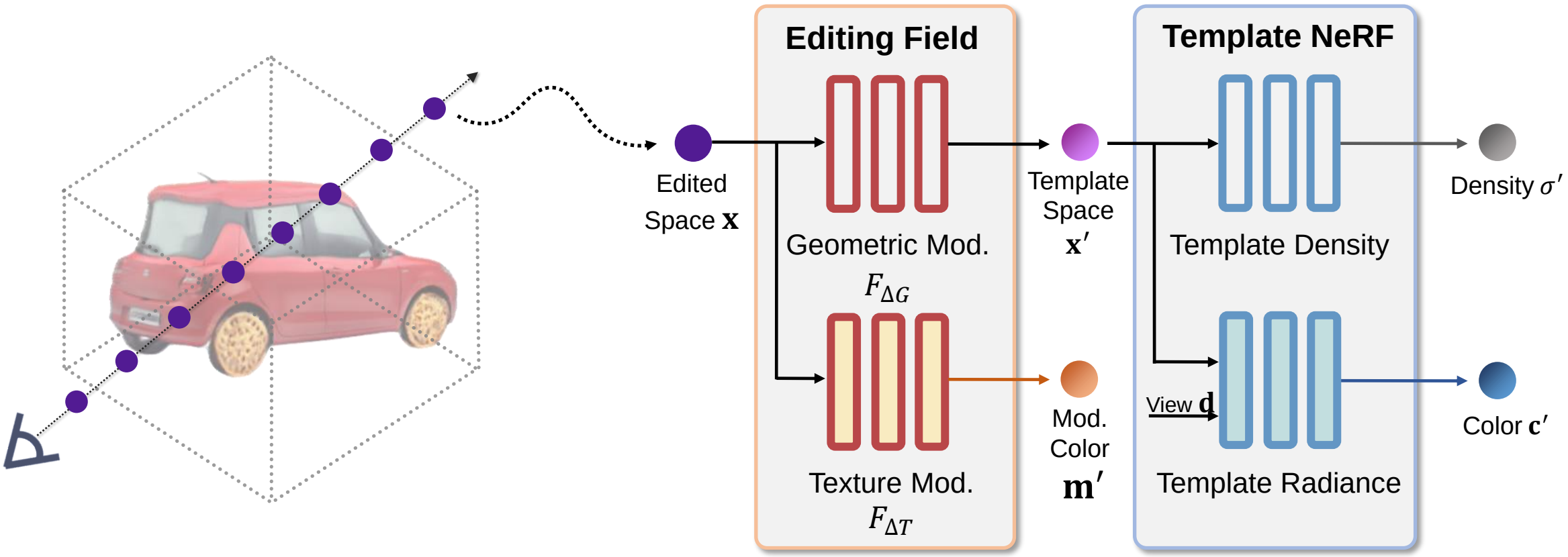
Edited NeRF

SINE: Semantic-driven Image-based NeRF Editing



Edited NeRF

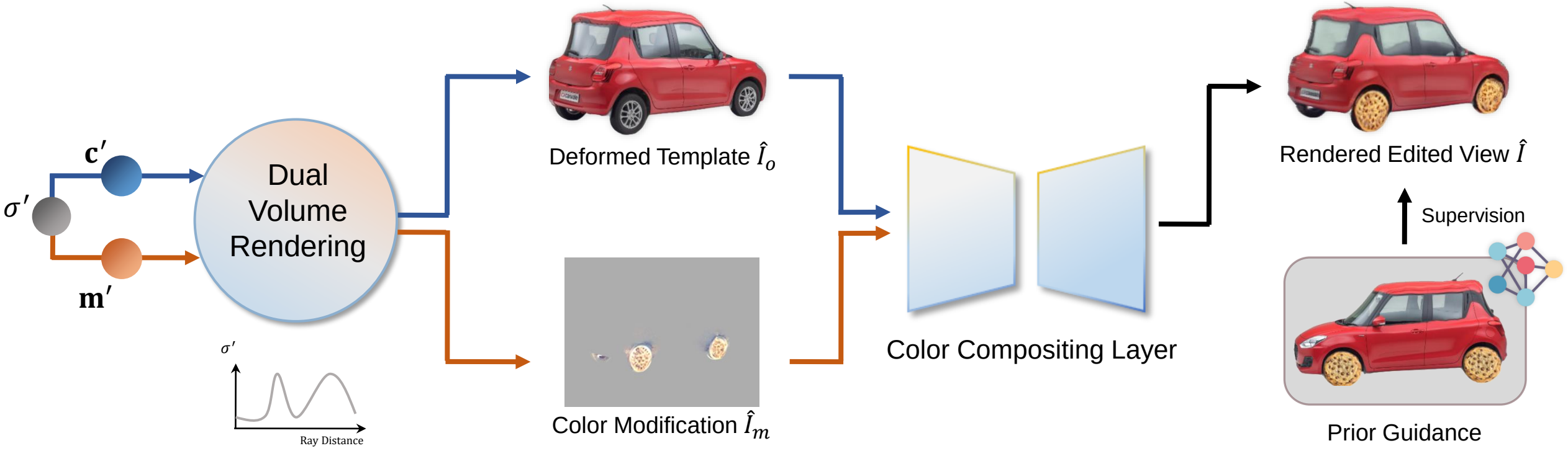
SINE: Semantic-driven Image-based NeRF Editing



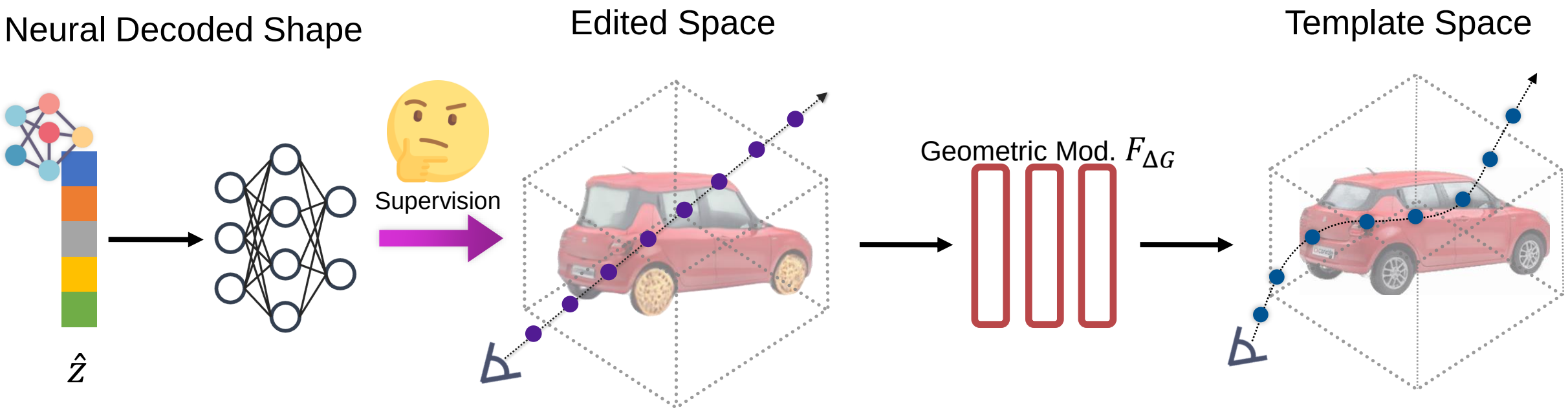
Edited NeRF

Editing on the Template NeRF

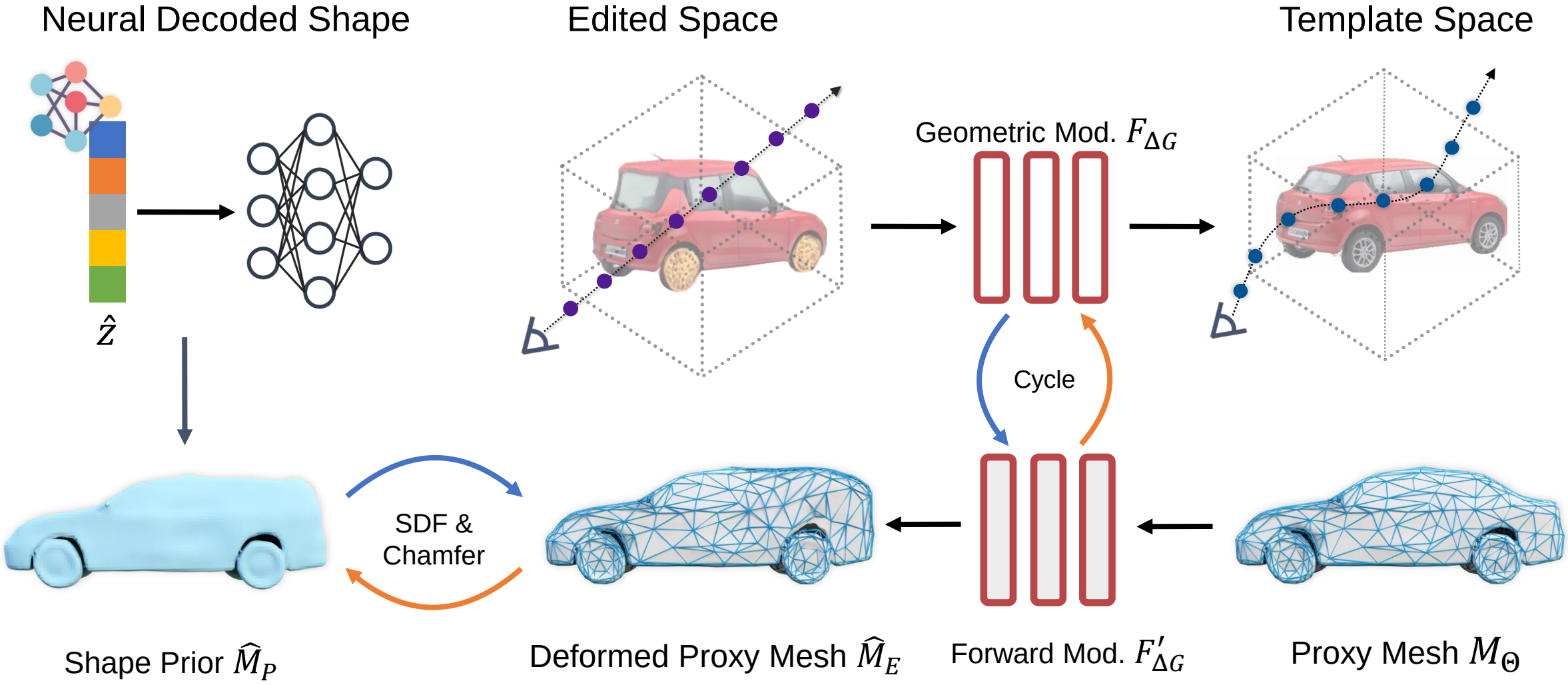
SINE: Semantic-driven Image-based NeRF Editing



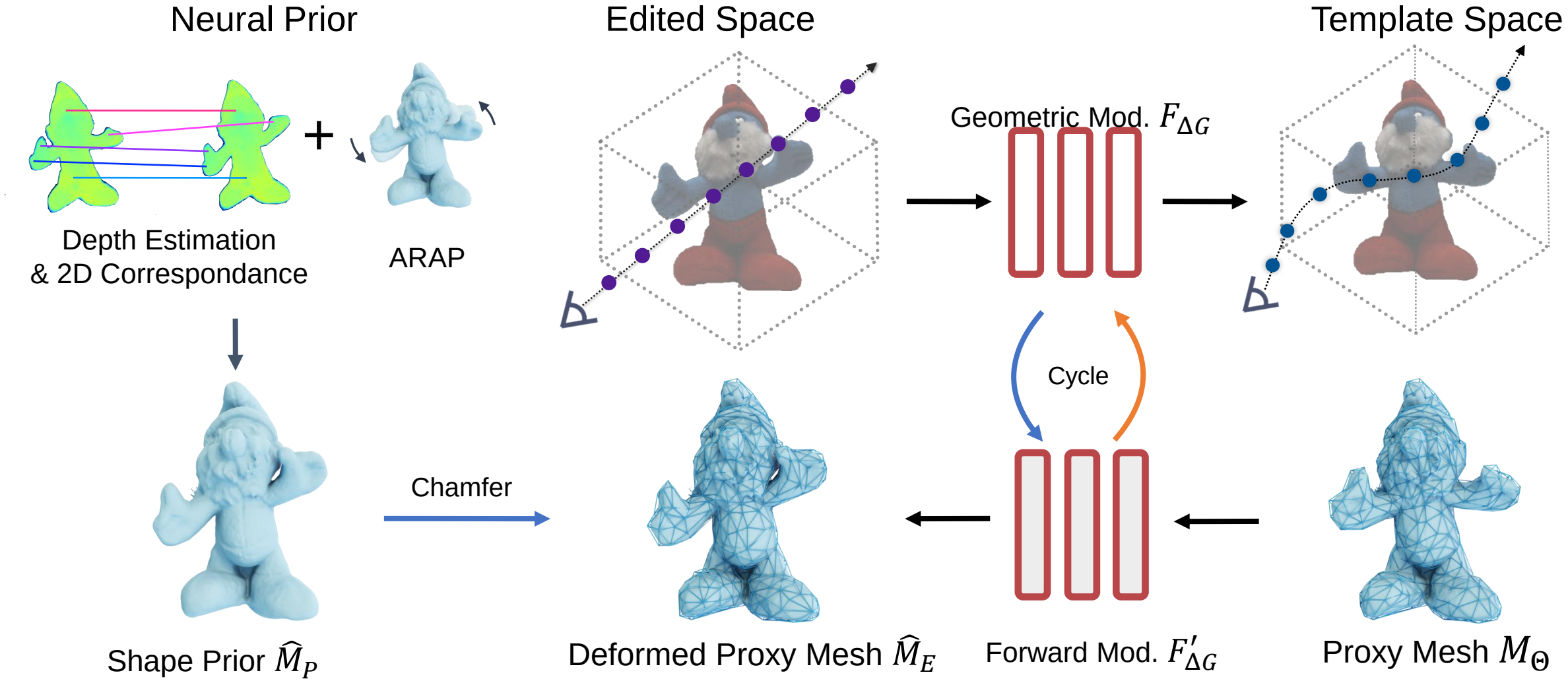
SINE: Semantic-driven Image-based NeRF Editing



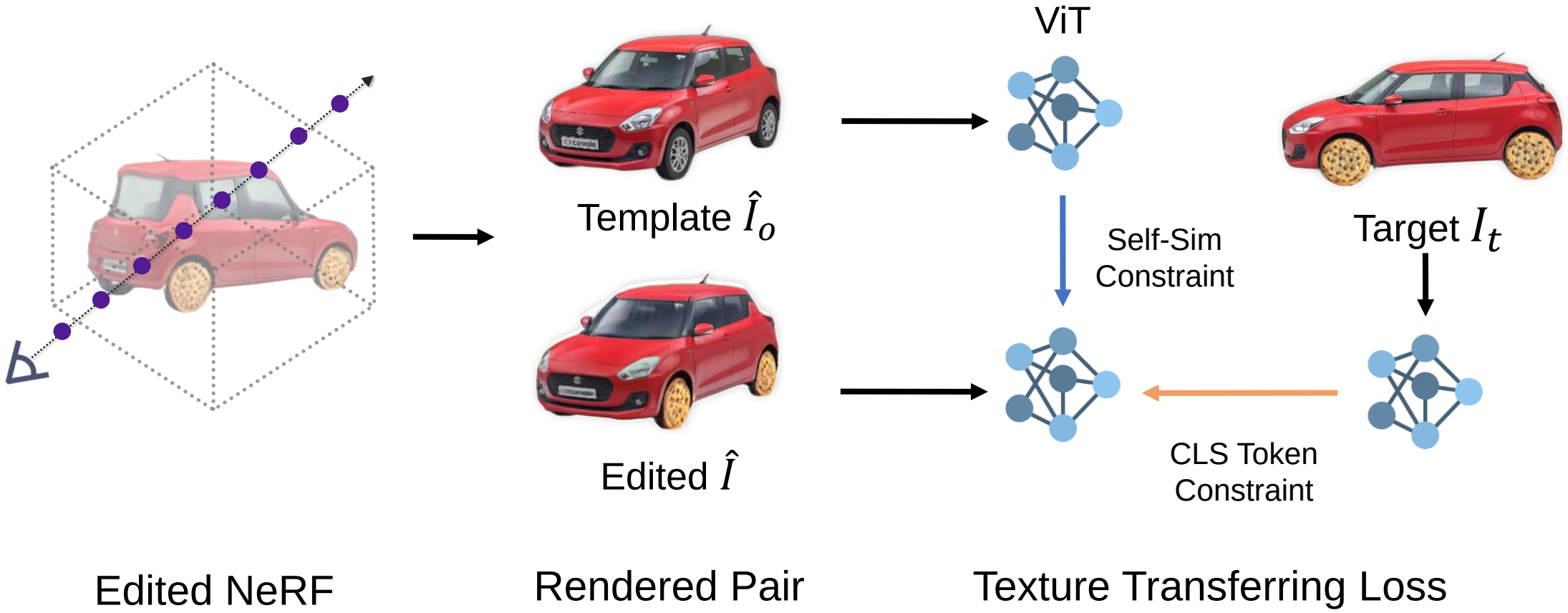
SINE: Semantic-driven Image-based NeRF Editing



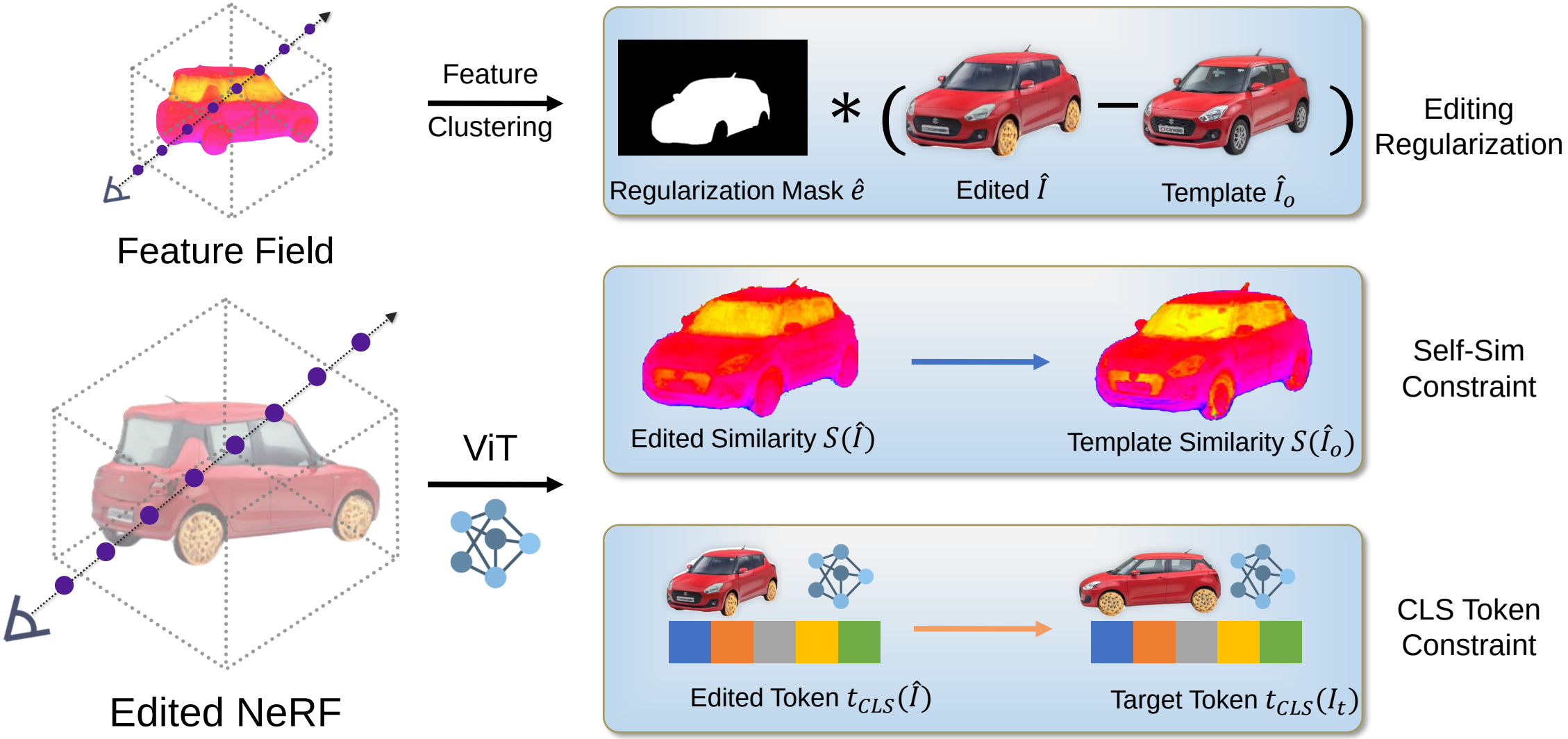
SINE: Semantic-driven Image-based NeRF Editing



SINE: Semantic-driven Image-based NeRF Editing



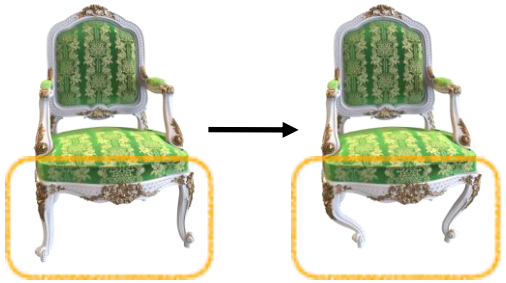
SINE: Semantic-driven Image-based NeRF Editing



Geometric Editing on the Real-world Cars

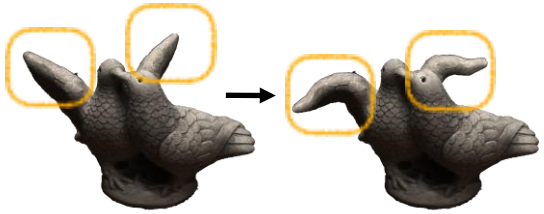
Geometric Editing on the Generic Objects

Source



2D Editing

Source



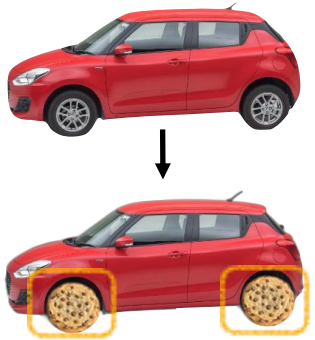
2D Editing

Edited

Edited

Texture Editing with Single-View User Editing

Source



2D Editing

Source

2D Editing

Edited

Edited

Texture Editing with Text-prompts

Source View

“Swarovski Blue Crystal Car”

“Ice Sculpture Car”

Source View

“Golden Pinecone”

“Burning Pinecone”

Using single image editing from Text2LIVE [Bar-Tal, 2022].

Texture Editing with Text-prompts

Source View

“Stained Glass Vasedeck”

“Snowy Vasedeck”

Source View

“Plastic Round Table”

“Kids Plasticine Round Table”

Using single image editing from Text2LIVE [Bar-Tal, 2022].

Online Perception with NeRF

[CVPR'22, ICCV'23, Arxiv'23]

Q2.1

Can we exploit NeRF for online 3D perception?

Challenges

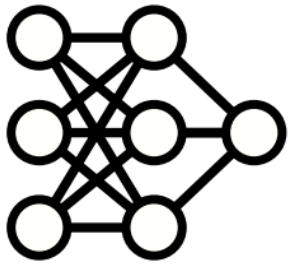
- Efficiency

NeRF [Mildenhall et al., ECCV’20]

F2-NeRF [Wang et al., CVPR’23]

Challenges

- Catastrophic forgetting



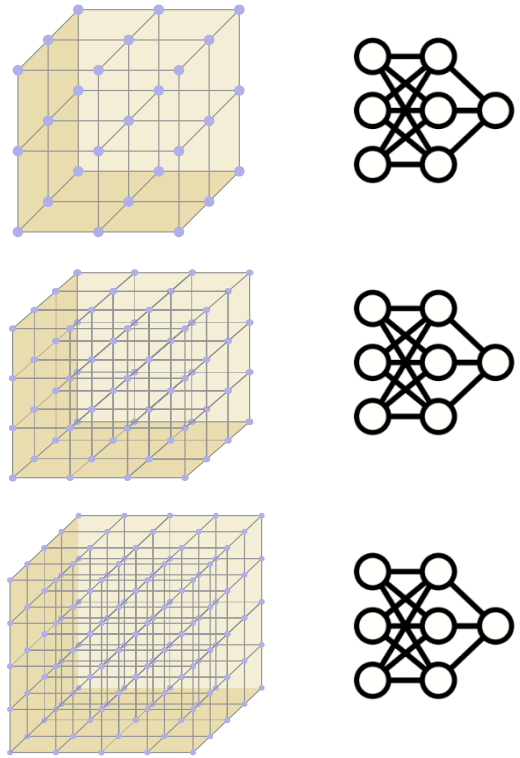
A single MLP

Predicted Poses

GT Poses

iMAP [Sucar et al., ICCV'21]

NICE-SLAM



Feature grids + tiny MLPs

Pretrained tiny MLPs → **Fast convergence**

Local update → **No forgetting problem**

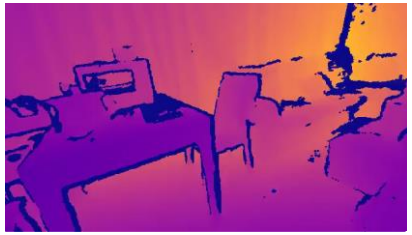
Predicted Poses

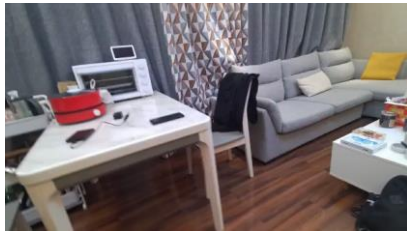
GT Poses

Zhu et al. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. CVPR 2022.

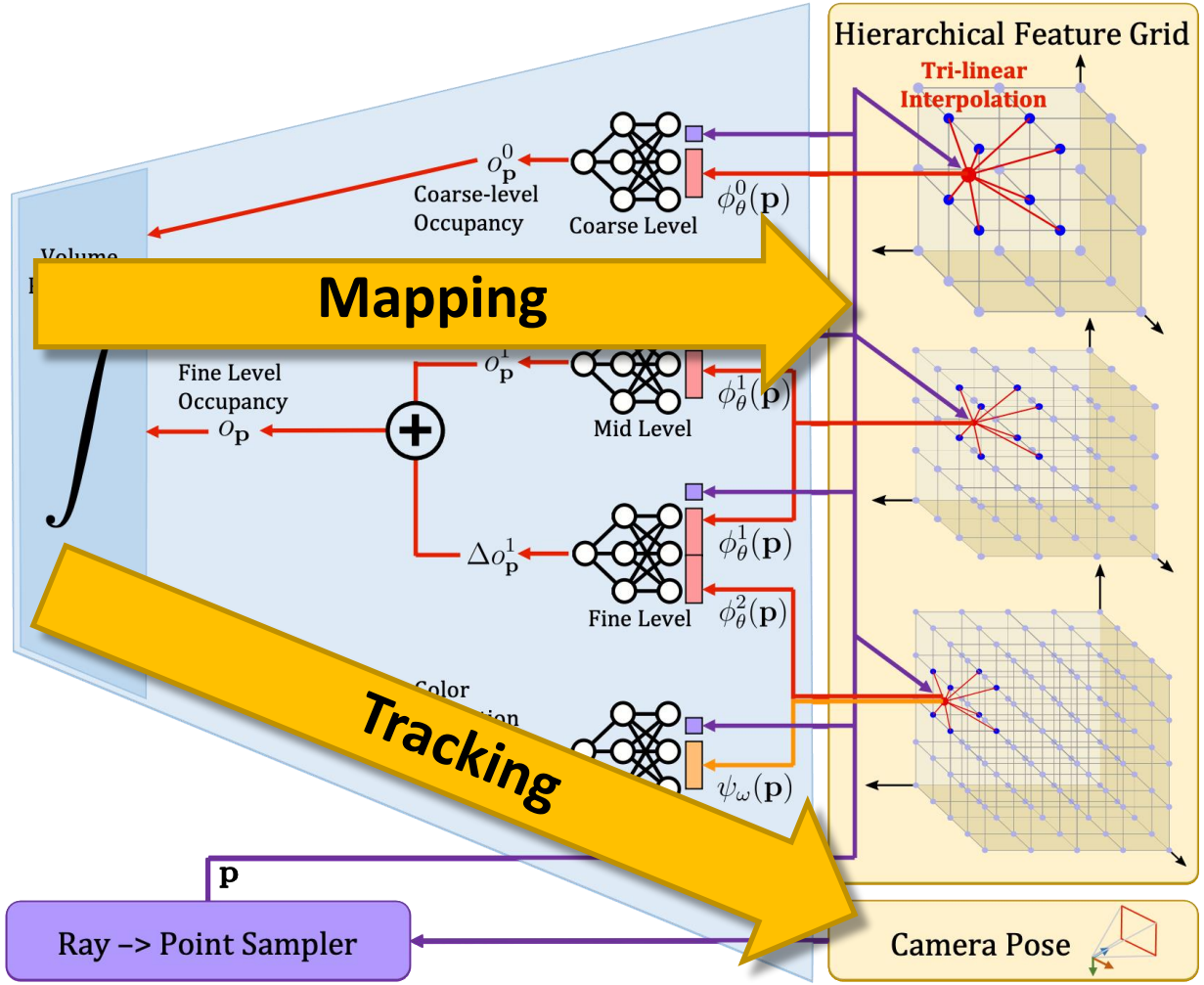
Pipeline

Input Depth

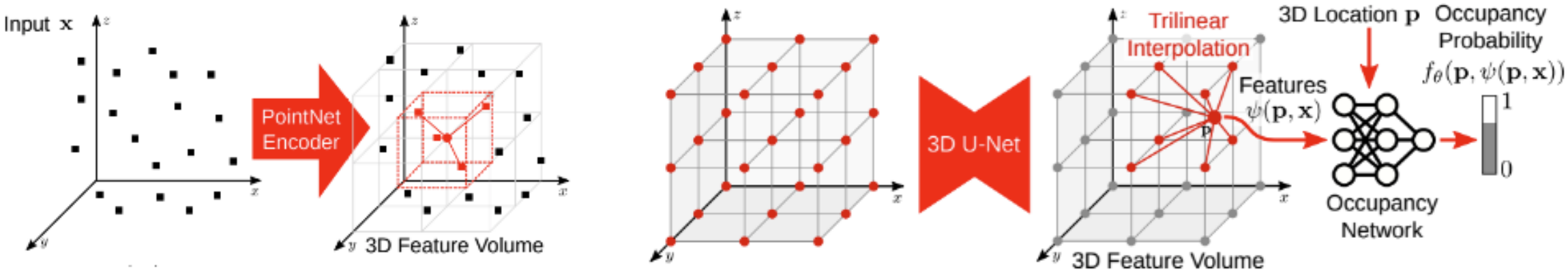


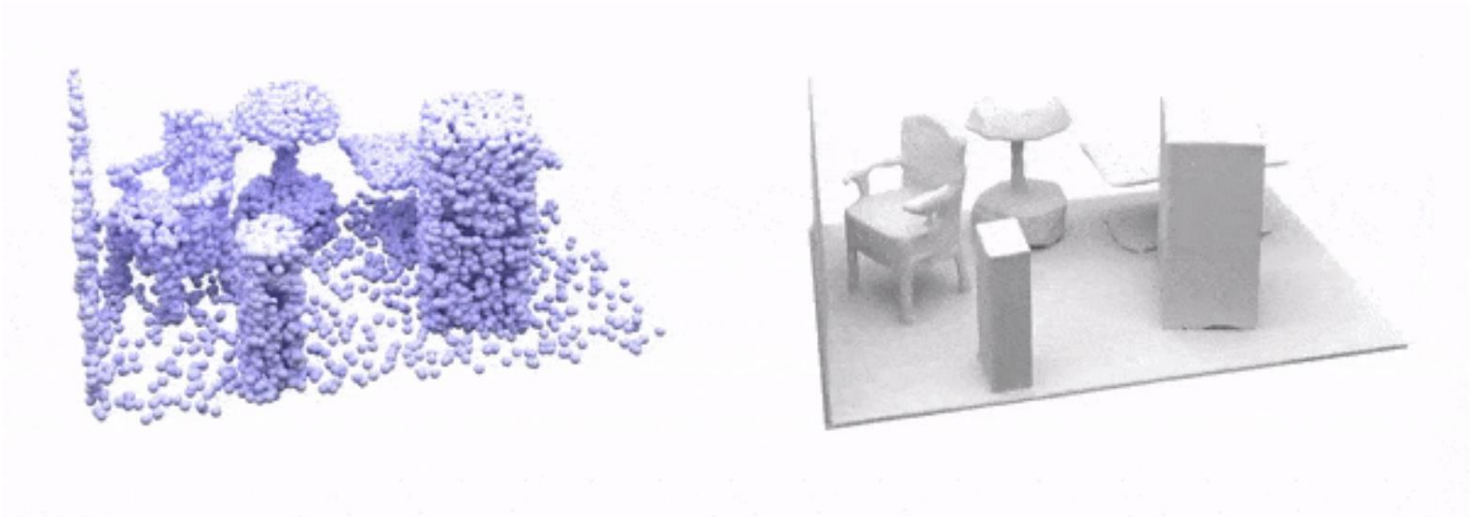


Input RGB



Pretraining





Convolutional occupancy network [Peng et al. 2020]

iMAP*

(our re-implementation of iMAP)

NICE-SLAM

4x Speed

Predicted Poses

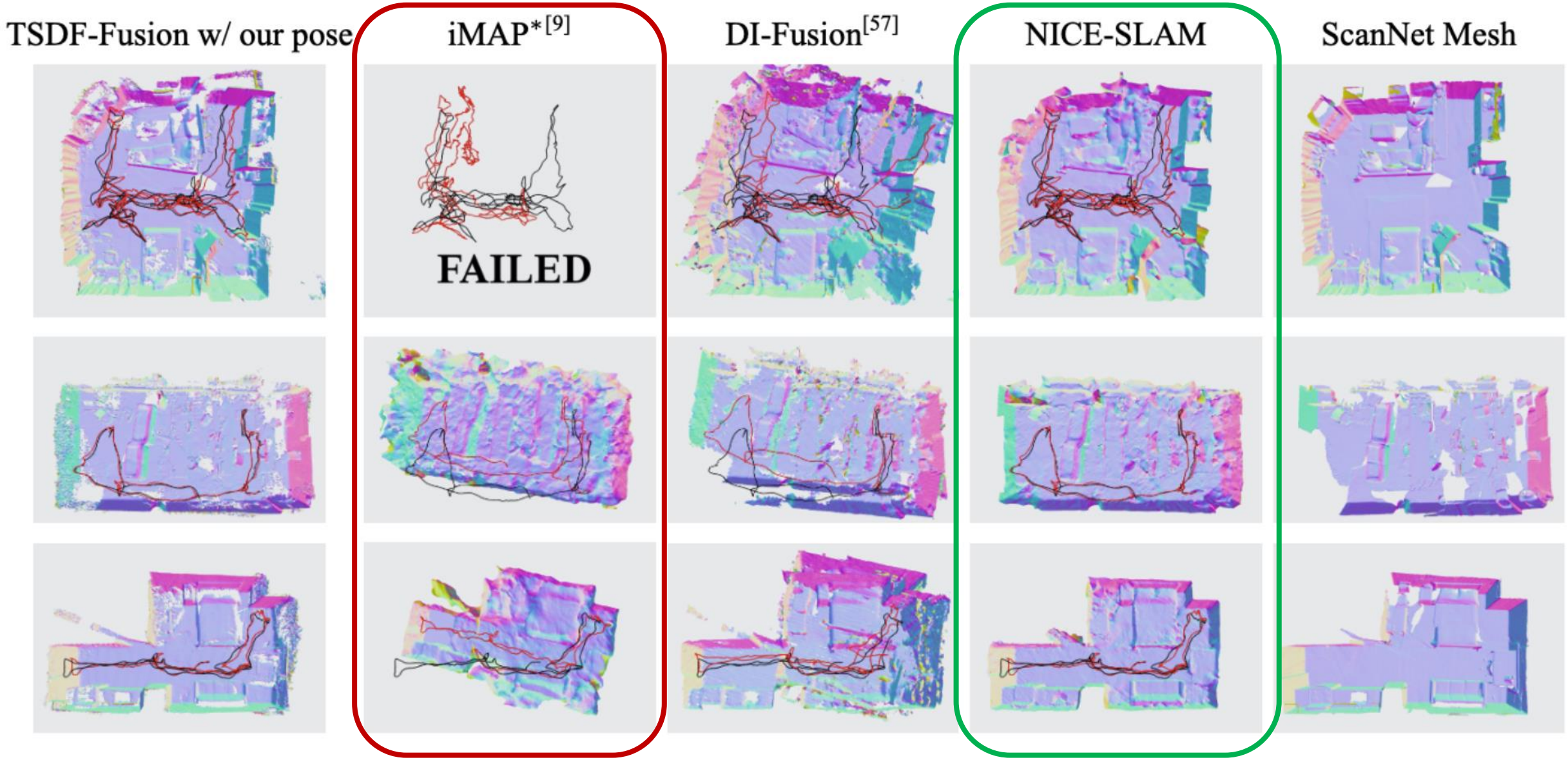
GT Poses

iMAP*
(our re-implementation of iMAP)

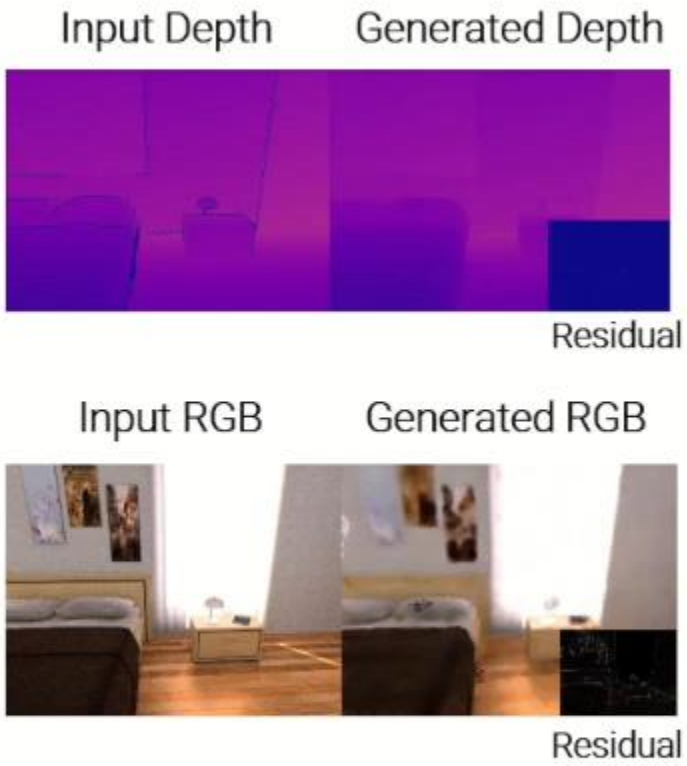
NICE-SLAM

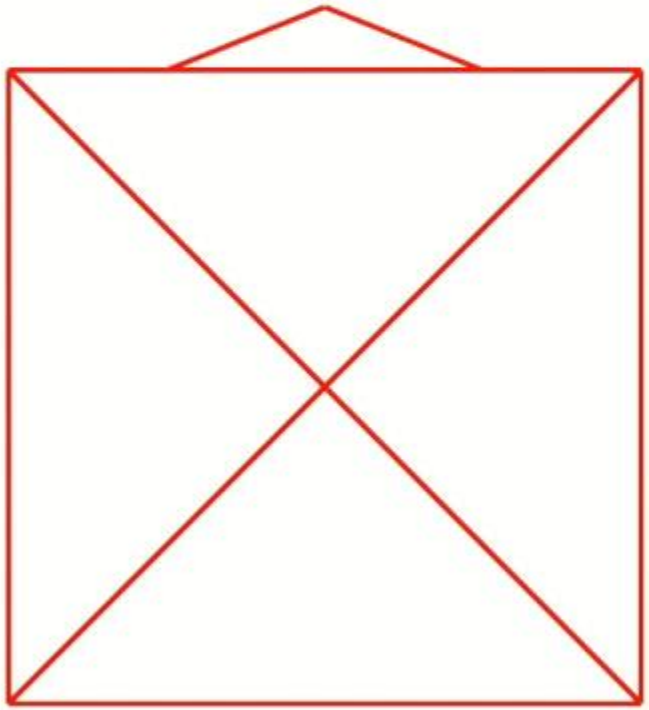
10x Speed

Comparison


TSDF-Fusion w/ our pose
iMAP*[9]
DI-Fusion[57]
NICE-SLAM
ScanNet Mesh
FAILED

Robustness to Dynamic Objects





Accuracy and Efficiency

Camera Tracking Results on TUM RGB-D. ATE RMSE[cm]

	fr1/desk	fr2/xyz	fr3/office
iMAP [46]	4.9	2.0	5.8
iMAP* [46]	7.2	2.1	9.0
DI-Fusion [16]	4.4	2.3	15.6
NICE-SLAM	2.7	1.8	3.0
BAD-SLAM [42]	1.7	1.1	1.7
Kintinuous [59]	3.7	2.9	3.0
ORB-SLAM2 [26]	**1.6**	**0.4**	**1.0**

Computation & Runtime

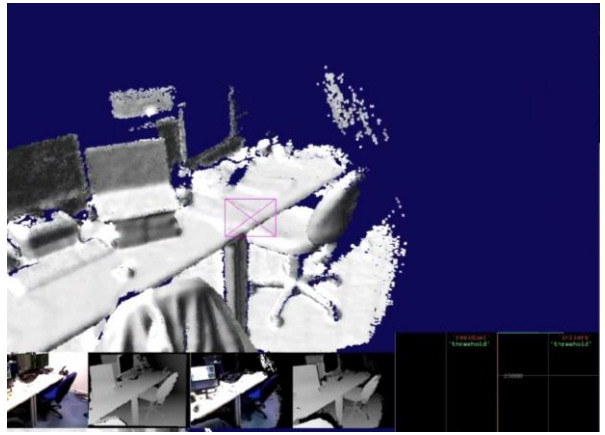
	FLOPs $[\times 10^3]\downarrow$	Tracking [ms]$\downarrow$	Mapping [ms]$\downarrow$
iMAP [46]	443.91	101	448
NICE-SLAM	**104.16**	**47**	**130**

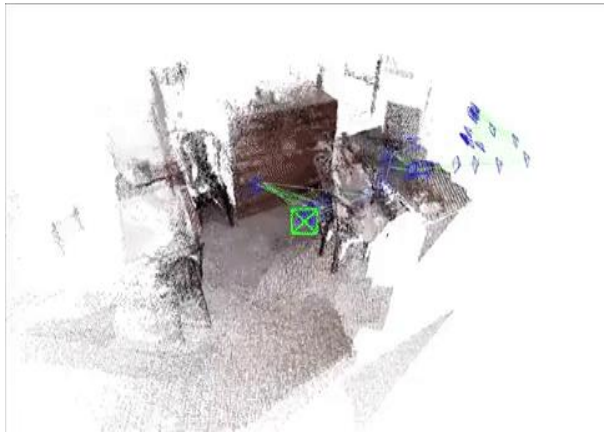
Q2.2

Can we use light-weight ToF sensors?

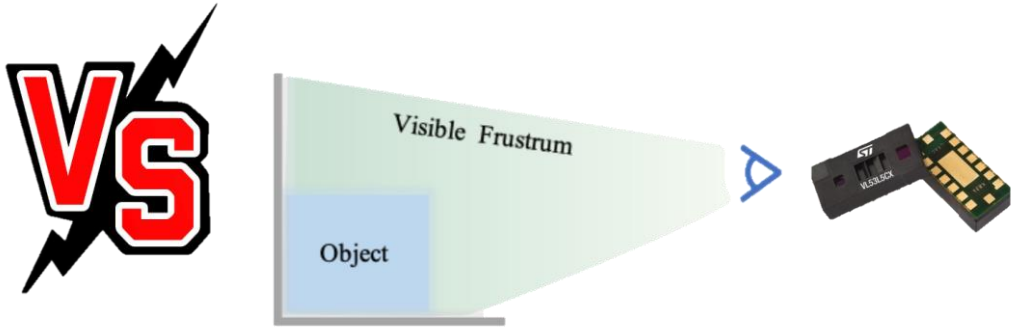
Motivation

Dense SLAM





Light-Weight ToF Sensor



Heavily relies on

Microsoft Kinect

Intel Realsense

Low-resolution depth distributions

- Expensive
- Heavy-sized
- Not common on mobile devices

+ Cheap
+ Compact
+ Widely available on mobile devices

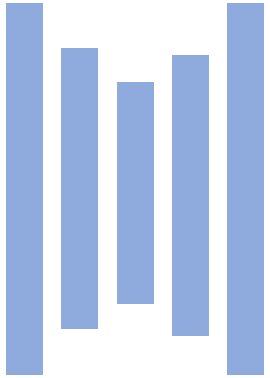
Challenges

RGB/Light-Weight ToF

Unsuitable

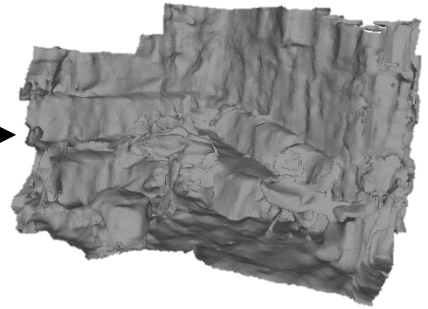
Existing RGB-D SLAM System

Failure



DELTAR

Depth Estimation



Poor Results

+ High-resolution with rich details

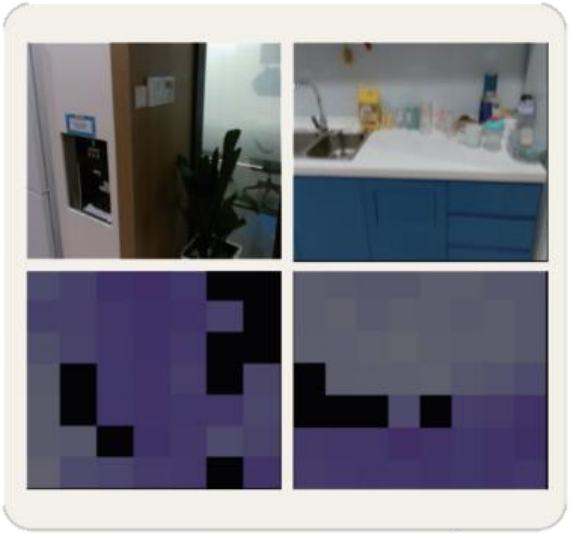
- Not accurate & Containing artifacts

- Temporal inconsistency

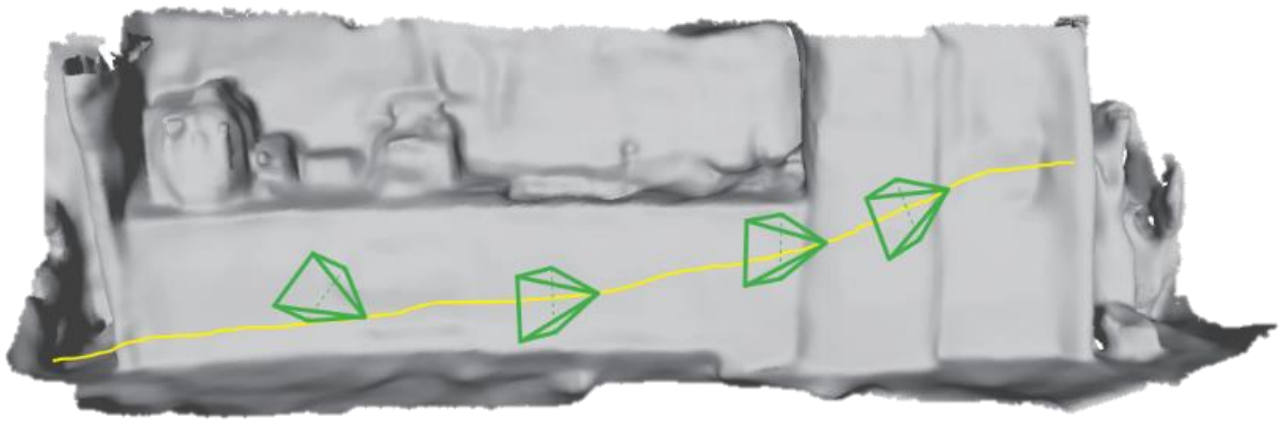
Li et al. DELTAR: Depth Estimation from a Light-Weight ToF Sensor and RGB Image. ECCV 2022.

We propose the first dense SLAM system with a light-weight ToF sensor

- Using a light-weight ToF instead of a heavy-sized depth sensor
- Cooperating with a monocular camera



RGB + Light-Weight ToF



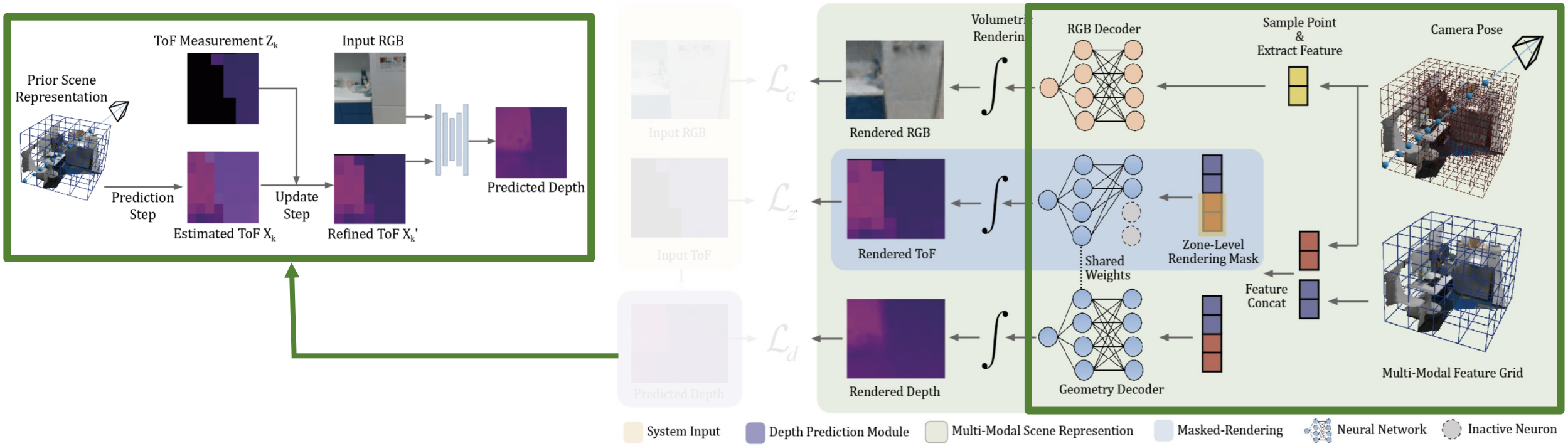
Camera Tracking and Dense Reconstruction

Liu et al. Multi-Modal Neural Radiance Field for Monocular Dense SLAM with a Light-Weight ToF Sensor. ICCV 2023 (Oral).

Our Solution

Depth Prediction boosted with Temporal Filtering

Multi-modal Scene Representation



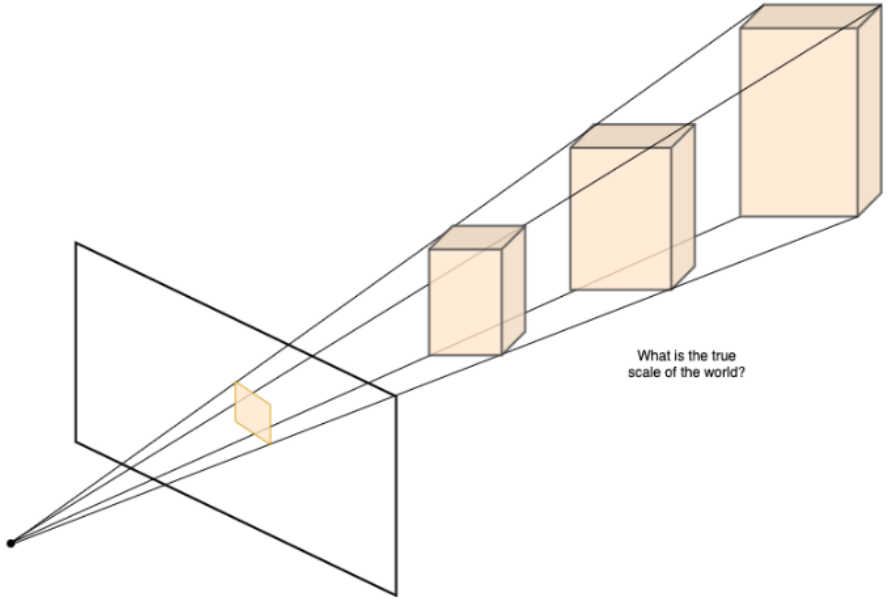
- A novel framework for dense SLAM working with light-weight ToF sensors
- The raw signals of light-weight ToF are fully leveraged
- Temporal information is exploited to deal with the noisy signals

Q2.3

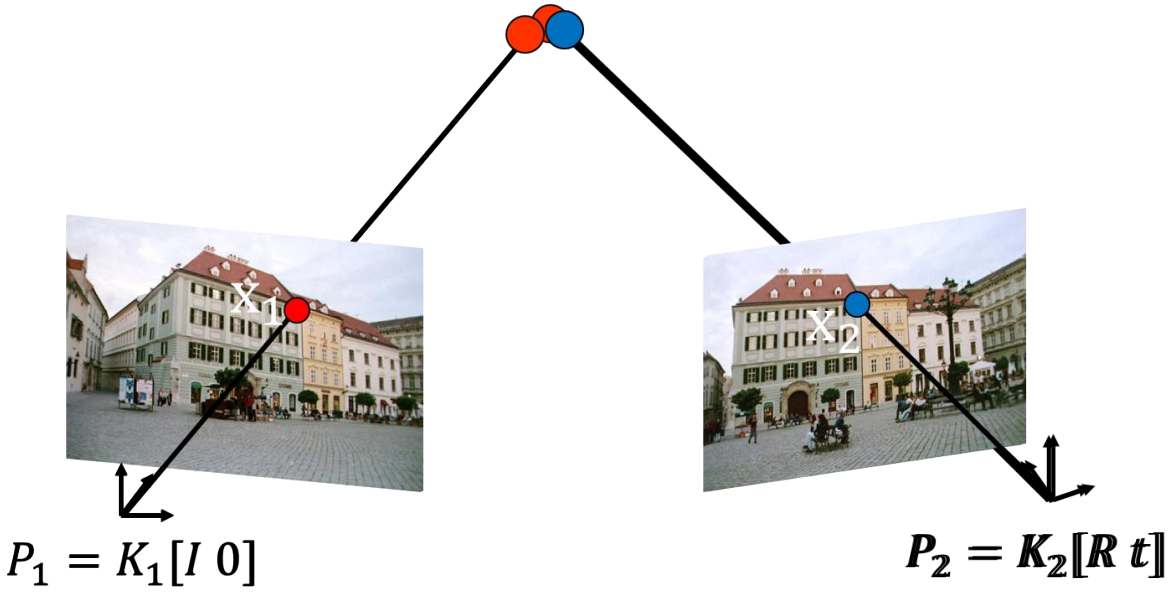
Can we just utilize the RGB sequences?

Challenges

Depth ambiguity



Lack of geometry constraint



Monocular Depth Estimation +

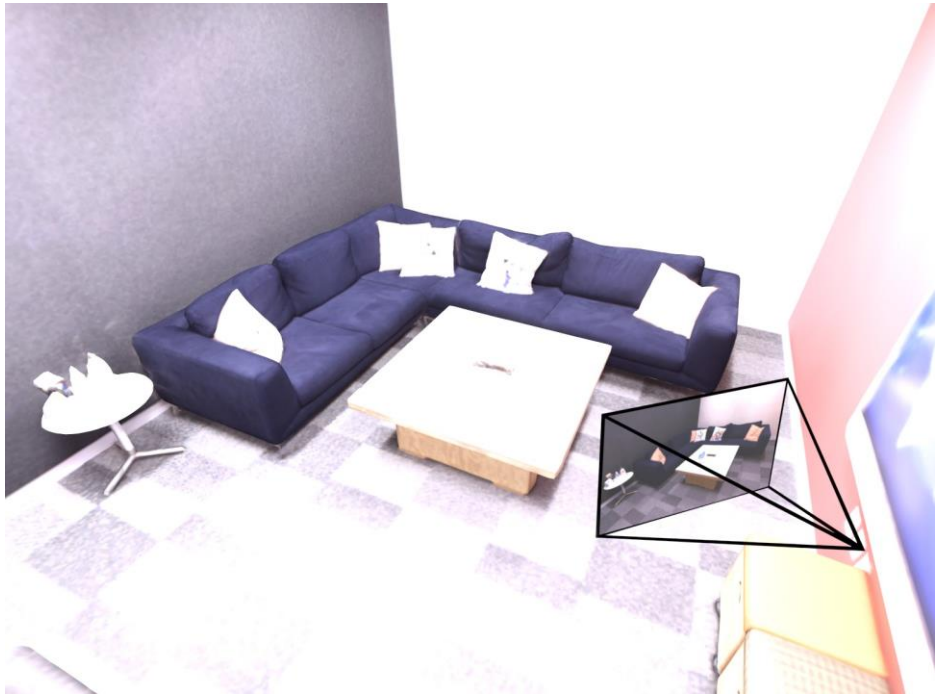
Scale&Shift-Invariant Depth Loss

RGB Warping loss +

Explicit flow constraint

Pipeline

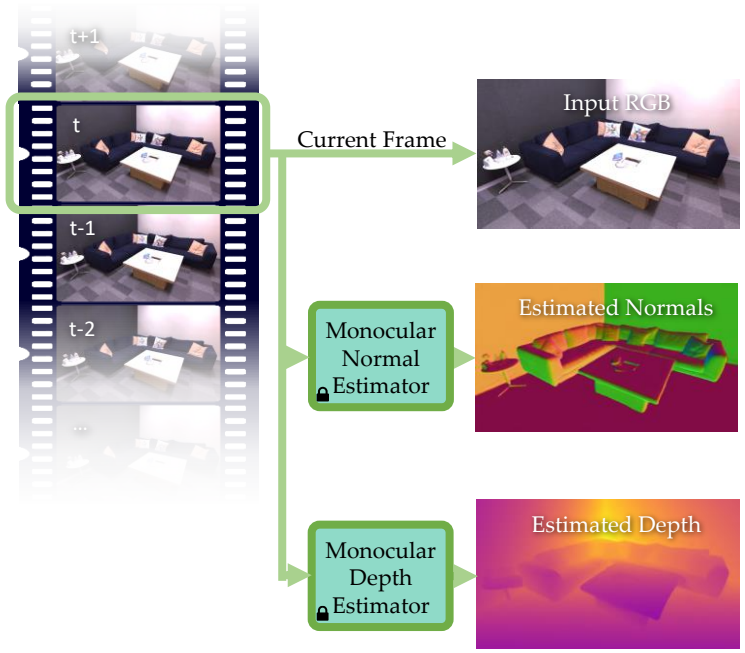
Input RGB Stream



Zhu et al. NICER-SLAM: Neural Implicit Scene Encoding for RGB SLAM. Arxiv 2023.

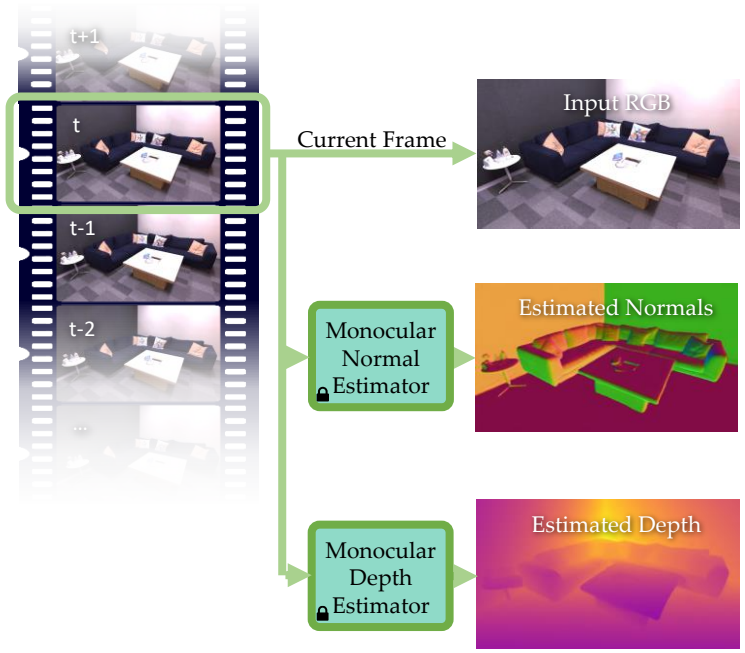
Pipeline

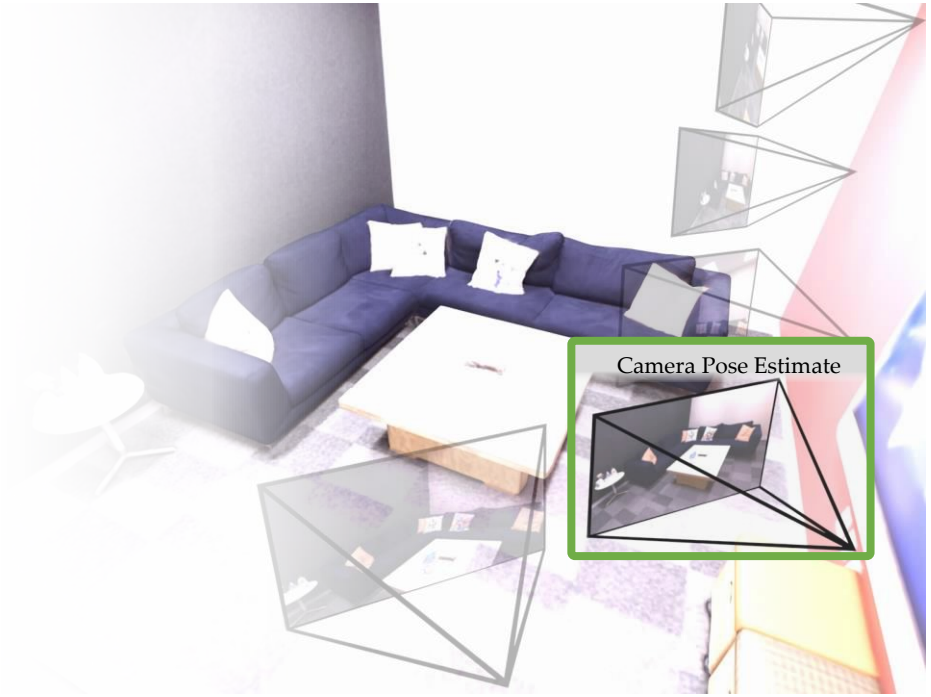
Input RGB Stream



Pipeline

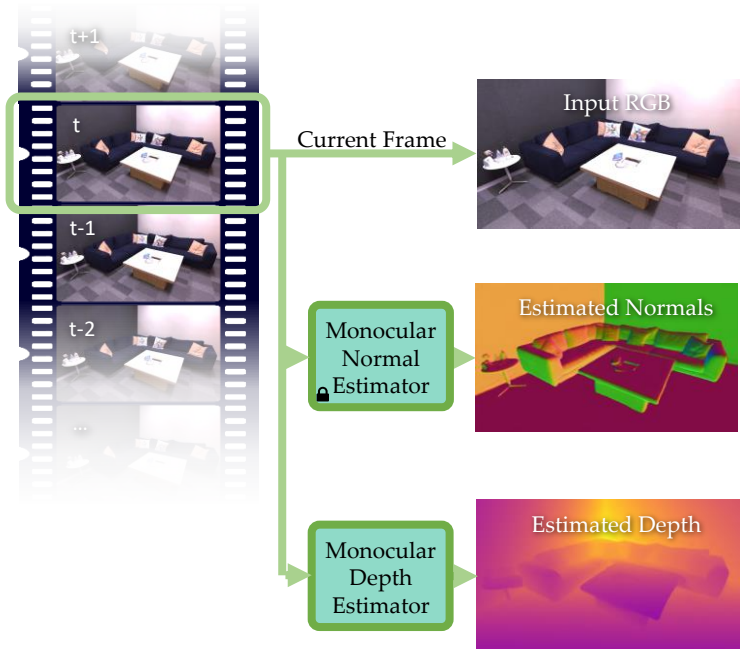
Input RGB Stream

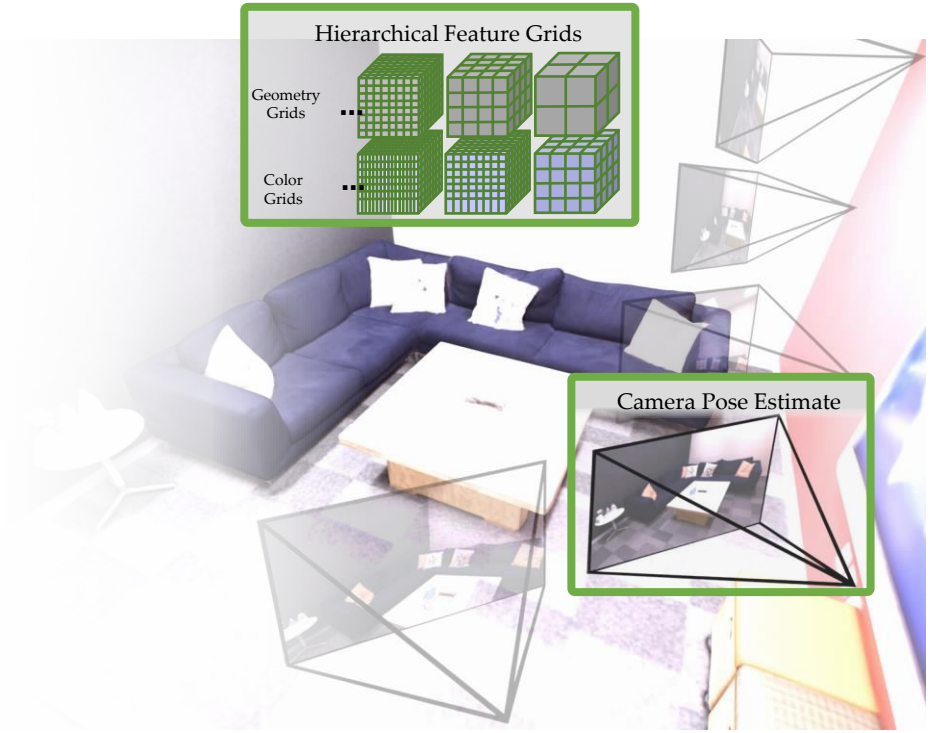




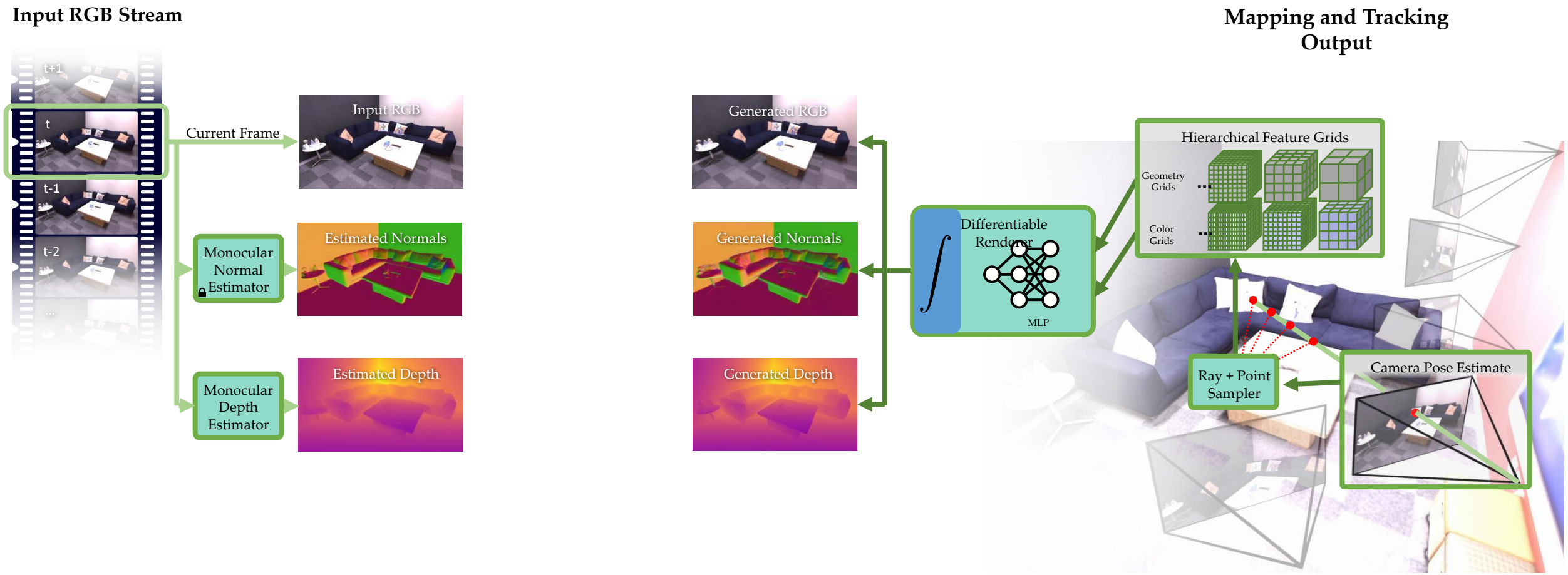
Pipeline

Input RGB Stream

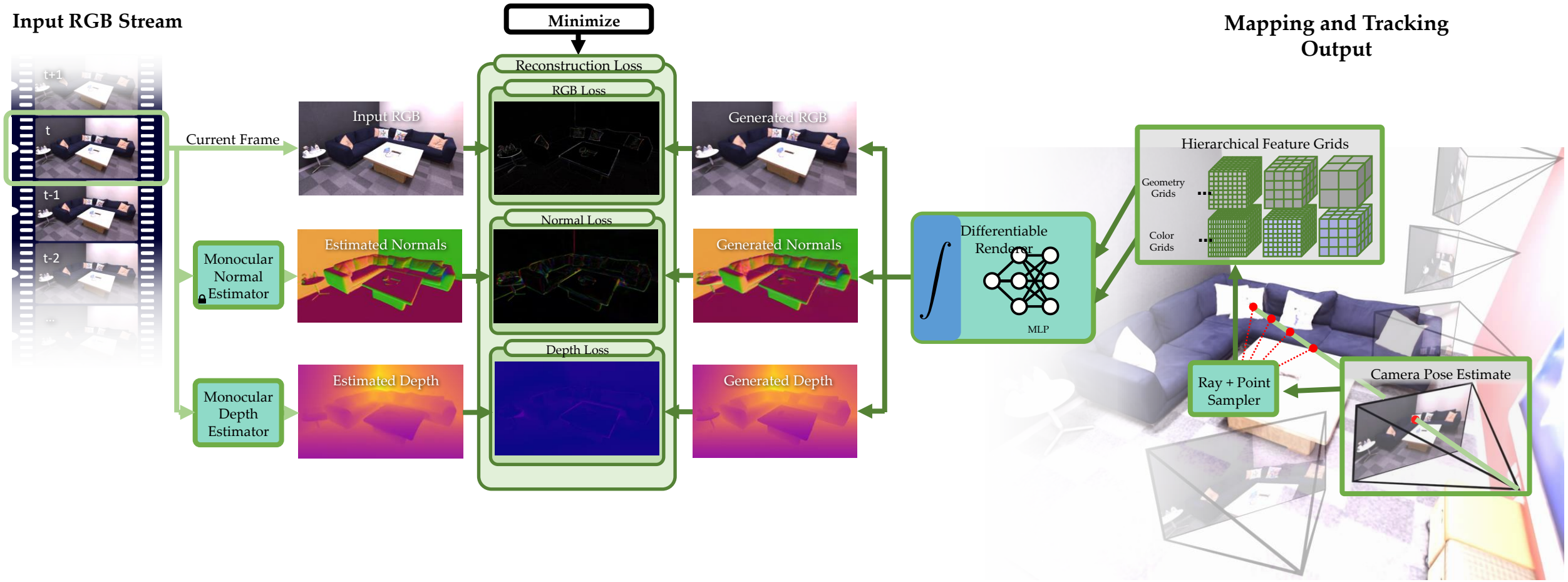




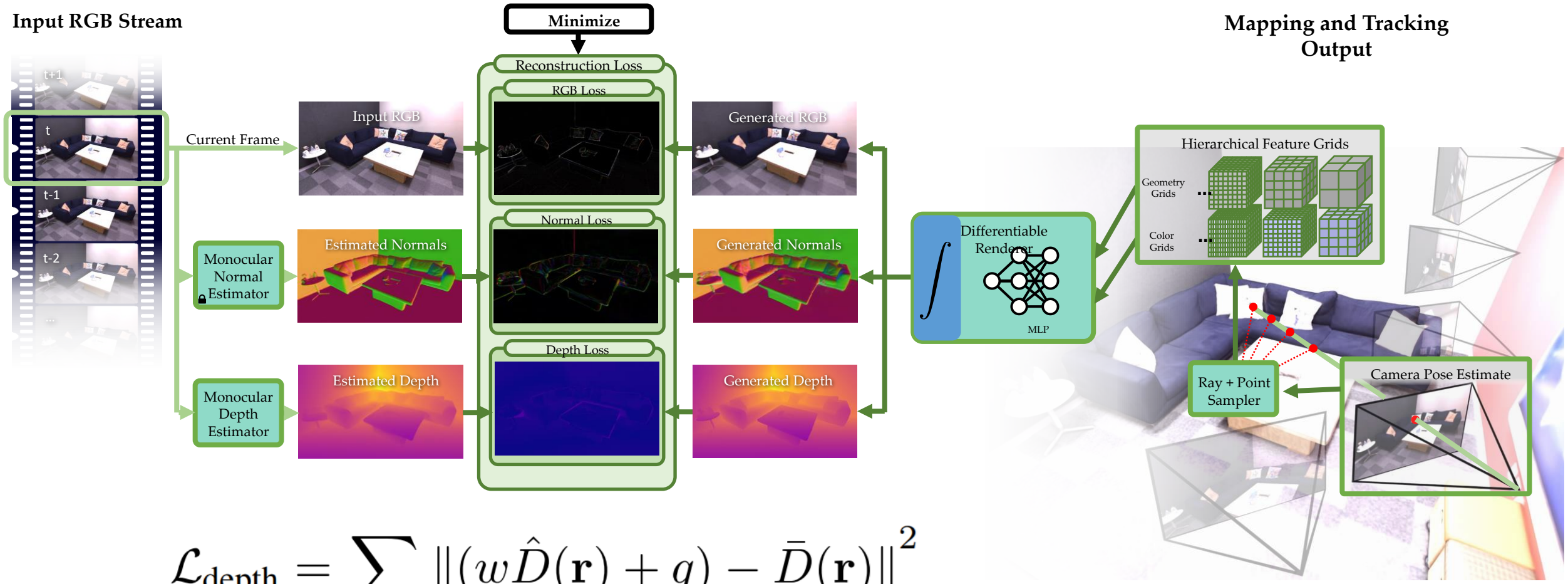
Pipeline


Input RGB Stream
t+1
t
t-1
t-2
Current Frame
Input RGB
Monocular Normal Estimator
Estimated Normals
Monocular Depth Estimator
Estimated Depth
Mapping and Tracking Output
Generated RGB
Generated Normals
Generated Depth
Differentiable Renderer
MLP
Hierarchical Feature Grids
Geometry Grids
Color Grids
Ray + Point Sampler
Camera Pose Estimate

Pipeline


Input RGB Stream
t+1
t
t-1
t-2
Current Frame
Monocular Normal Estimator
Monocular Depth Estimator
Input RGB
Estimated Normals
Estimated Depth
Minimize
Reconstruction Loss
RGB Loss
Normal Loss
Depth Loss
Generated RGB
Generated Normals
Generated Depth
Differentiable Renderer
MLP
Mapping and Tracking Output
Hierarchical Feature Grids
Geometry Grids
Color Grids
Ray + Point Sampler
Camera Pose Estimate

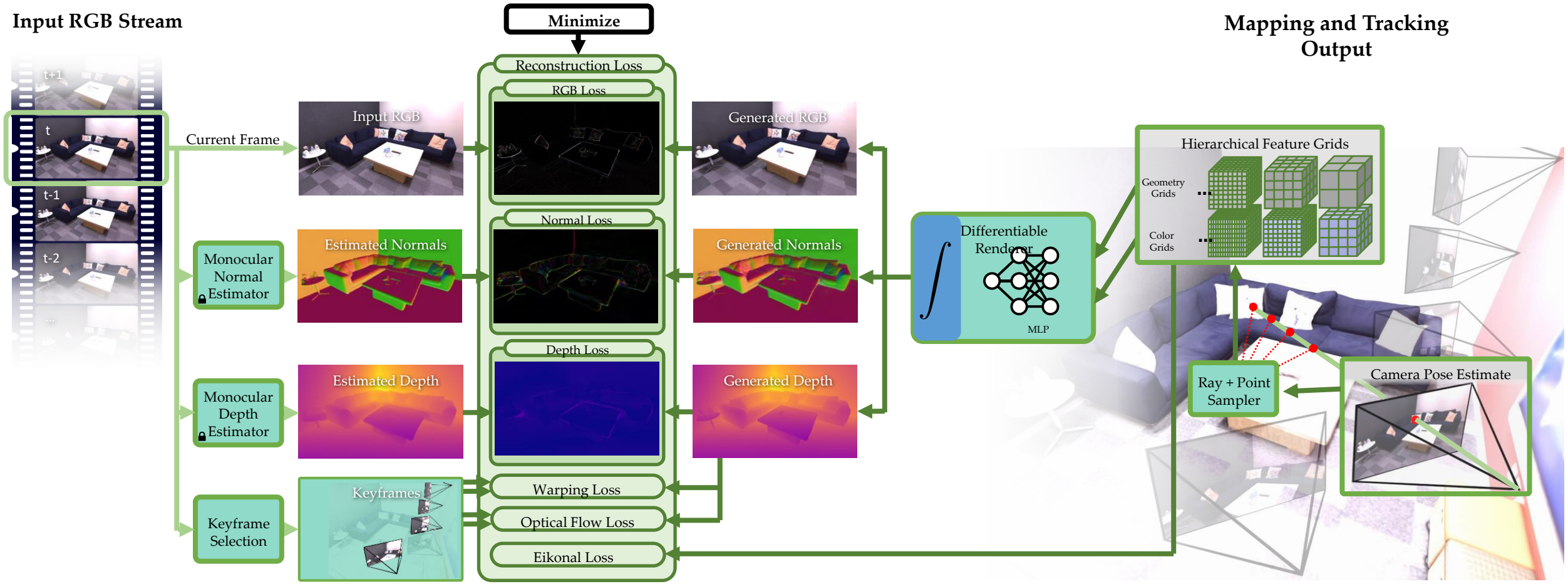
Pipeline



$$\mathcal{L}_{\text{depth}} = \sum_{\mathbf{r} \in \mathcal{R}} \left\| (w\hat{D}(\mathbf{r}) + q) - \bar{D}(\mathbf{r}) \right\|^2$$

Scale&Shift-Invariant Depth Loss

Pipeline



$$\mathcal{L}_{\text{flow}} = \sum_{\mathbf{r}_m \in \mathcal{R}} \sum_{n \in \mathcal{K}_m} \|(\mathbf{r}_m - \mathbf{r}_n) - \mathbf{GM}(\mathbf{r}_{m \to n})\|_1$$

$$\mathcal{L}_{\text{warp}} = \sum_{\mathbf{r}_m \in \mathcal{R}} \sum_{n \in \mathcal{K}_m} \|C(\mathbf{r}_m) - C(\mathbf{r}_{m \to n})\|_1$$

Results

		rm-0	rm-1	rm-2	off-0	off-1	off-2	off-3	off-4	Avg.
NeRF-SLAM	Acc.[cm]↓	11.84	10.62	11.86	9.32	14.40	11.54	16.31	11.11	12.13
	Comp.[cm]↓	5.63	5.88	9.22	13.29	10.17	6.95	7.81	5.26	8.03
	Comp.Rat.[< 5cm %]↑	61.13	68.19	47.85	37.64	56.17	66.20	55.67	61.86	56.84
	Normal Cons.[%]↑	63.39	53.31	57.52	64.09	57.13	57.06	59.73	58.59	58.85
DIM-SLAM*	Acc.[cm]↓	14.19	9.56	8.41	10.16	7.86	16.50	13.01	13.08	11.60
	Comp.[cm]↓	6.24	6.45	12.17	5.95	8.33	8.28	6.77	8.62	7.85
	Comp.Rat.[< 5cm %]↑	69.77	66.30	51.21	74.16	62.10	54.92	63.88	55.43	62.22
	Normal Cons.[%]↑	77.69	82.16	78.89	81.44	79.41	73.68	77.09	78.05	78.55
DROID-SLAM	Acc.[cm]↓	12.18	8.35	3.26	3.01	2.39	5.66	4.49	4.65	5.50
	Comp.[cm]↓	8.96	6.07	16.01	16.19	16.20	15.56	9.73	9.63	12.29
	Comp.Rat.[< 5cm %]↑	60.07	76.20	61.62	64.19	60.63	56.78	61.95	67.51	63.62
	Normal Cons.[%]↑	72.81	74.71	79.21	77.53	78.57	75.79	77.69	76.38	76.59
NICER-SLAM	Acc.[cm]↓	**2.53**	3.93	3.40	5.49	3.45	4.02	**3.34**	3.03	3.65
	Comp.[cm]↓	3.04	4.10	3.42	6.09	4.42	4.29	4.03	3.87	4.16
	Comp.Rat.[< 5cm %]↑	88.75	76.61	86.10	65.19	77.84	74.51	82.01	83.98	79.37
	Normal Cons.[%]↑	93.00	91.52	**92.38**	87.11	86.79	90.19	90.10	90.96	90.27

Results

	rm-0	rm-1	rm-2	off-0	off-1	off-2	off-3	off-4	Avg.
RGB-D input									
NICE-SLAM	1.69	2.04	1.55	0.99	0.90	1.39	3.97	3.08	1.95
Vox-Fusion	0.27	1.33	0.47	0.70	1.11	0.46	0.26	0.58	0.65
RGB input									
COLMAP	0.62	23.7	0.39	0.33	0.24	0.79	**0.14**	1.73	3.49
TANDEM	0.54	0.43	0.47	0.61	0.33	5.42	0.68	0.75	1.15
DSO	**0.26**	0.25	**0.19**	0.38	0.20	2.53	0.22	**0.38**	0.55
Orbeez-SLAM	0.34	0.41	0.27	0.36	F	F	0.294	2.89	0.76
NeRF-SLAM	17.26	11.94	15.76	12.75	10.34	14.52	20.32	14.96	14.73
DIM-SLAM	0.48	0.78	0.35	0.67	0.37	0.36	0.33	0.36	0.46
DIM-SLAM*	1.06	0.49	0.32	0.43	0.26	0.65	0.55	3.69	0.93
DROID-SLAM	0.34	**0.13**	0.27	**0.25**	0.42	**0.32**	0.52	0.40	**0.33**
DROID-SLAM*	0.58	0.58	0.38	1.06	0.40	0.70	0.53	1.33	0.70
NICER-SLAM	1.36	1.60	1.14	2.12	3.23	2.12	1.42	2.01	1.88

Summary

- We can edit neural radiance fields with compositional NeRF representations.

- We can conduct online 3D tracking and mapping using NeRF with various input signals.

- Future works:
  - Attribute-compositional representation
  - Scaling up to large scenarios
  - Efficient and robust as traditional SLAM
  - …

Thank you!