



# Neural Scene Representations for SLAM

Leonard Bruns

# Overview

- Introduction: neural fields, NeRFs, and beyond
- Neural SLAM: SLAM with neural scene representations
- Neural graph mapping: efficient loop closure in neural SLAM
- Outlook: open research problems



# Introduction

Neural Fields, NeRFs, and beyond



## Introduction

# Shape and Scene Representations

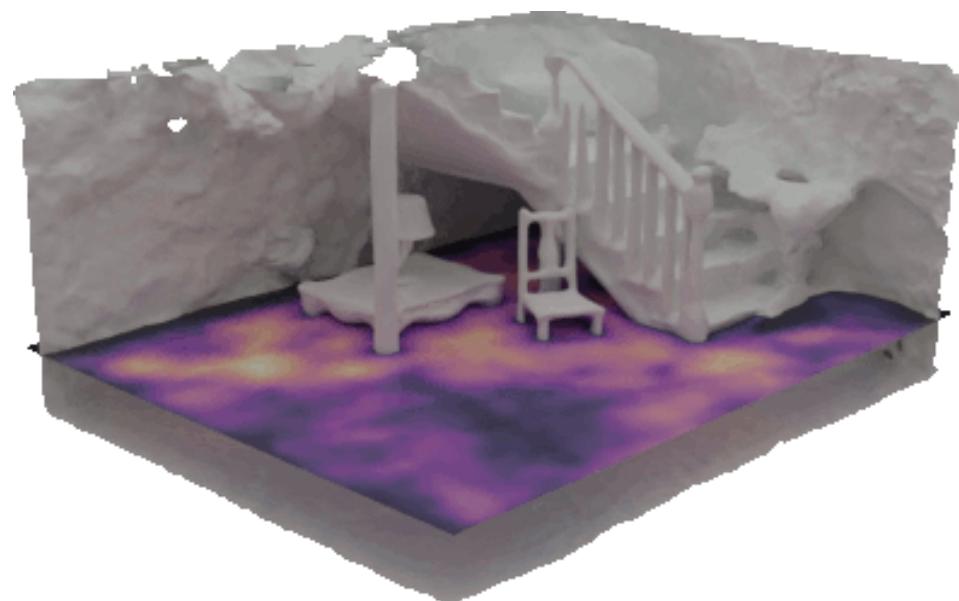
- Various shape / scene representations exist
- Surface representations
  - Store surface position and quantities on surface
  - Examples
    - Meshes
    - Point clouds
- Volumetric representations
  - Store quantities in volume
  - Examples
    - Voxel grid (dense, octrees, voxel hashing)
    - Neural fields
    - Set of Gaussians

## Introduction

# Neural Fields

- Core idea: represent 3D shape by a simple neural network

$$f_{\theta} : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \times \mathbb{R}$$
$$\boldsymbol{x} \mapsto (\boldsymbol{c}, \rho)$$



- Map 3D point  $\boldsymbol{x}$  to color  $\boldsymbol{c}$  and quantity  $\rho$  (e.g., SDF, occupancy, etc.)
- Properties
  - Continuous
  - Differentiable
  - Adaptive → potentially memory efficient

## Introduction

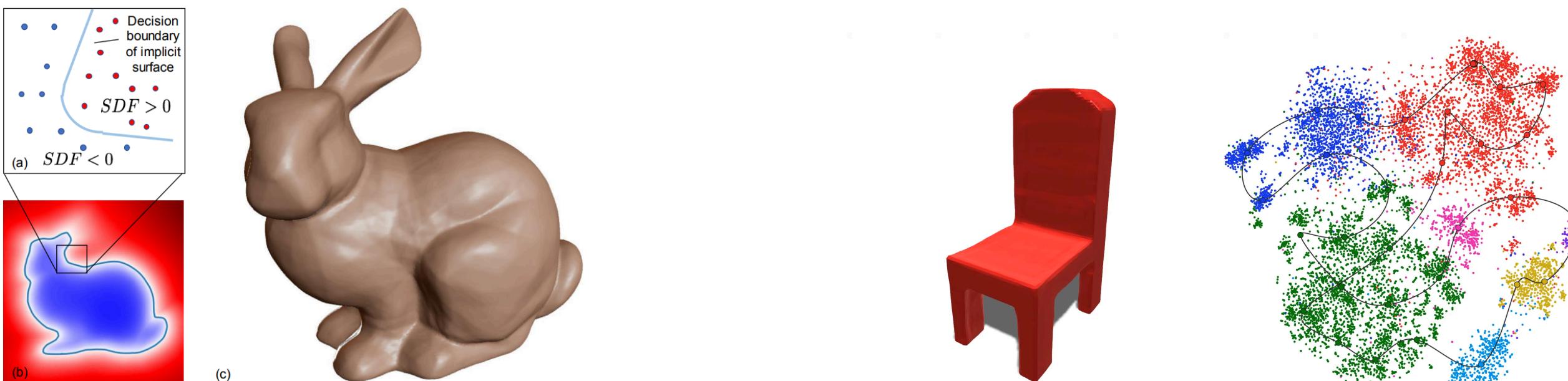
# Neural Scene Representations

## Early Works

- DeepSDF [1], occupancy network [2], ...
- Trained using direct 3D supervision (i.e., from mesh collections)
- Demonstrate shape interpolation by adding additional latent input

$$f_{\theta} : \mathbb{R}^3 \times \mathbb{R}^N \rightarrow \mathbb{R}$$

$$(\mathbf{x}, \mathbf{z}) \mapsto \rho$$



[1] DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, Park et al., 2019

[2] Occupancy Networks: Learning 3D Reconstruction in Function Space, Mescheder et al., 2019

Video source: [https://www.youtube.com/watch?v=C\\_XNdGGs6qM](https://www.youtube.com/watch?v=C_XNdGGs6qM)

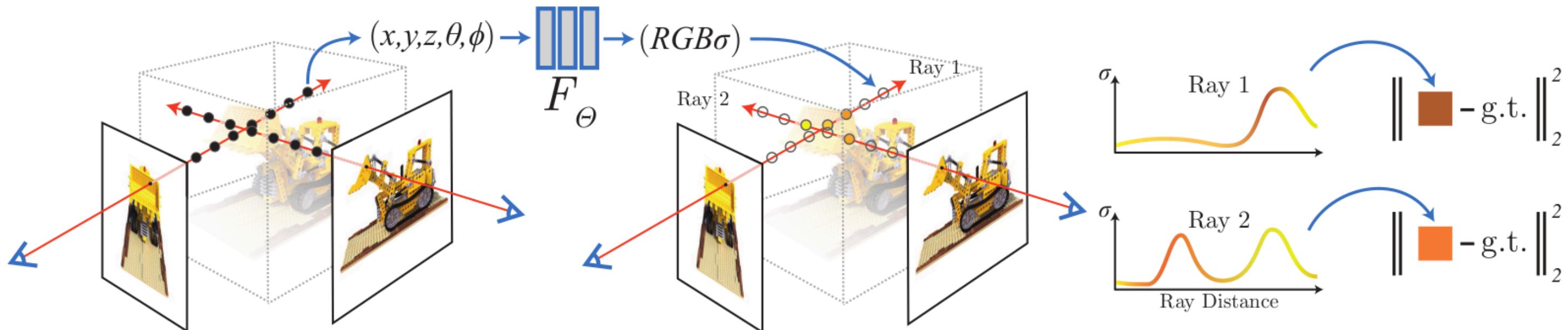
## Introduction

# Neural Scene Representations

## Neural Radiance Field (NeRF)

- Train neural field using indirect 2D supervision
- Camera poses fixed (e.g., from structure-from-motion)
- Add 2D view direction as additional input

$$f_{\theta} : \mathbb{R}^3 \times \mathbb{R}^2 \rightarrow \mathbb{R}^3 \times \mathbb{R}$$
$$(\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma)$$

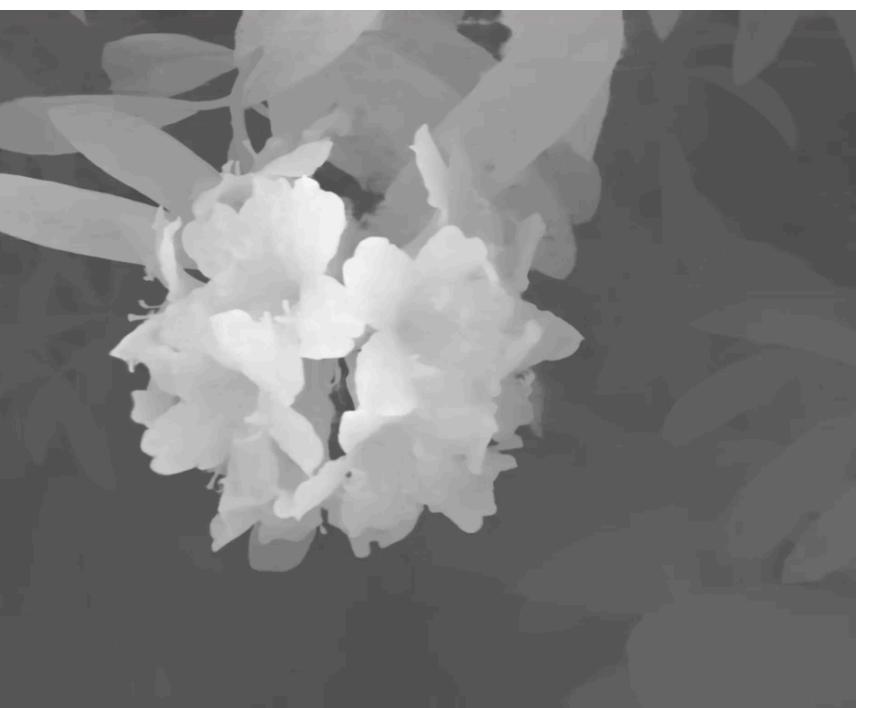
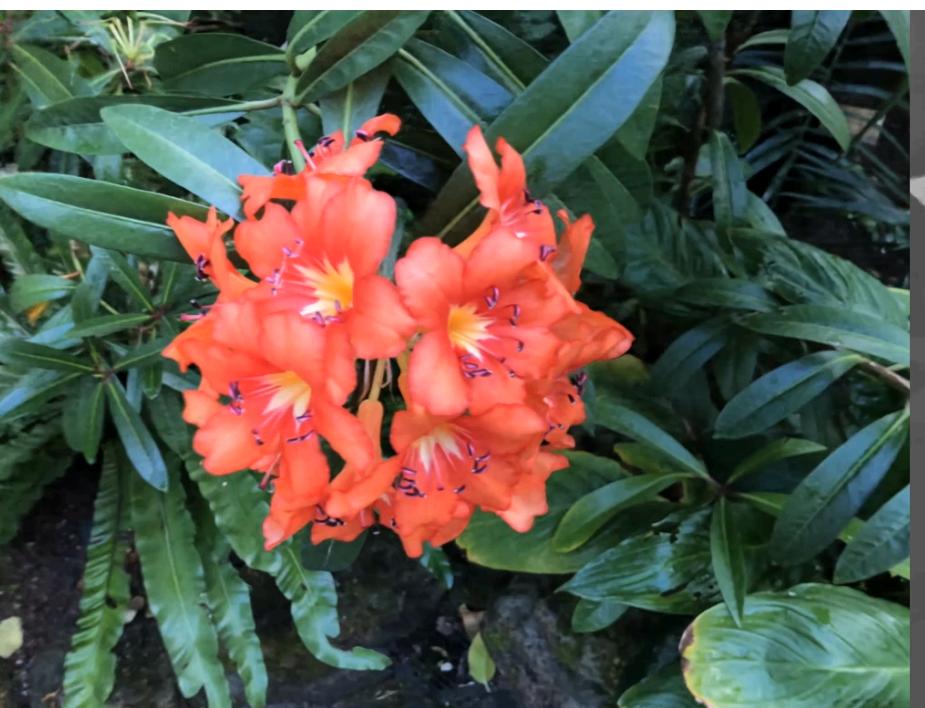
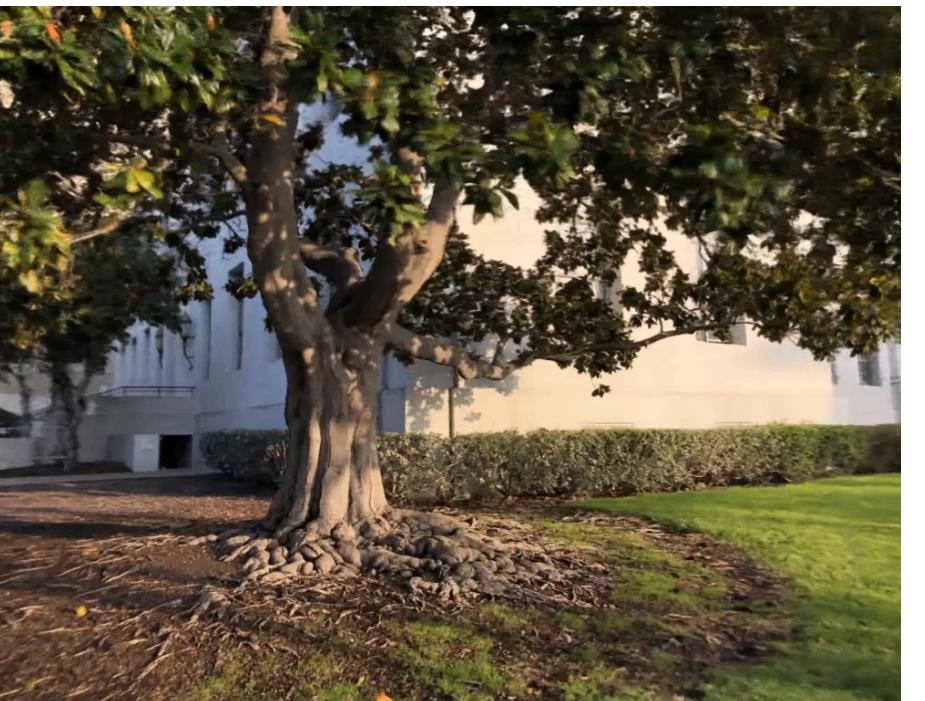


## Introduction

# Neural Scene Representations

## Neural Radiance Field (NeRF)

- NeRF Examples



## Introduction

# Neural Scene Representations

## Neural Radiance Field (NeRF)

- NeRF Examples



- Limitations

- Mainly limited to small forward-facing scenes
- Isolevel for surfaces not well-defined
- Optimization time (~hours)
- Rendering time (~seconds for single 1080p frame)

## Introduction

# Neural Scene Representations

## Follow-up Works

- Surfaces: NeuS [1]



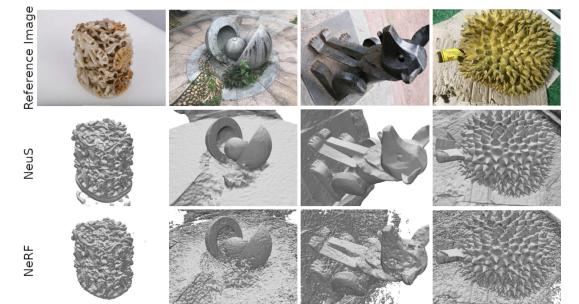
[1] NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction, Wang et al., 2021

## Introduction

# Neural Scene Representations

## Follow-up Works

- Surfaces: NeuS [1]



- Speed: Instant Neural Graphics Primitives [2]



- [1] NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction, Wang et al., 2021  
[2] Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, Müller et al., 2022

## Introduction

# Neural Scene Representations

## Follow-up Works

- Surfaces: NeuS [1]



- Speed: Instant Neural Graphics Primitives [2]



- Quality: Zip-NeRF [3]



[1] NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction, Wang et al., 2021

[2] Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, Müller et al., 2022

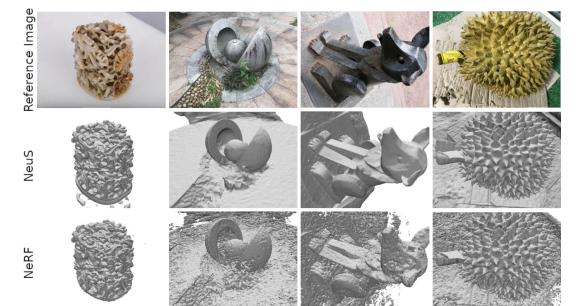
[3] Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields, Barron et al., 2023

## Introduction

# Neural Scene Representations

## Follow-up Works

- Surfaces: NeuS [1]



- Speed: Instant Neural Graphics Primitives [2]



- Quality: Zip-NeRF [3]



- [1] NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction, Wang et al., 2021  
[2] Instant Neural Graphics Primitives with a Multiresolution Hash Encoding, Müller et al., 2022  
[3] Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields, Barron et al., 2023

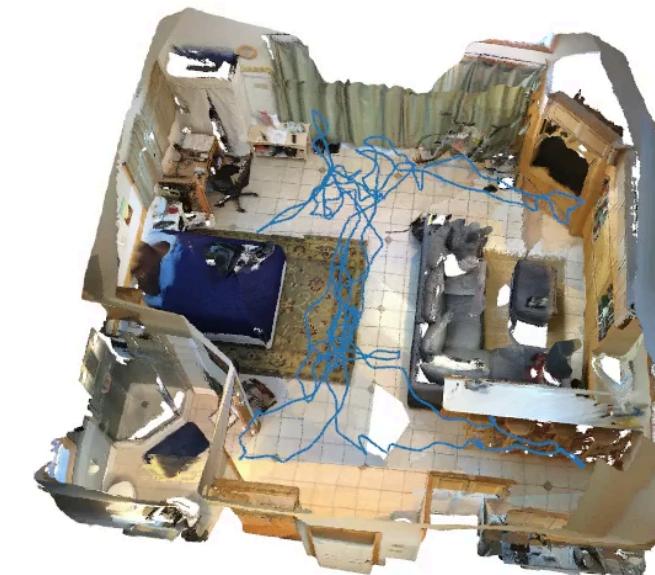
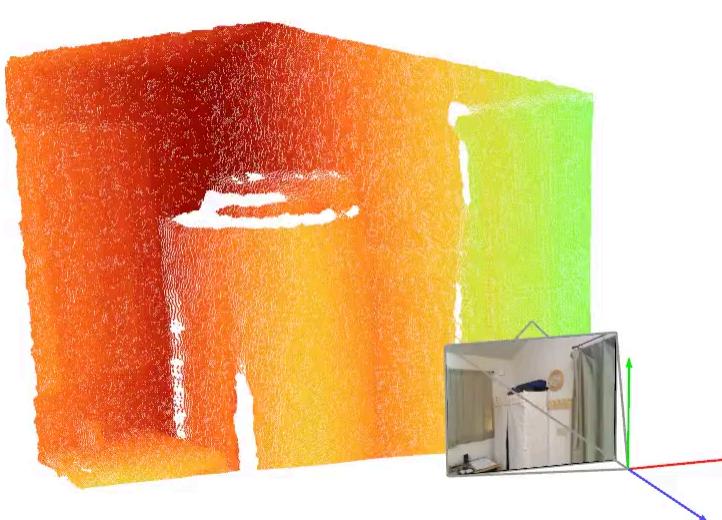


# Neural SLAM

SLAM with Neural Scene Representations

# Dense Visual SLAM

- Visual Simultaneous Localization and Mapping (SLAM)
  - Estimate camera trajectory of a moving camera from its image stream
  - Sparse keypoint-based maps
- Dense visual SLAM
  - Estimate dense 3D scene representation during localization (e.g., mesh)
  - RGB-D data
  - Volumetric scene representation
    - Classic approaches use voxel grids to represent 3D scene



## Neural SLAM

# iMAP

- First work demonstrating SLAM using neural field as scene representation
- Alternates between frame-to-map alignment and map refinement
- Compared to NeRF
  - Removed view-direction dependency
  - Smaller neural network
  - Depth supervision

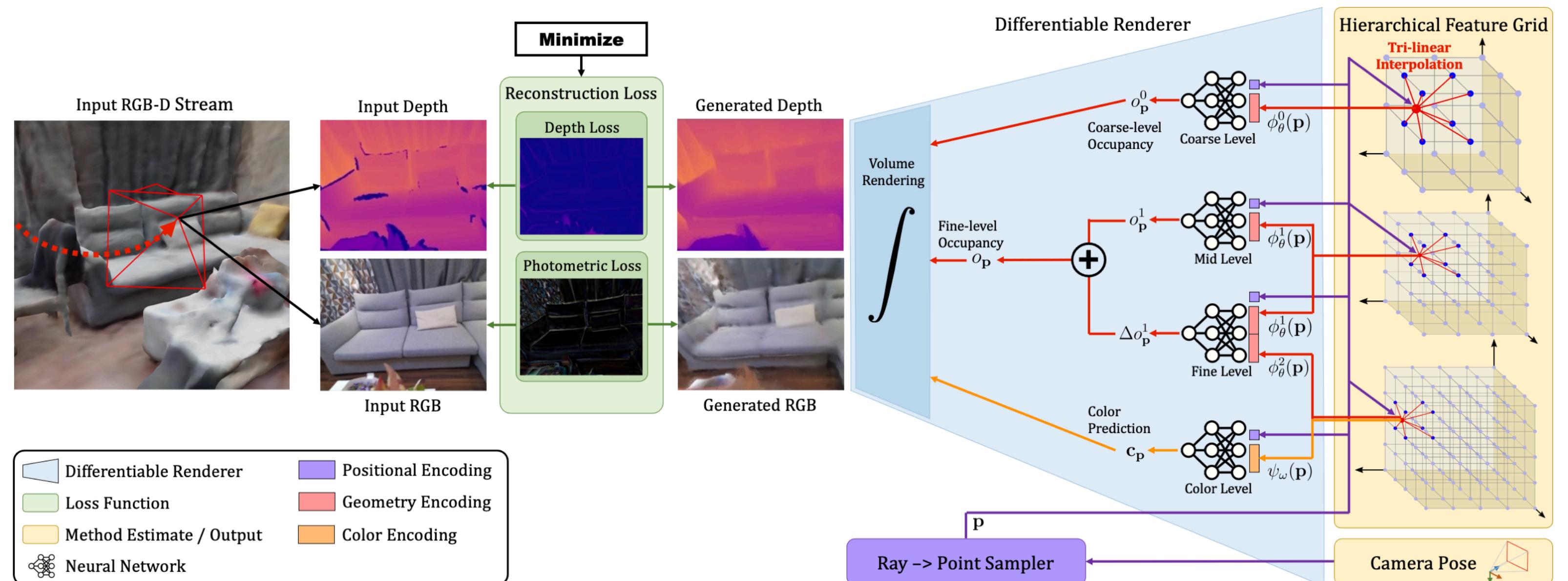


- Limitation: continuous reoptimization of previous areas required to prevent forgetting

# Neural SLAM

## NICE-SLAM

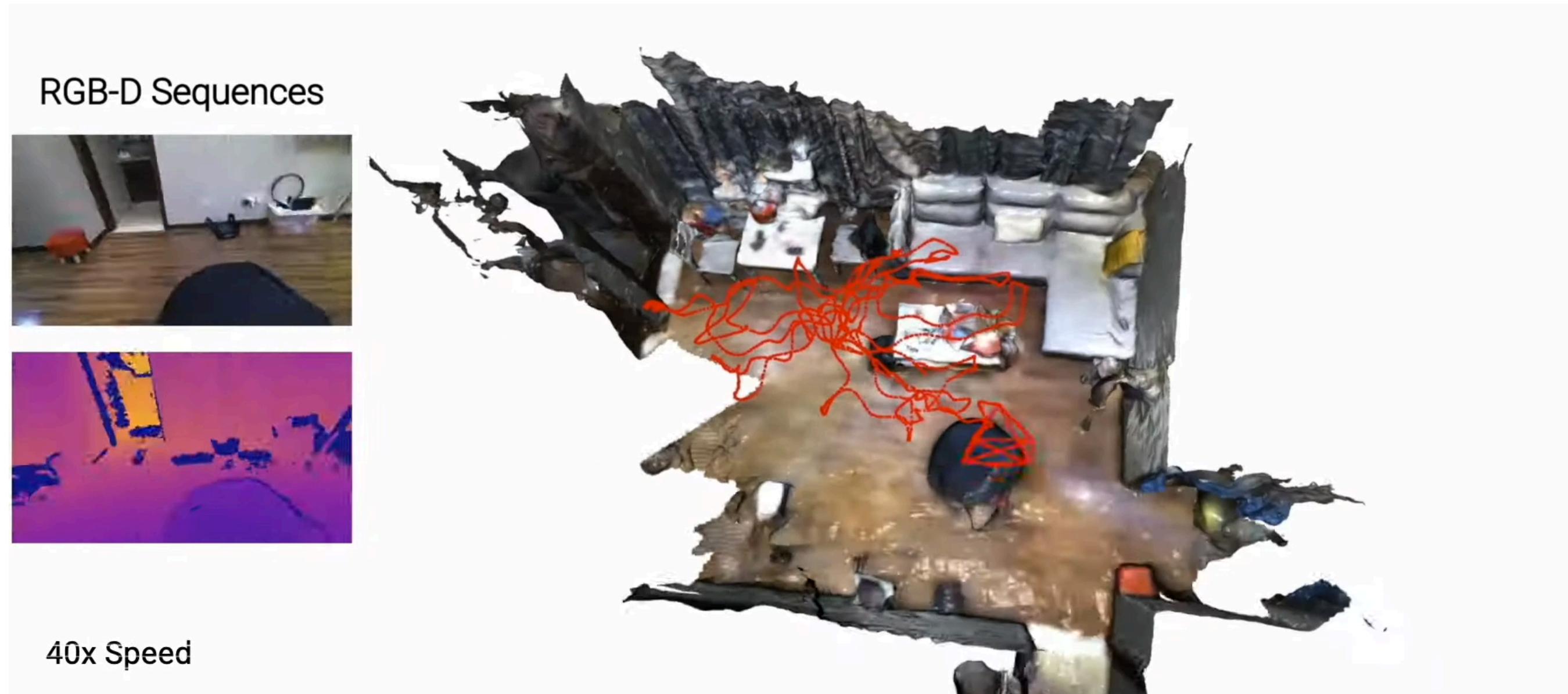
- Replaces single MLP by combination of feature grid and frozen, pretrained decoder  
→ No more forgetting, better scalability



## Neural SLAM

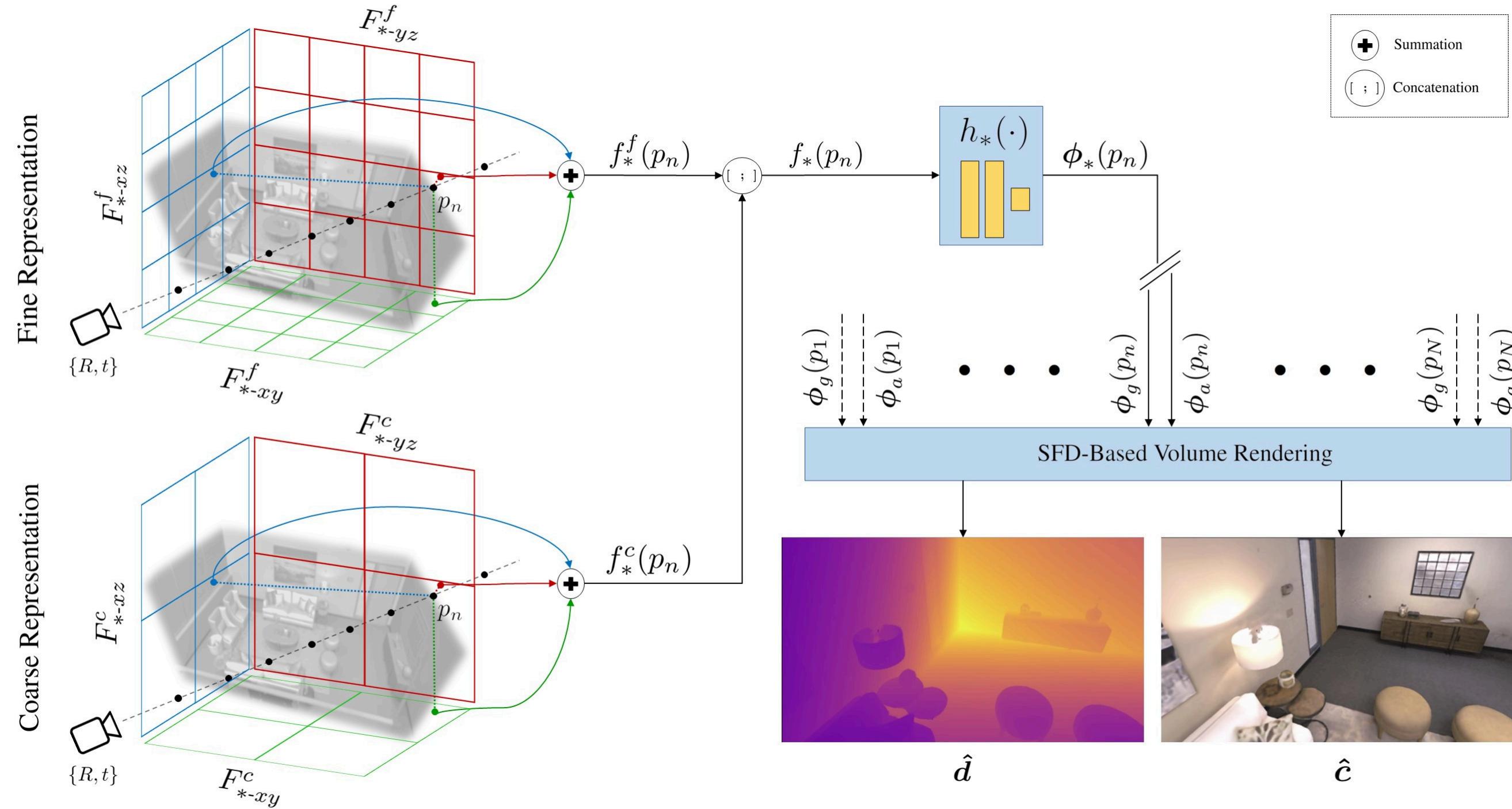
# NICE-SLAM

- Replaces single MLP by combination of feature grid and frozen, pretrained decoder  
→ No more forgetting, better scalability



# Follow-up Works

- ESLAM [1]: triplane encoding

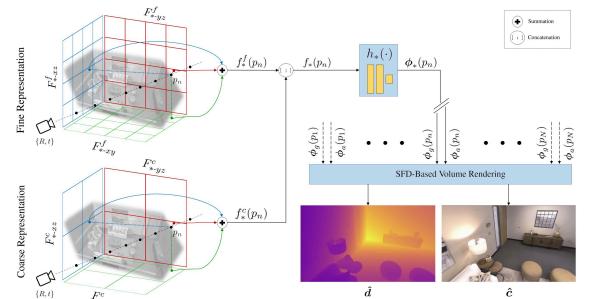


[1] ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields, Johari et al., 2023

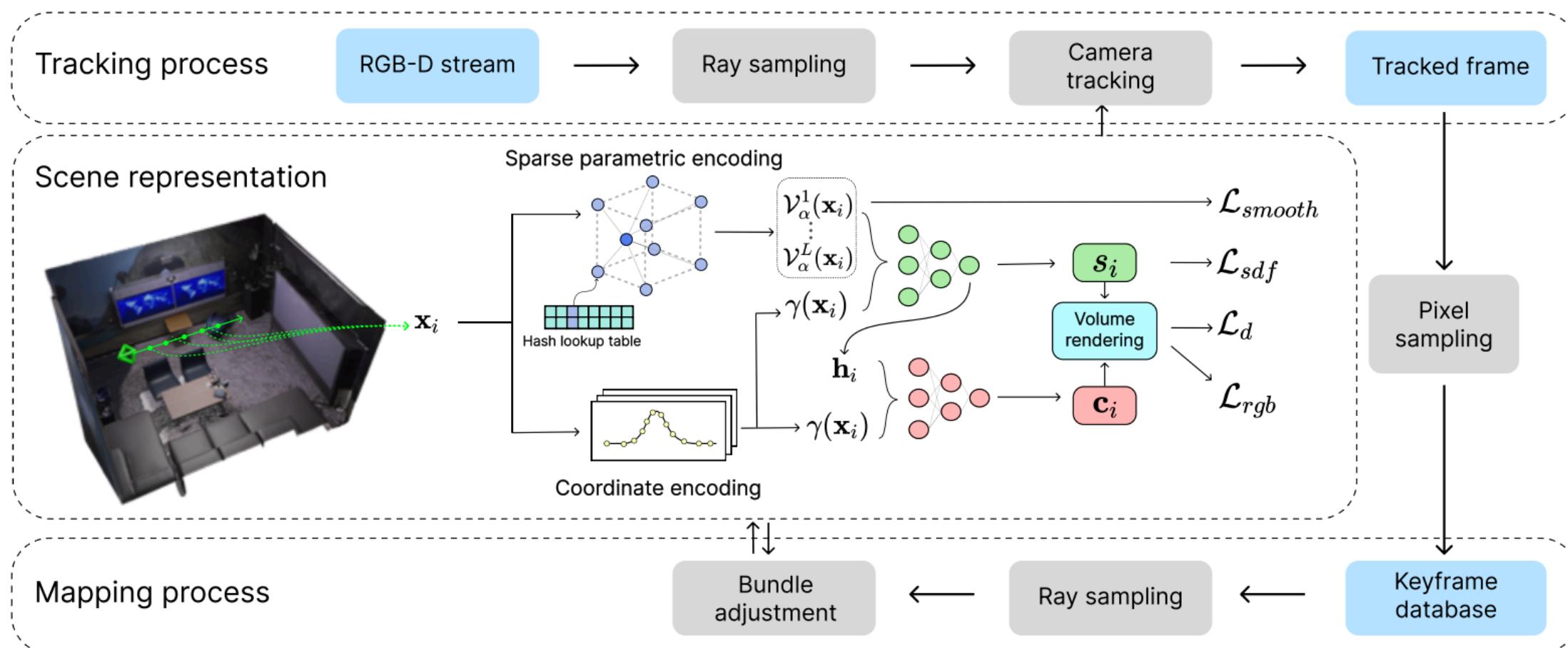
# Neural SLAM

# Follow-up Works

- ESLAM [1]: triplane encoding



- Co-SLAM [2]: joint hash / coordinate encoding

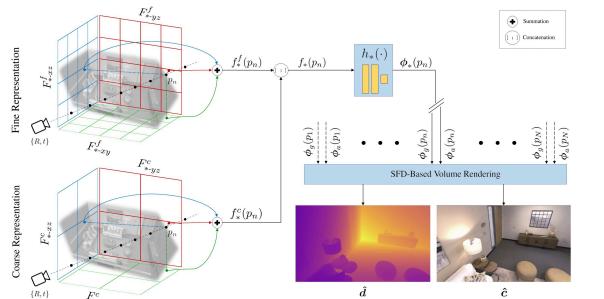


- [1] ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields, Johari et al., 2023  
[2] Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM, Wang et al., 2023

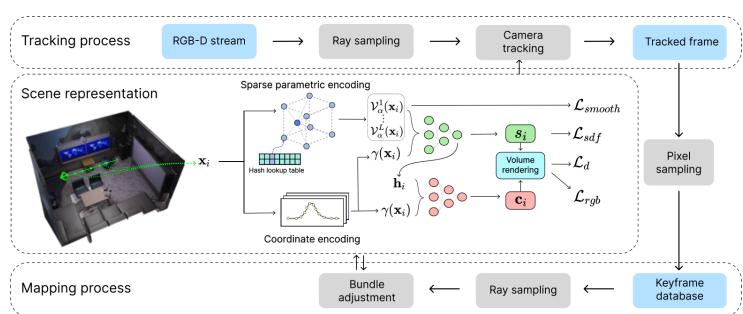
# Neural SLAM

# Follow-up Works

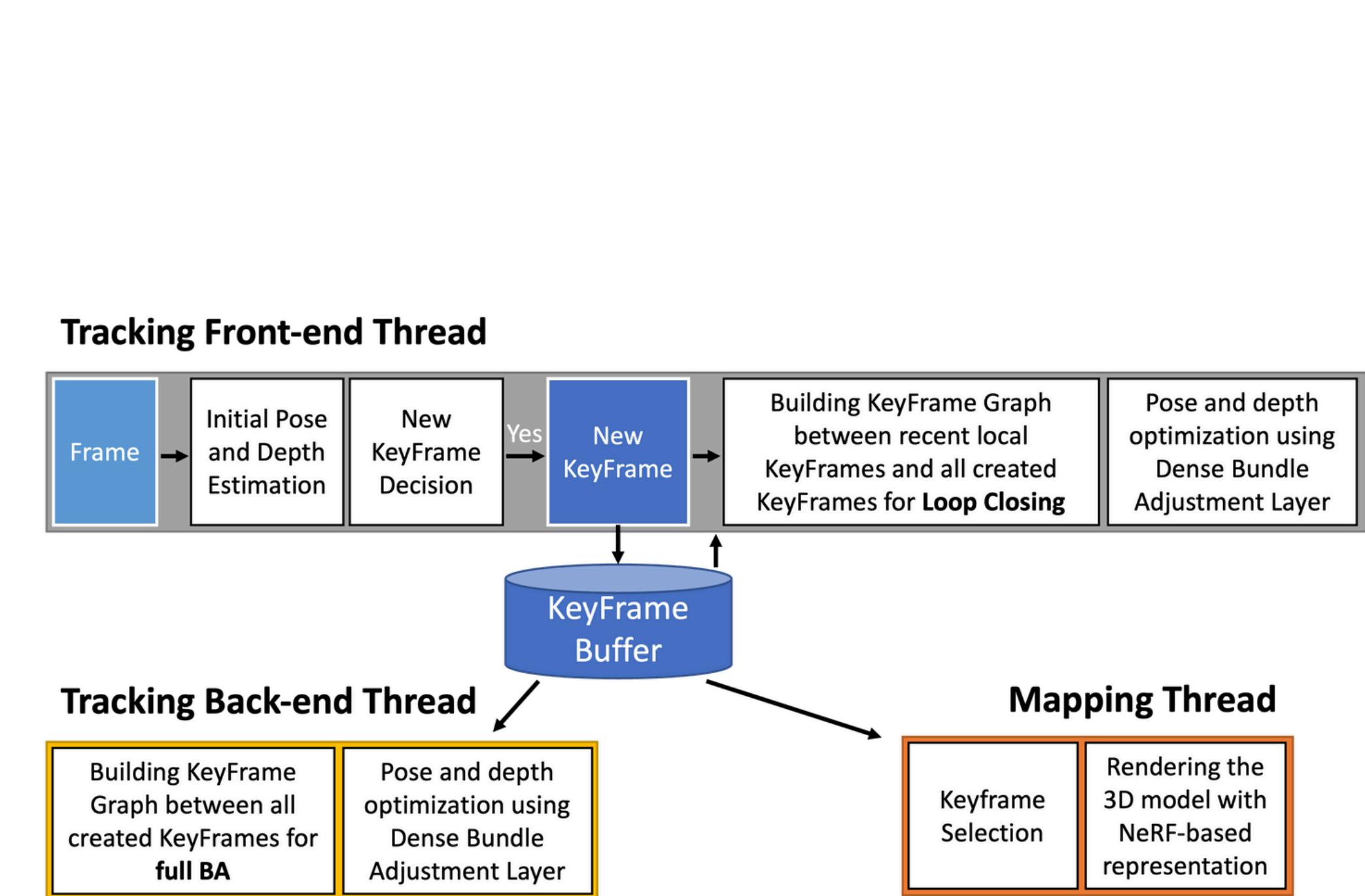
- ESLAM [1]: triplane encoding



- Co-SLAM [2]: joint hash / coordinate encoding



- GO-SLAM [3]: loop closure via reoptimization

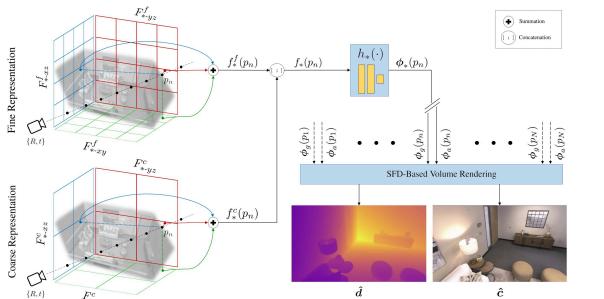


- [1] ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields, Johari et al., 2023  
[2] Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM, Wang et al., 2023  
[3] GO-SLAM: Global Optimization for Consistent 3D Instant Reconstruction, Zhang et al., 2023

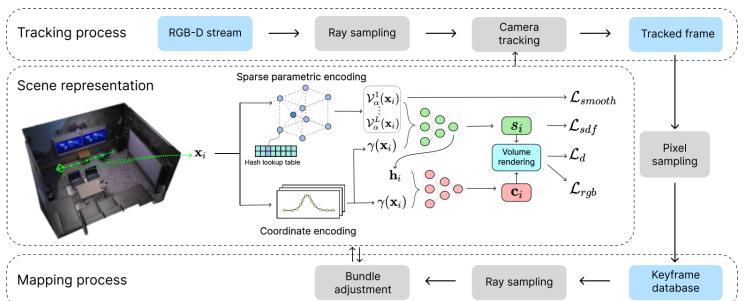
# Neural SLAM

# Follow-up Works

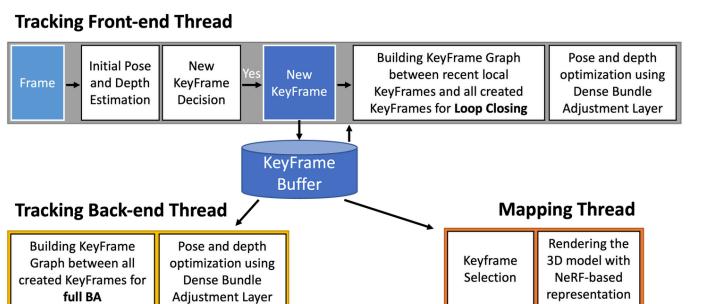
- ESLAM [1]: triplane encoding



- Co-SLAM [2]: joint hash / coordinate encoding



- GO-SLAM [3]: loop closure via reoptimization



- [1] ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields, Johari et al., 2023  
[2] Co-SLAM: Joint Coordinate and Sparse Parametric Encodings for Neural Real-Time SLAM, Wang et al., 2023  
[3] GO-SLAM: Global Optimization for Consistent 3D Instant Reconstruction, Zhang et al., 2023

# Common Limitation: Monolithic Field

- All methods use a single monolithic data structure
- Expensive to optimize neural fields
- Sensor data gets baked into a data structure
  - Similar to classic voxel-based data structures
- Cannot easily deform volumetric scene representation on loop closure
  - GO-SLAM has to reintegrate all keyframes that have moved on loop closure  
→ The larger the loop, the more reoptimization is required

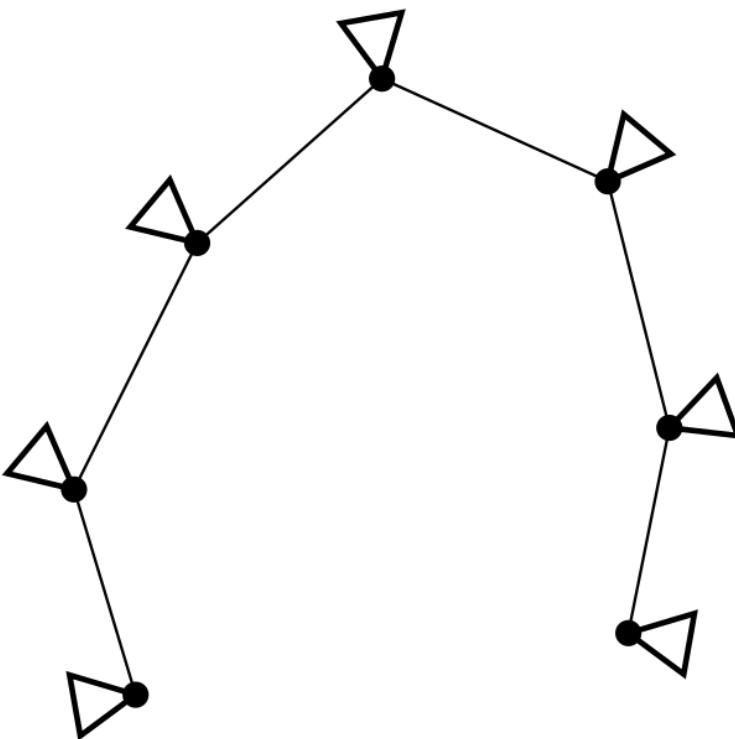


# Neural Graph Mapping

Efficient Loop Closure for Neural SLAM

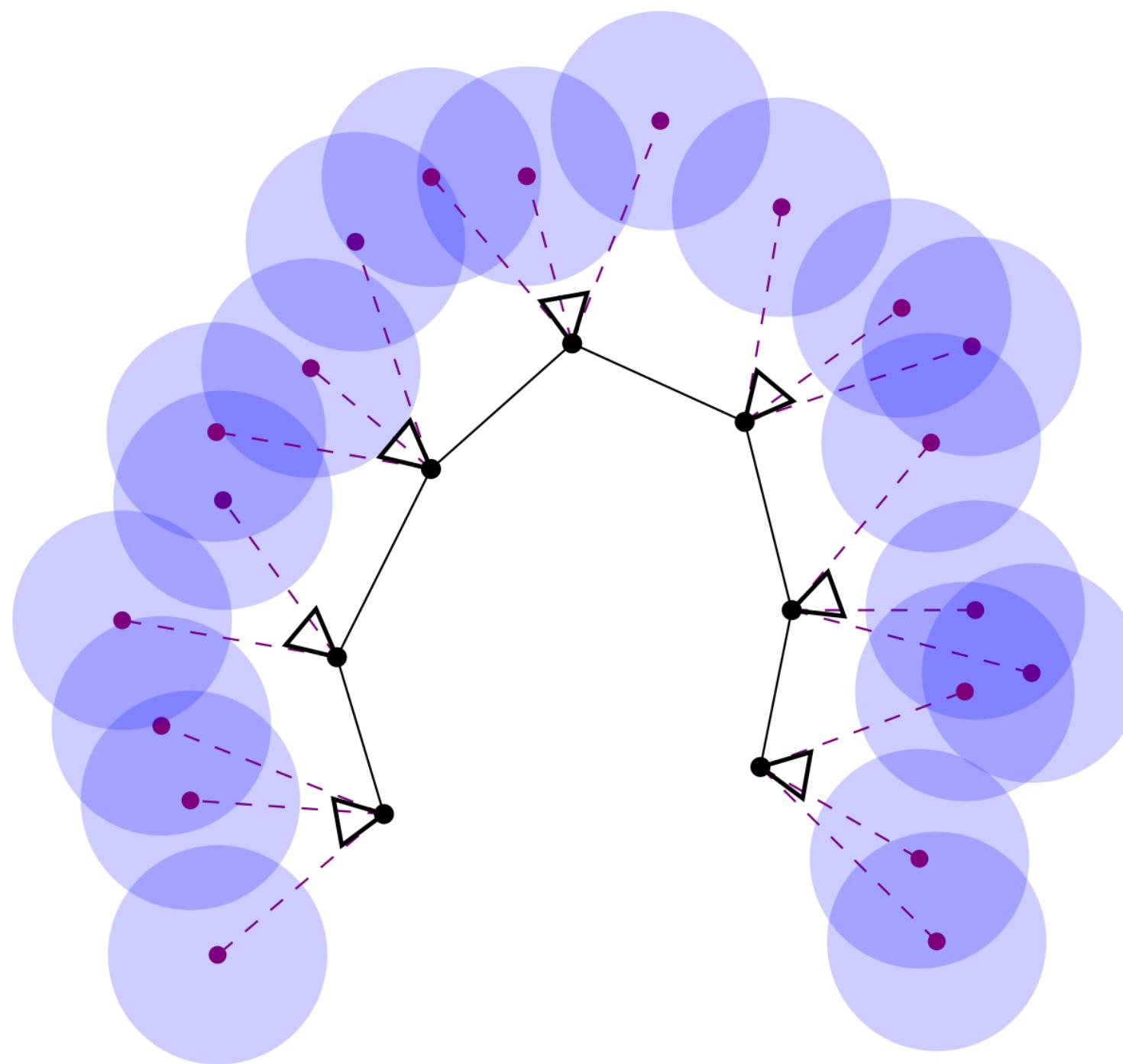
# Neural Graph Mapping

- Idea: attach lightweight neural fields to pose graph of sparse visual SLAM
- Each field captures scene within a fixed radius  $r$



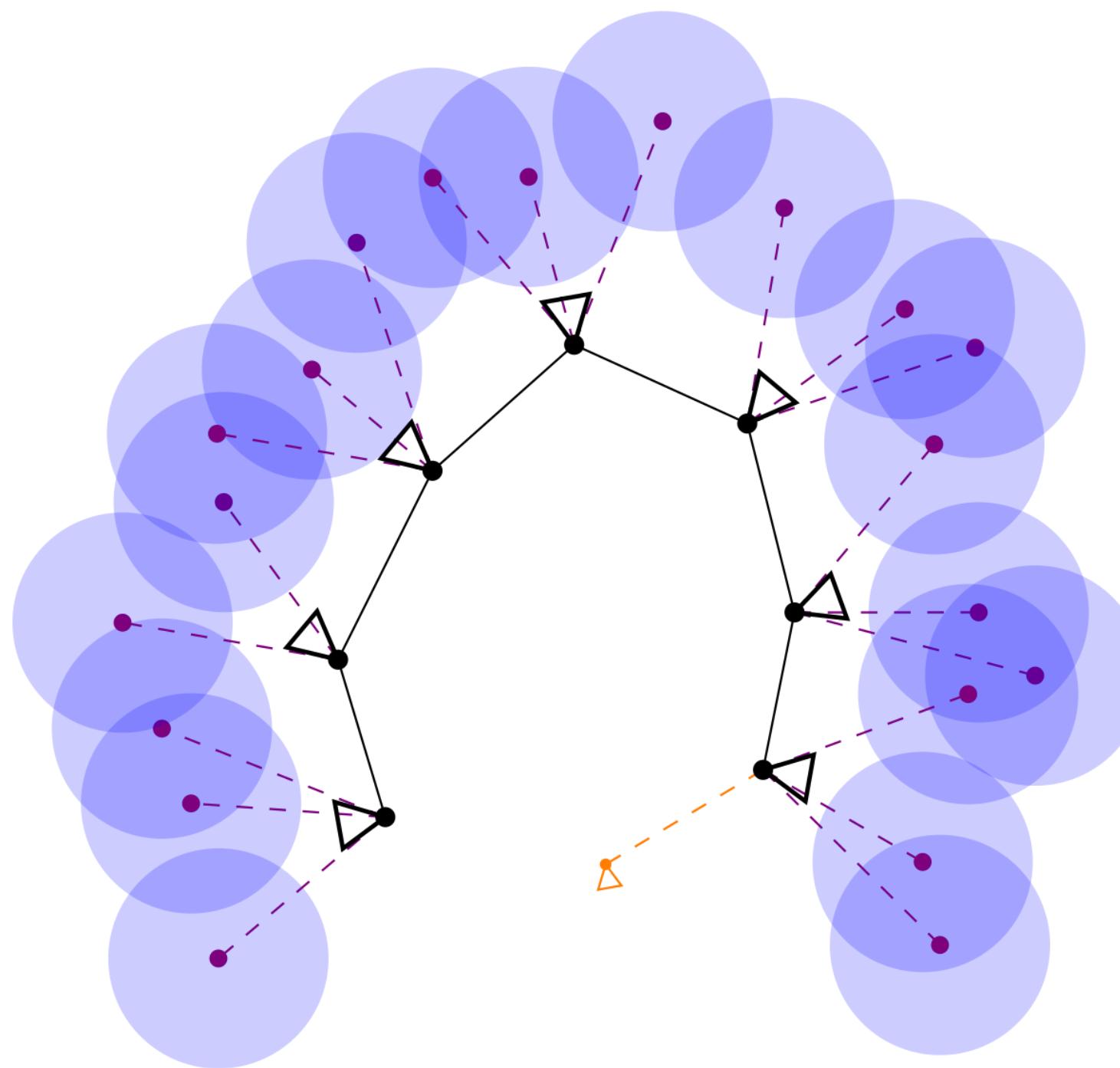
# Neural Graph Mapping

- Idea: attach lightweight neural fields to pose graph of sparse visual SLAM
- Each field captures scene within a fixed radius  $r$



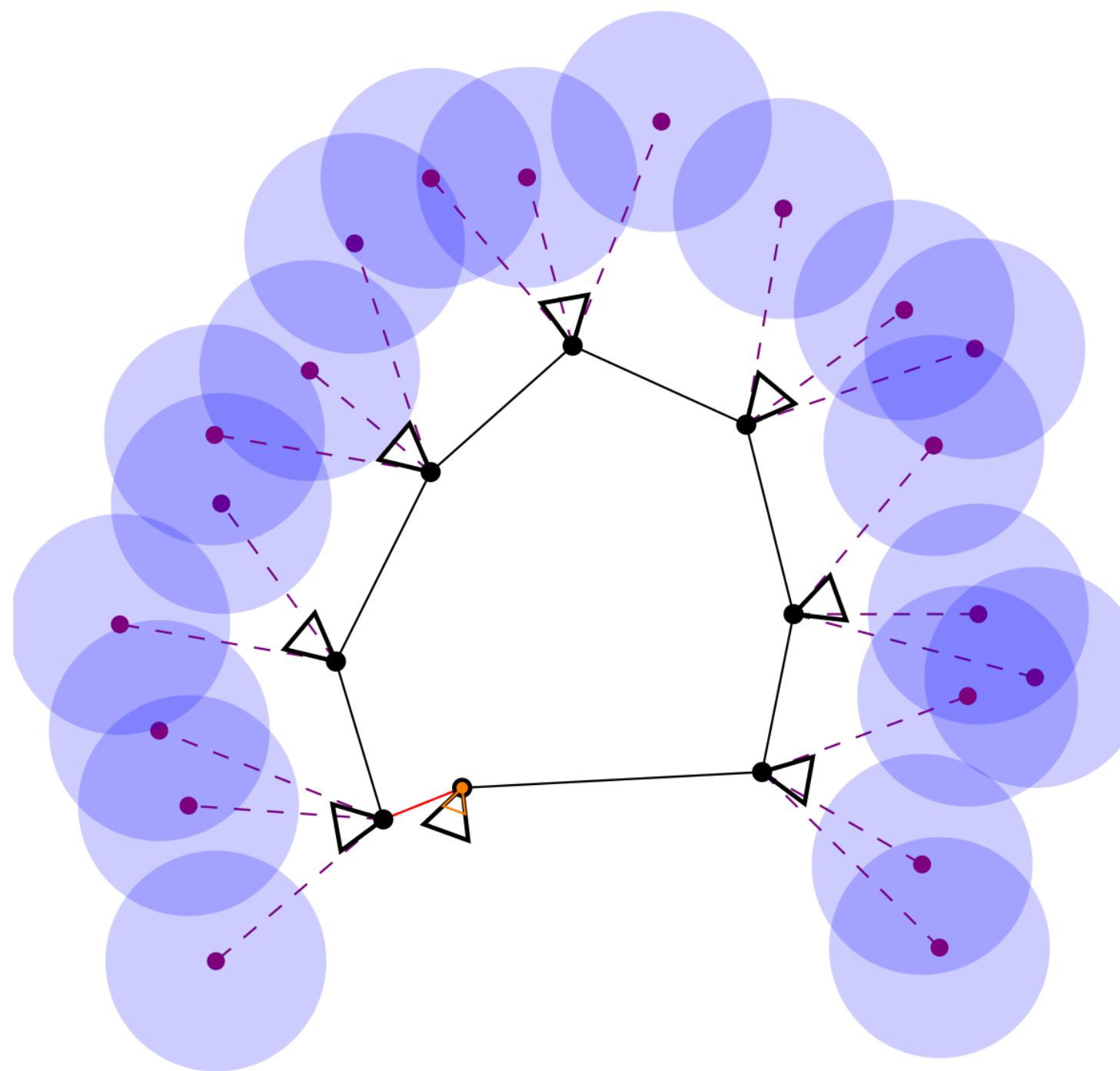
# Neural Graph Mapping

- Idea: attach lightweight neural fields to pose graph of sparse visual SLAM
- Each field captures scene within a fixed radius  $r$



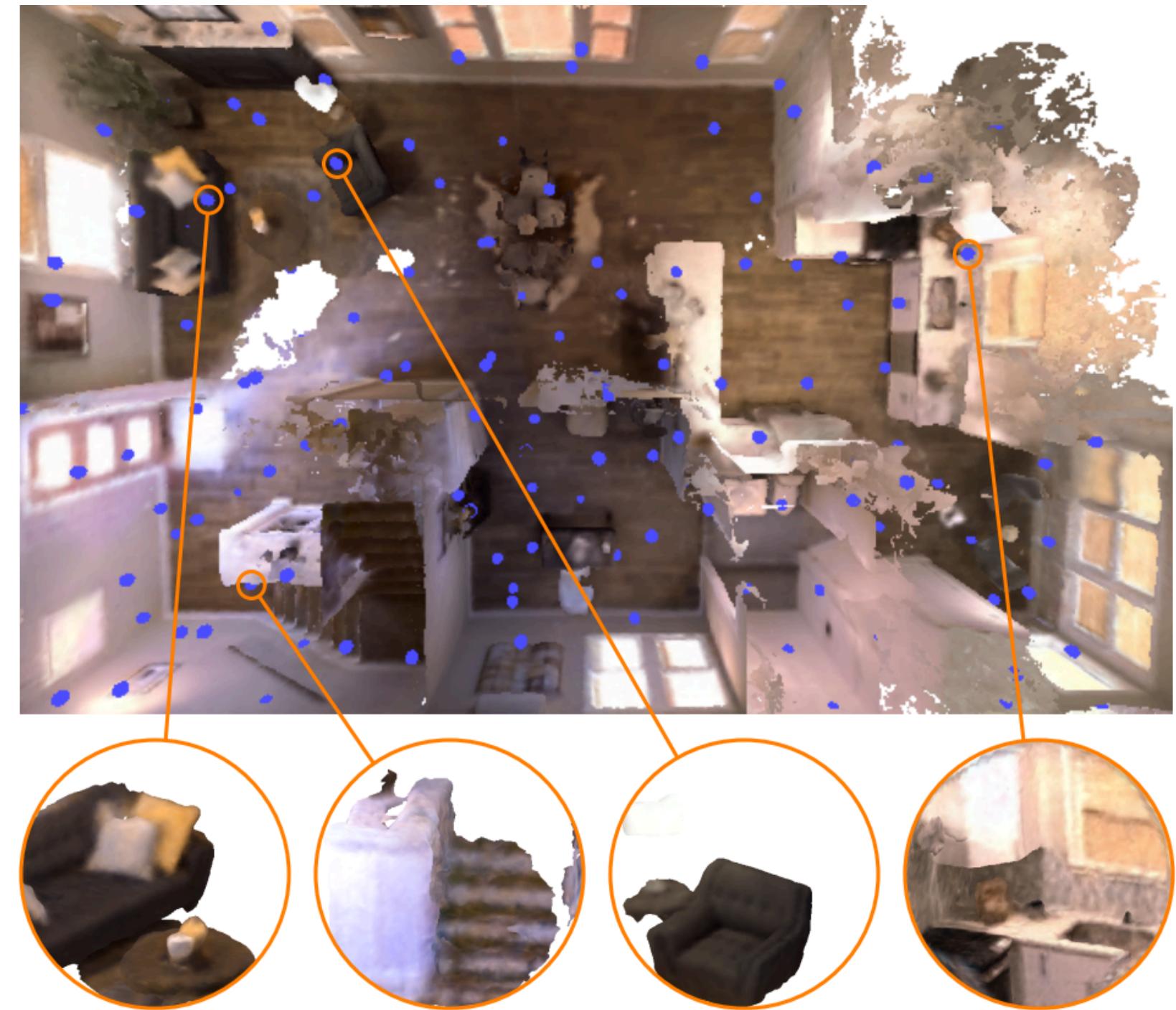
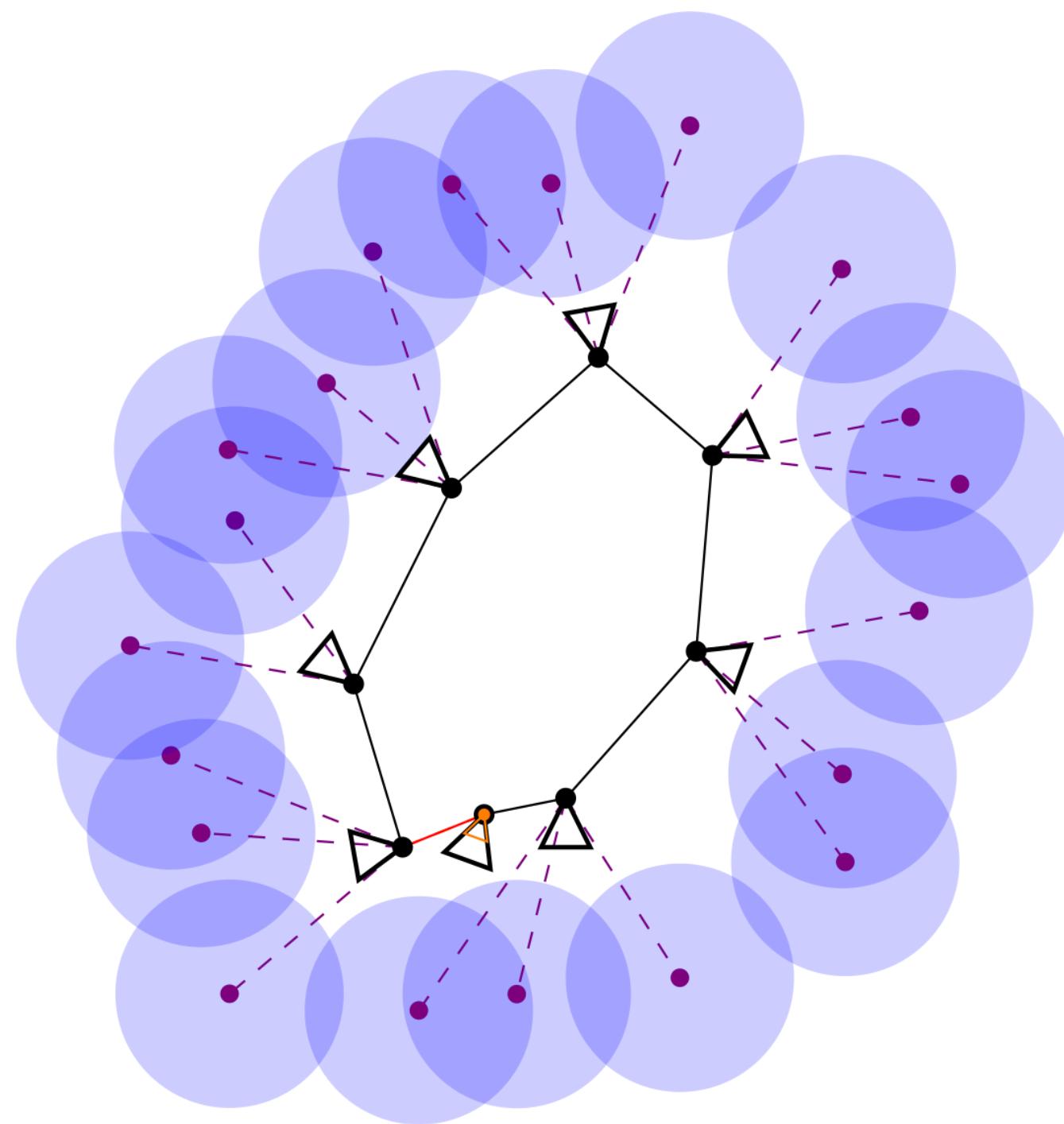
# Neural Graph Mapping

- Idea: attach lightweight neural fields to pose graph of sparse visual SLAM
- Each field captures scene within a fixed radius  $r$



# Neural Graph Mapping

- Idea: attach lightweight neural fields to pose graph of sparse visual SLAM
- Each field captures scene within a fixed radius  $r$



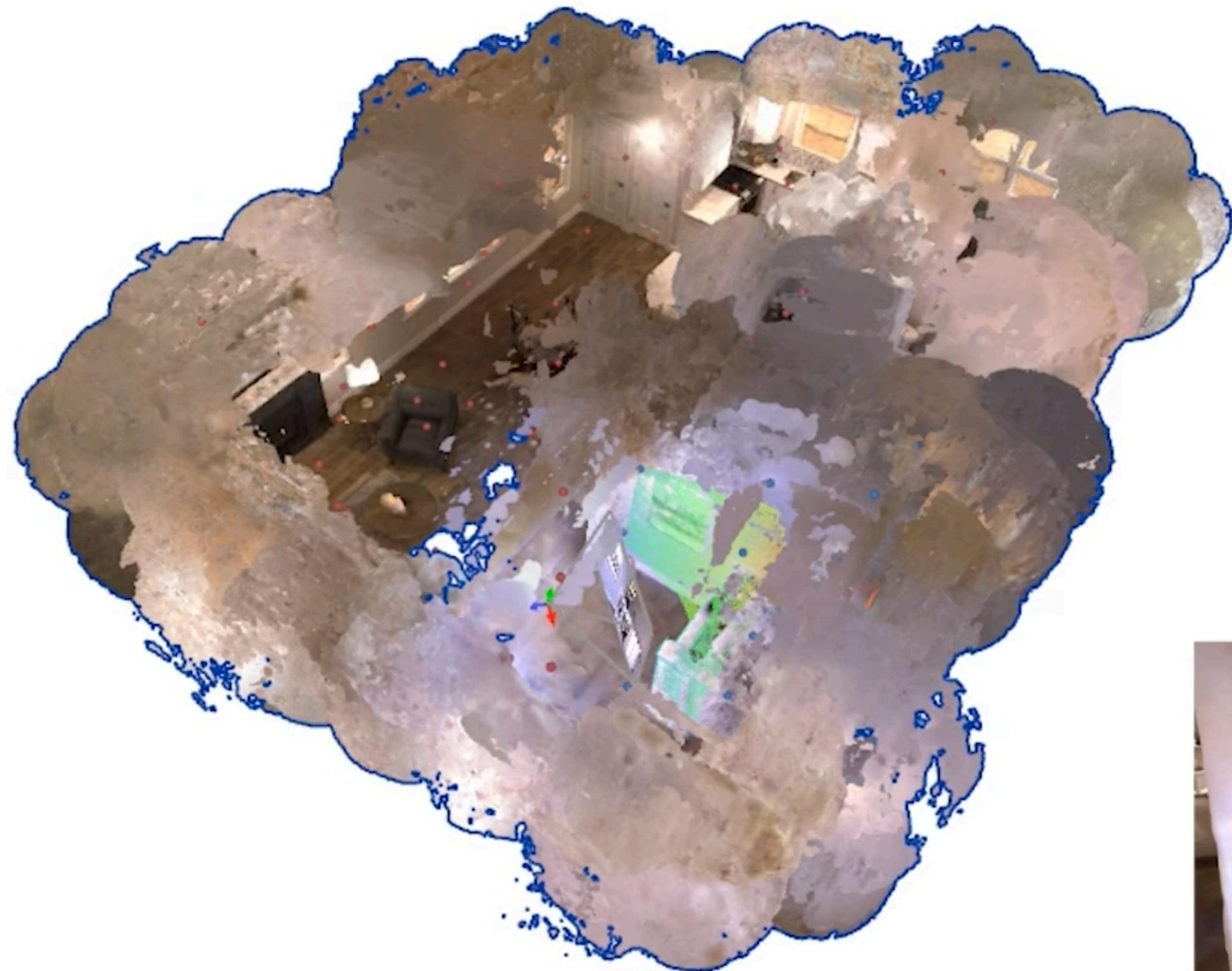
# Neural Graph Mapping

## Example

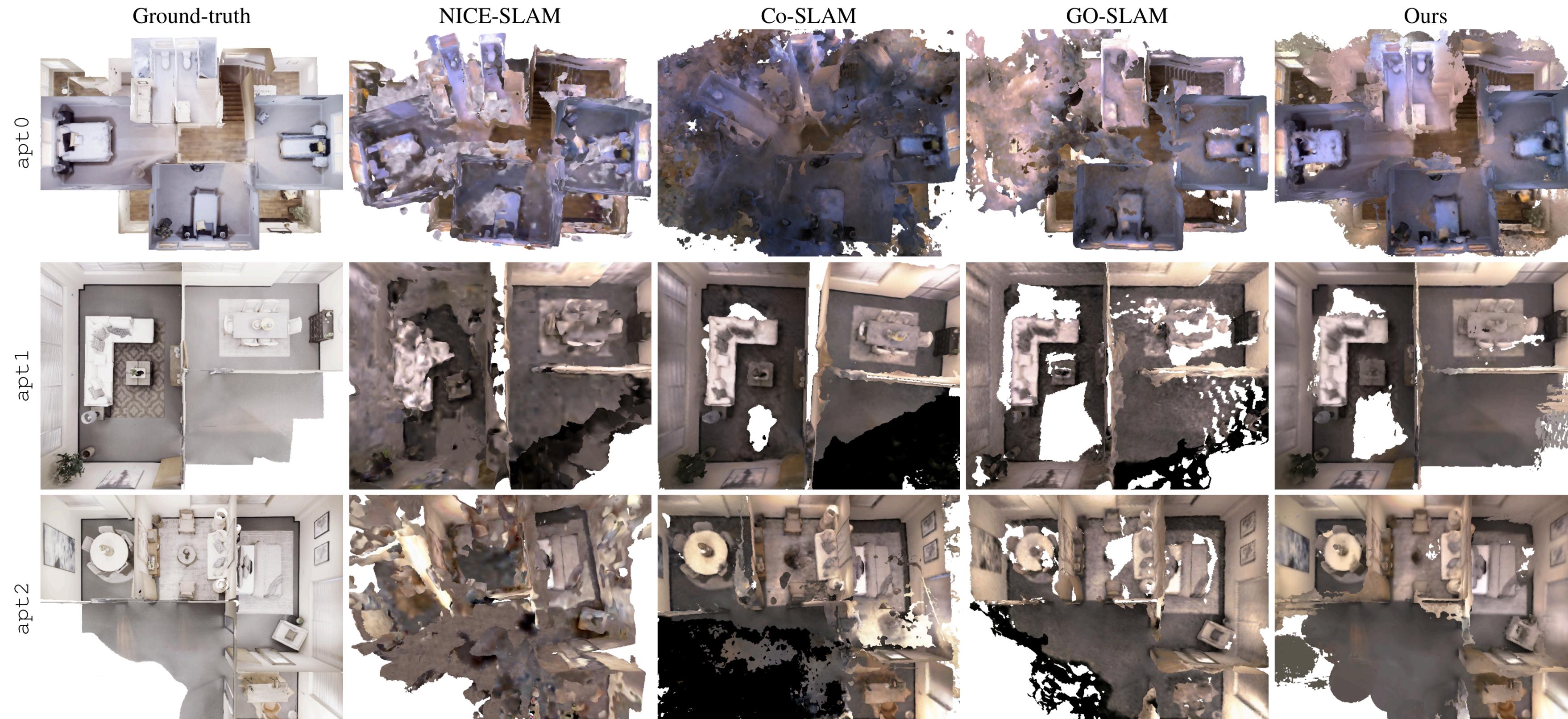


## Neural Graph Mapping

# Another Example



# Comparison to Monolithic Approaches



## Neural Graph Mapping

# Outlook

- Higher memory usage compared to monolithic neural field methods
  - How to reduce memory requirements?
- Neural field-based map not used for SLAM currently
  - How to combine sparse visual SLAM and dense visual SLAM?
- Currently limited to indoor environments with depth data available
  - How to extend to outdoor environments?
- Neural SLAM in general requires lots of resources
  - Can we use meta-learning for few-shot mapping?



# Outlook

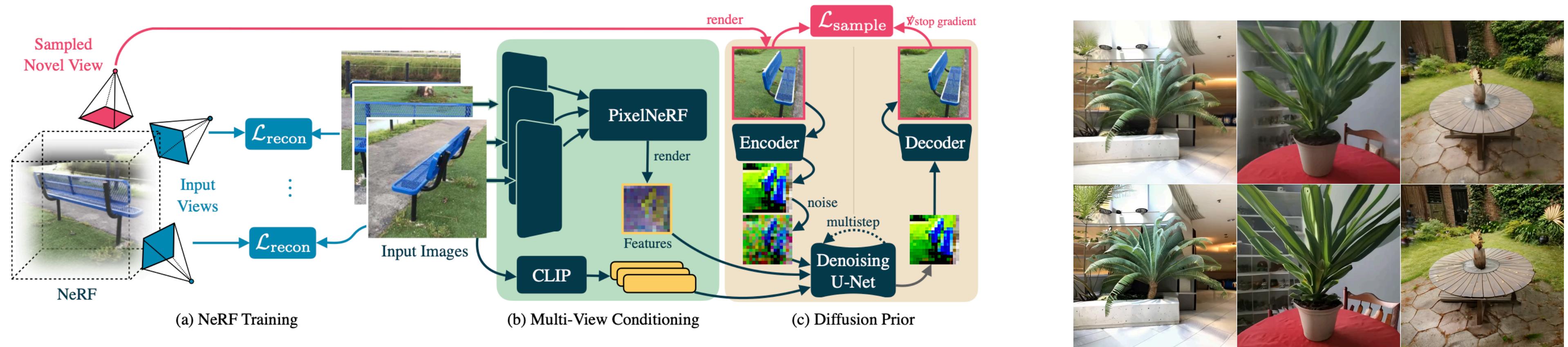
Open Research Problems



# Outlook

## Learning vs Optimization

- Optimization-based approaches inherently limited to what is captured by sensors
  - Sufficient for elaborate sensor setups and human operator
  - However, far from human-level spatial understanding (infilling / etc.)
- Learning approaches might infill from partial observations
- How to combine strong generative model with partial observations?
  - Some work in this direction: ReconFusion



# Outlook

## Gaussian Splatting

- Represent scene by a set of 3D Gaussians instead of differentiable function
- Each Gaussian has
  - Shape: position + covariance
  - Opacity
  - Coefficients of spherical harmonics ( $\rightarrow$  view-dependent color)
- Rasterization instead of volume rendering  
 $\rightarrow$  Real-time (>100 FPS), high-resolution rendering

# Outlook

## Gaussian Splatting

- Represent scene by a set of 3D Gaussians instead of differentiable function
- Each Gaussian has
  - Shape: position + covariance
  - Opacity
  - Coefficients of spherical harmonics ( $\rightarrow$  view-dependent color)
- Rasterization instead of volume rendering  
 $\rightarrow$  Real-time (>100 FPS), high-resolution rendering

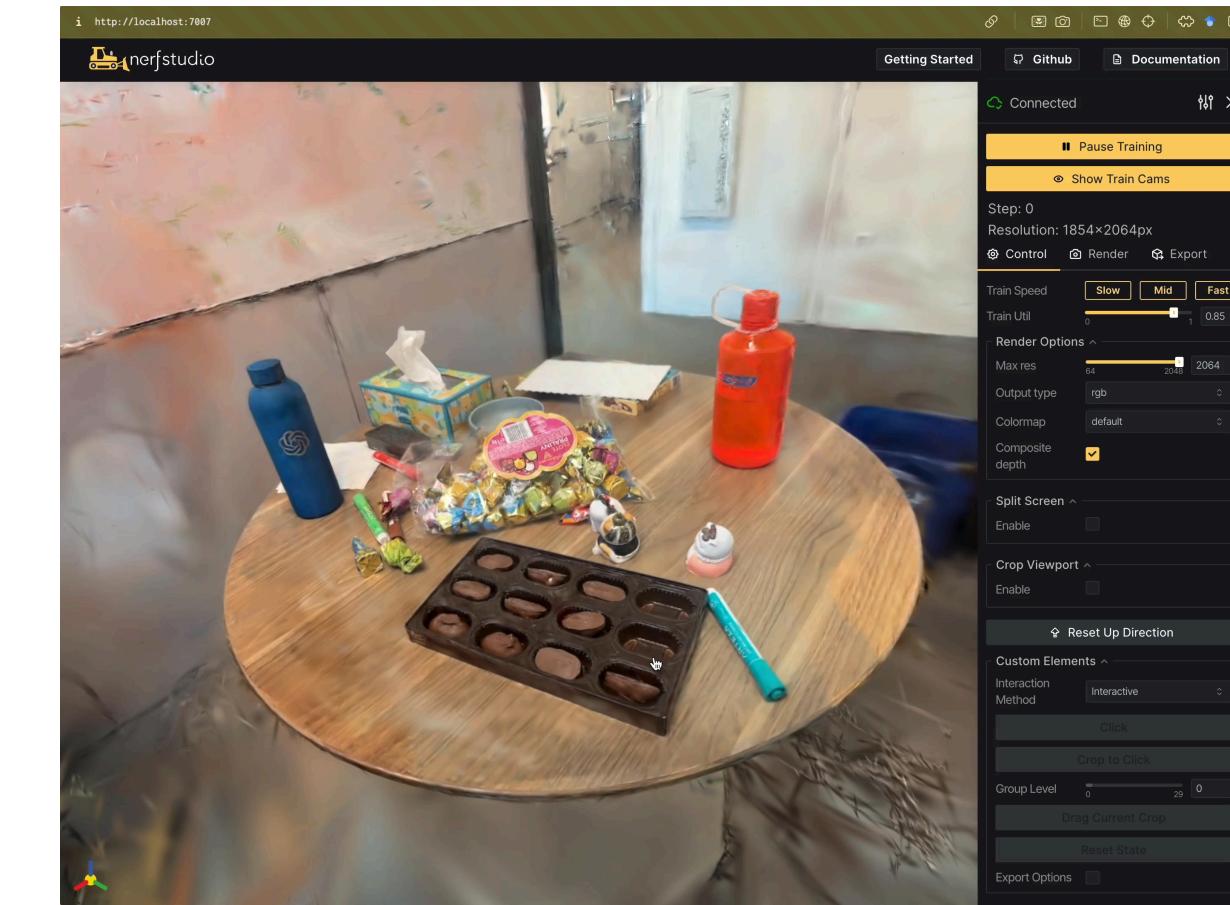


3D Gaussian Splatting for Real-Time Radiance Field Rendering, Kerbl et al., 2023  
<https://lumalabs.ai/featured>

# Outlook

## Object-based / Interactive Map Decomposition

- Scenes learned in this way are not interactive
  - Objects are not separated; and not complete
  - Reflections are baked in
  - Shadows are baked in
- Some early work in this direction: LERF [1], GARField [2]



[1] LERF: Language Embedded Radiance Fields, Kerr et al., 2023

[2] GARField: Group Anything with Radiance Fields, Kim et al., 2024

