3D Scene Understanding

for Real World Applications

Federico Tombari
Google, TUM

ELLIS Summer School - Large Scale AI
Modena, Sept. 18-22 2023

3D Scene Understanding - Ingredients

Reconstruction/Geometry

Planes
Point cloud
3D mesh
Voxel map
Nerf / Implicit rep

Semantics

Segments
Semantic Instance Seg
3D bounding boxes
Panoptic

Layouts/Abstraction

Birds eye view
3D Scene graphs
Scene Captions

3D Scene Understanding - Applications

Augmented Reality

Smartphone

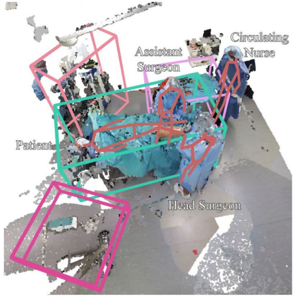
Headsets / Smart glasses

Robotics

Autonomous Driving

Interior design and architecture

Medical



Current Smartphone AR capabilities for Scene Understanding

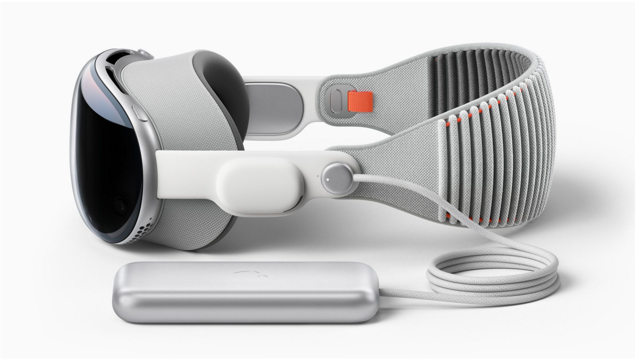
3D Semantic Mesh, ARKit (with Lidar)

Long range depth estimation /
Plane estimation, ARCore (monocular)

Niantic 3D Mapping (monocular)

Currently available features:
3D Planes, Depth prediction, Persistent anchors/objects,
SLAM and 3D Mapping, 3D semantic segmentation,
3D Layouts (with lidar)

Augmented Reality: from headsets to smart glasses



Apple VisionPro

Magic Leap 2

Xreal Air AR glasses

Vuzix Smart Glasses

Microsoft HoloLens 2

Project Aria, Meta

Immersivity / “smartness”

Glass form factor

Apple VisionPro Spatial Audio features (from VisionPro announcement)

Scene understanding for household/service robotics

Incheon Airport Service AIRSTAR Robot

Going from this..

Scene understanding for household/service robotics

Incheon Airport Service AIRSTAR Robot

TRI home helping robot

Going from this.. ..to this

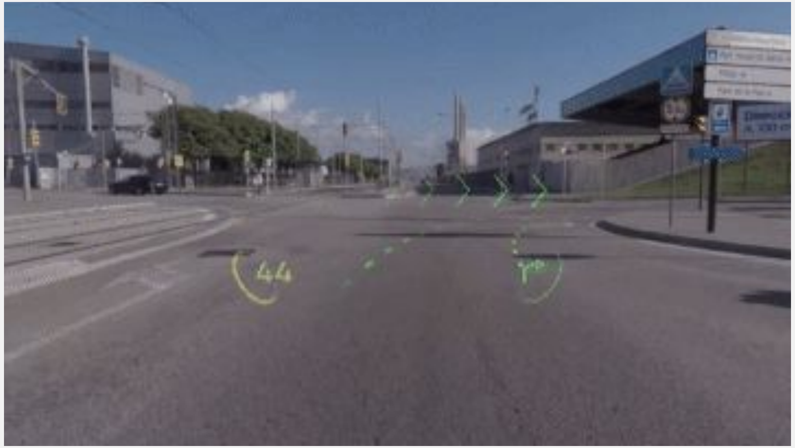
Scene understanding for Autonomous Driving

Waymo Car in San Francisco

MobilEye Car in New York City

AR Head-Up Display — Augmented windshield

wayray.com



AR for Automotive and Navigation

AR and gaze control
(from BMW)

AR and navigation
(Blue Vision)

Agenda

Reconstruction/Geometry

Planes
Point cloud
3D mesh
Voxel map
1. Nerf / Implicit rep

2. Open Set Semantics

Segments
Semantic Instance Seg
3D bounding boxes
Panoptic

Layouts/Abstraction

Birds eye view
3. 3D Scene graphs
Scene Captions

..with the lens of real world applications

Tutorial and project

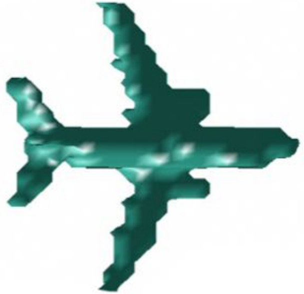
(1/3)

Nerf for real applications

Explicit 3D data representation

Point Cloud
- unordered list of 3D coordinates
- sparse, no topology
- can handle **full 3D**

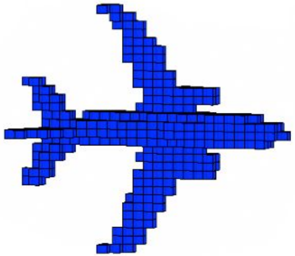
3D Mesh
- collection of 3D vertices and faces
- sparse, with topology
- can handle **full 3D**



Sparse

Organized

Voxel map
- discretized 3D coordinates on a regular grid
- organized, no topology
- can handle **full 3D**

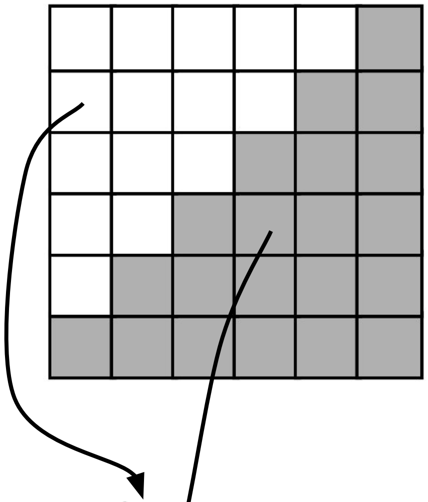


Range (depth) map
- 1-channel image encoding distances
- organized, no topology
- only **2.5D views**

Occupancy Field and Signed Distance Field

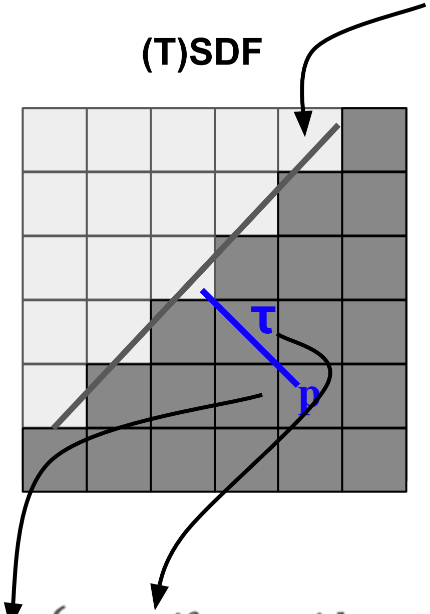
Surface implicitly defined as f(x) = 0

OF



$$f(p) = \begin{cases} 0, & \text{if p outside} \\ 1, & \text{otherwise} \end{cases}$$

(T)SDF



$$f(p) = \begin{cases} -\tau, & \text{if p outside} \\ \tau, & \text{otherwise} \end{cases}$$

- Special cases of a voxel map where each voxel stores:
  - OF: volume occupancy
  - SDF: distance to the nearest surface
- Common variations: Truncated SDF (TSDF), Unsigned DF (UDF)

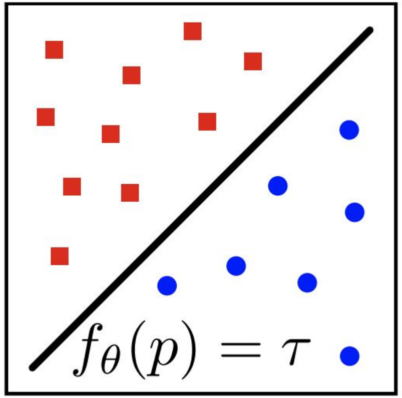
3D Implicit representations

Learn a **function f** via a non-linear classifier whose decision boundary is the desired 3D surface

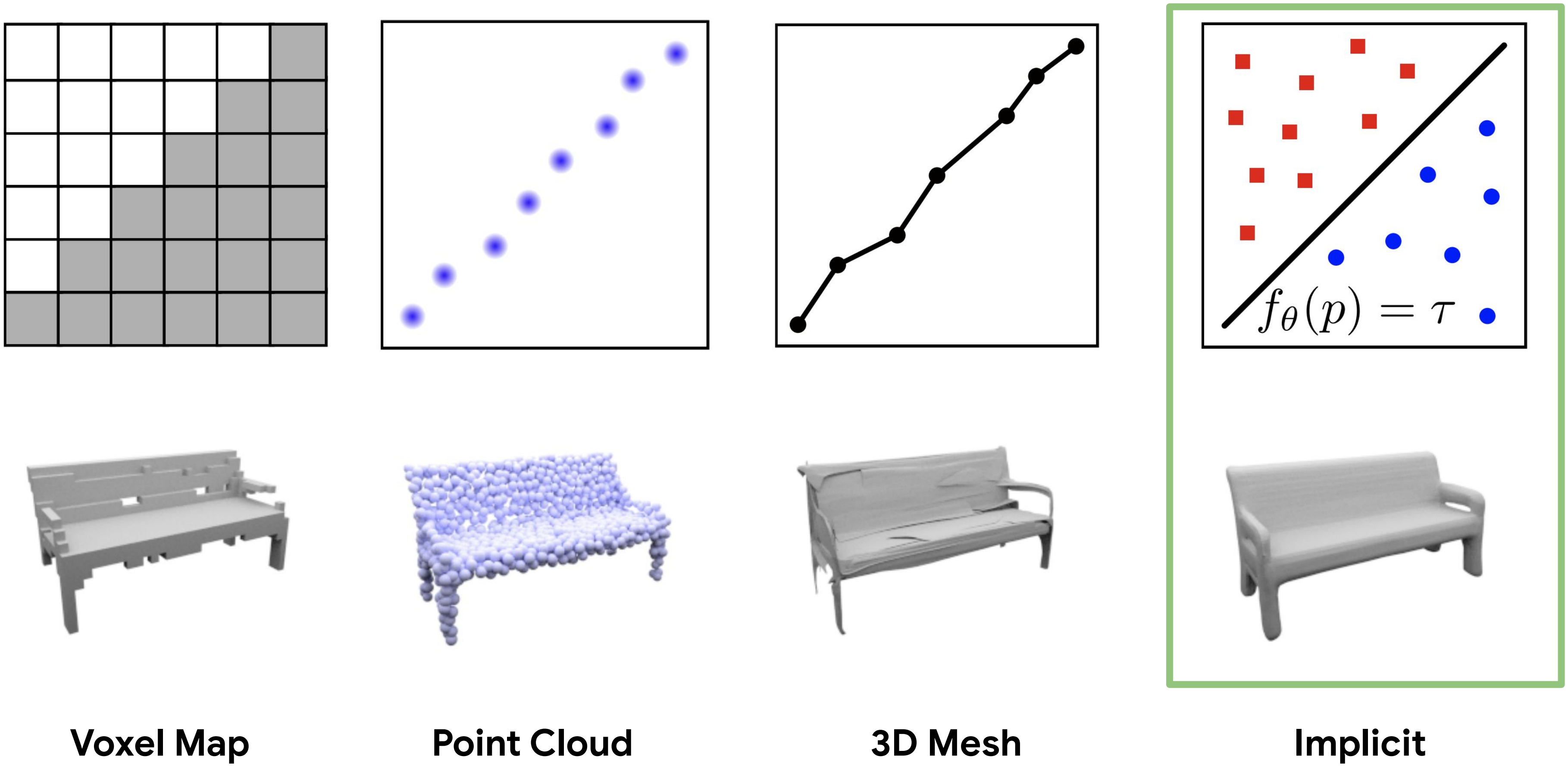
The function f approximates an **Occupancy Field [1] or a Signed Distance Field [2]**

Why?

- No discretization
- Arbitrary topology & resolution
- Low memory footprint

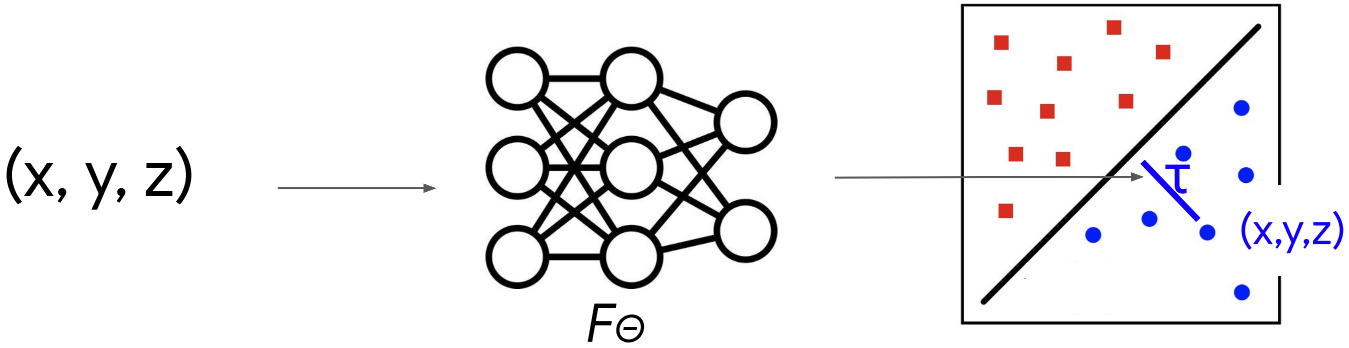


[1] L Mescheder et al, Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019
[2] JJ Park et al, DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation, CVPR 2019



L Mescheder et al, Occupancy Networks: Learning 3D Reconstruction in Function Space, CVPR 2019

Summary of 3D implicit representations



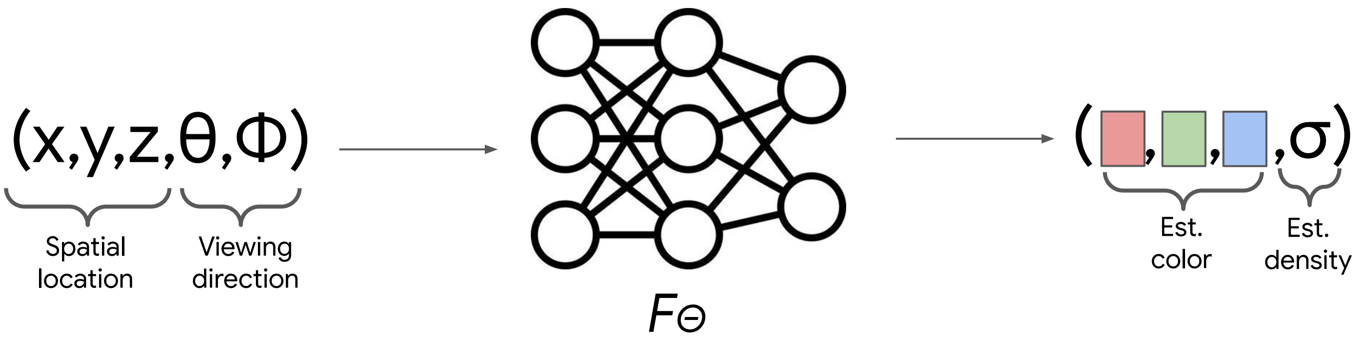
3D implicit representations are a type of data representation that uses a function to map from a domain to a range. The function is typically learned by overfitting a neural network.

They tend to be more compact and flexible than traditional explicit representations

Applications

- Data storage / compression
- 3D classification / segmentation
- 3D reconstruction from single view
- 3D generation

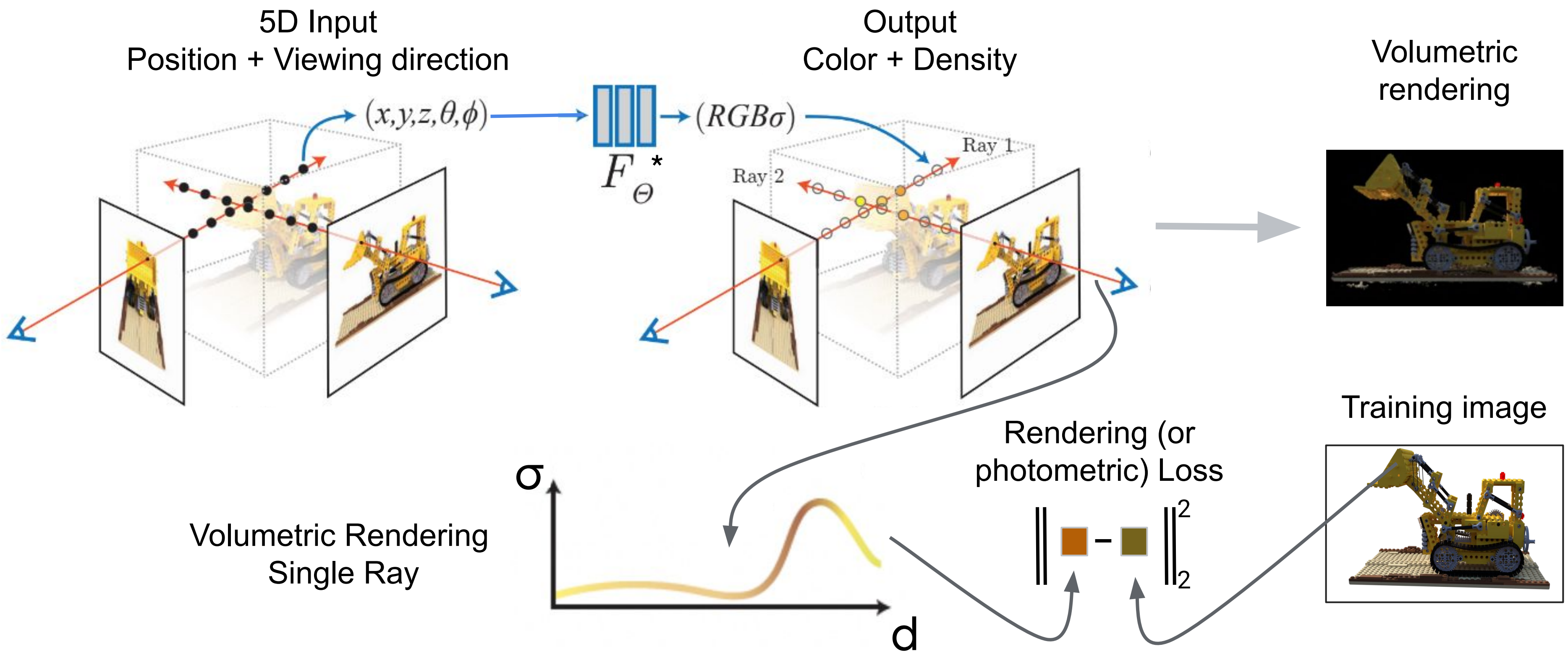
Implicit representations: from occupancy to radiance fields



- The network is a simple ReLU MLP that maps from location/view direction to color/density
- Density σ describes how solid/transparent a 3D point is (can model, e.g., fog)
- Conditioning on view direction allows for modeling **view-dependent effects**

Input: posed images (no explicit 3D geometry or depth)
Task: Novel View Synthesis

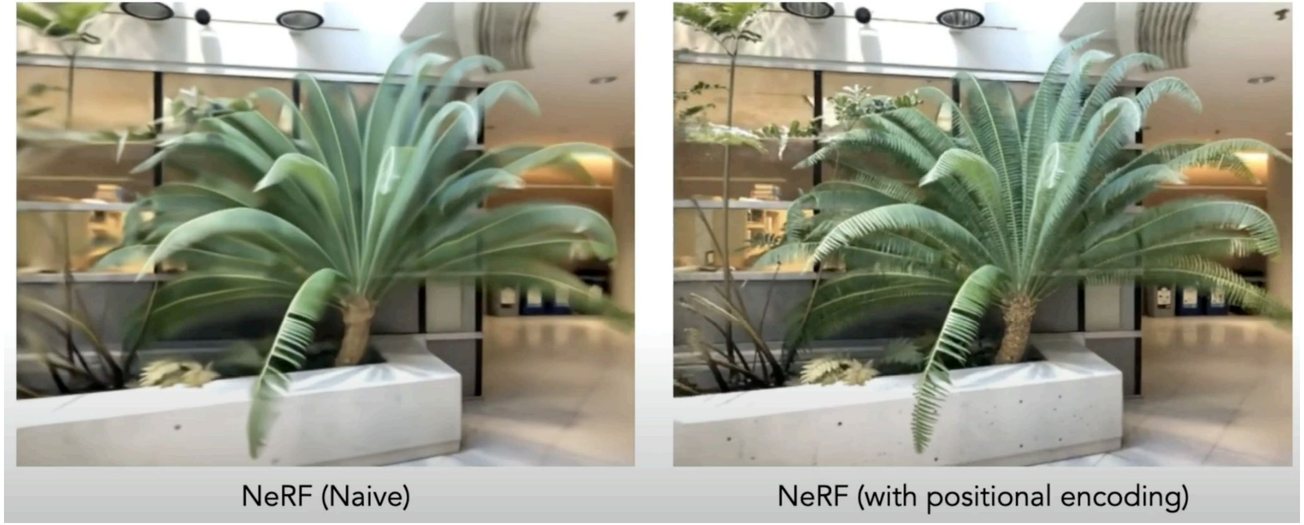
Training of NeRFs



*This is done in practice with 2 MLPs: one non-view dependant that regresses the density, the other that takes also the viewing direction and computes the RGB (since density should not be view-dependant!)

Images from Mildenhall et al. NeRF. Commun. ACM.

Positional encoding and Fourier features



$F_\Theta(x, y, z, \theta, \Phi)$ $F_\Theta(\gamma(x, y, z), \gamma(\theta, \Phi))$

Adding positional encodings to input coordinates (point and direction) helps recover fine details

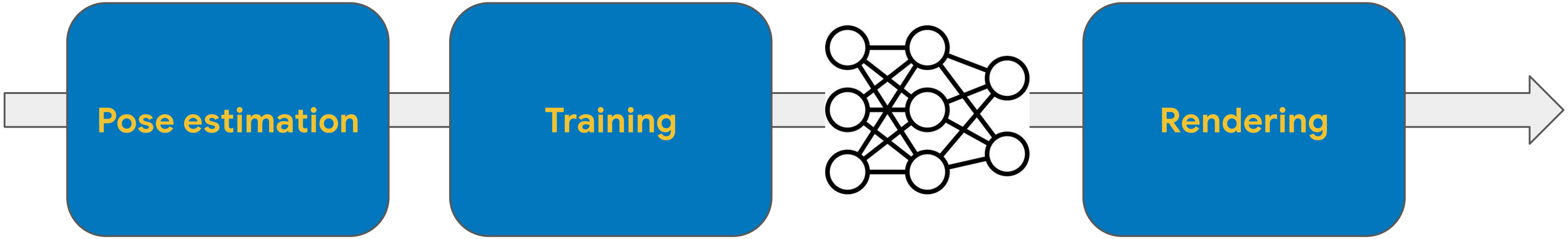
Tancik et al.: Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. NeurIPS, 2020

A simple and powerful representation..

.. still limited for many real world applications

Nerf main limitations for real world applications

- **Pose estimation** is a critical step. Nerf requires dense pose coverage with accurate pose estimation. **Noise and sparsity** highly affect quality.
- **Rendering in real time is still a problem**, especially on embedded/mobile settings (e.g. smartphone)



SPARF: Better poses under sparse settings

Nerfmeshing: real-time mobile rendering of neural meshes

SPARF: Neural Radiance Fields from Sparse and Noisy Poses

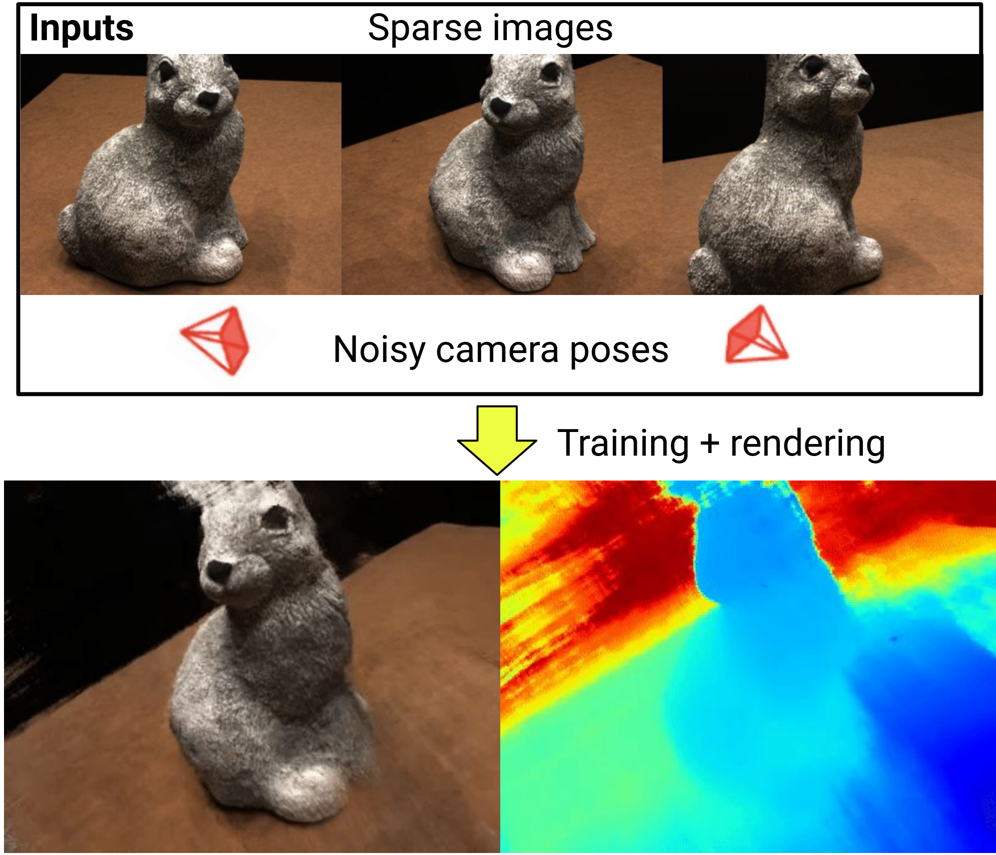
CVPR 2023 - Highlight

Prune Truong, Marie Julie Rakotosaona, Fabian Manhardt, Federico Tombari

Website: https://prunetruong.com/sparf.github.io/
Code: https://github.com/google-research/sparf

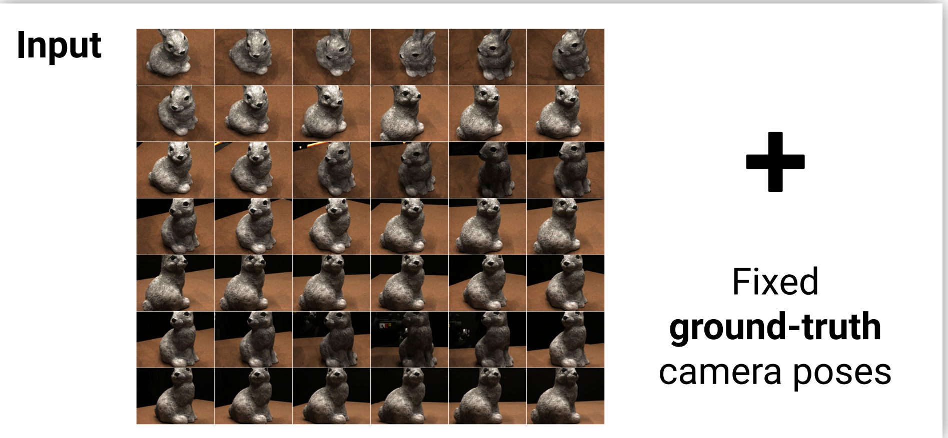
Novel-view synthesis given few images and noisy pose

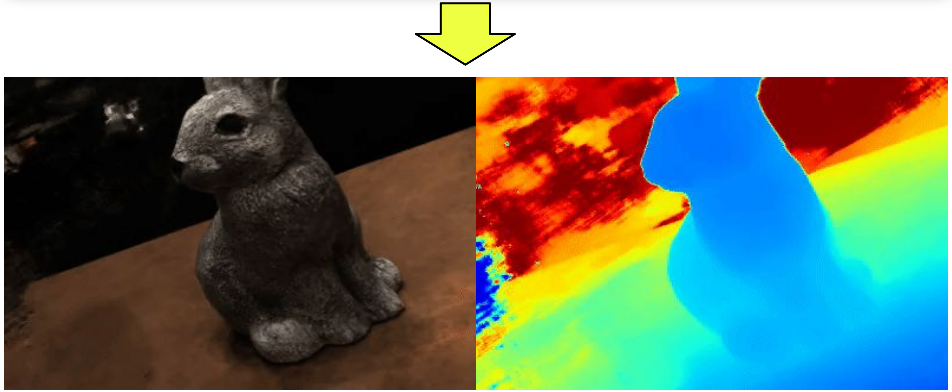
- Goal: Novel-view synthesis via Nerf with access to only **few wide-baseline images** (as low as 2 or 3), with **noisy camera poses**.
- Why? This is a realistic scenario, in e.g. robotics, AR/VR or autonomous driving



Our approach SPARF

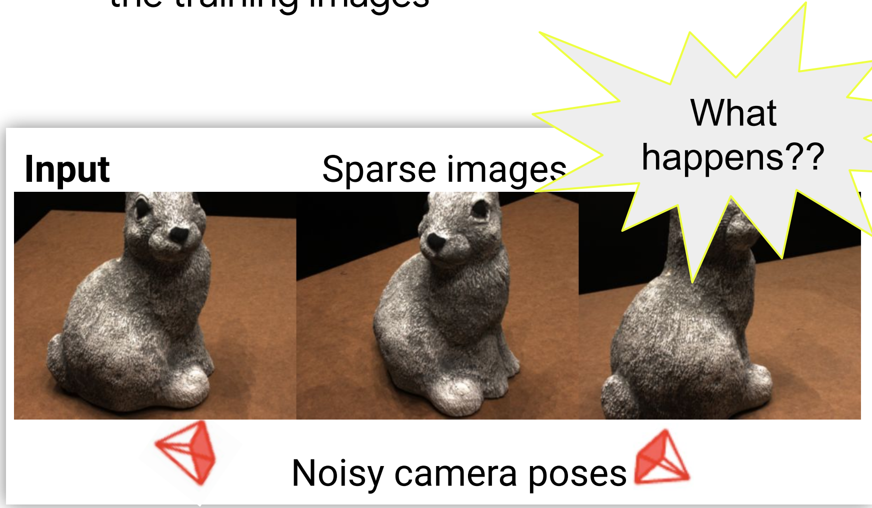
NeRFs: Challenge in the sparse-view setting





For realistic novel-view renderings, it requires:

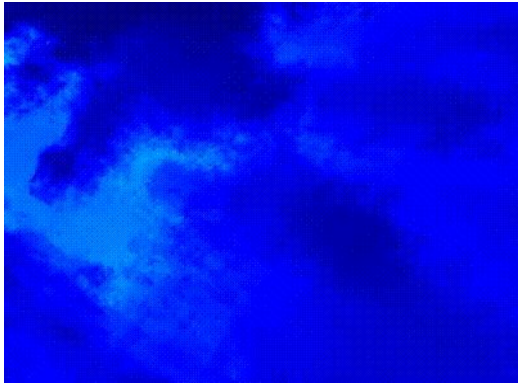
- Lots of training images (dense coverage of the 3d space)
- Known and accurate camera poses for the training images



NeRFs: Challenge in the sparse-view setting

Input

\+ Fixed **ground-truth** camera poses



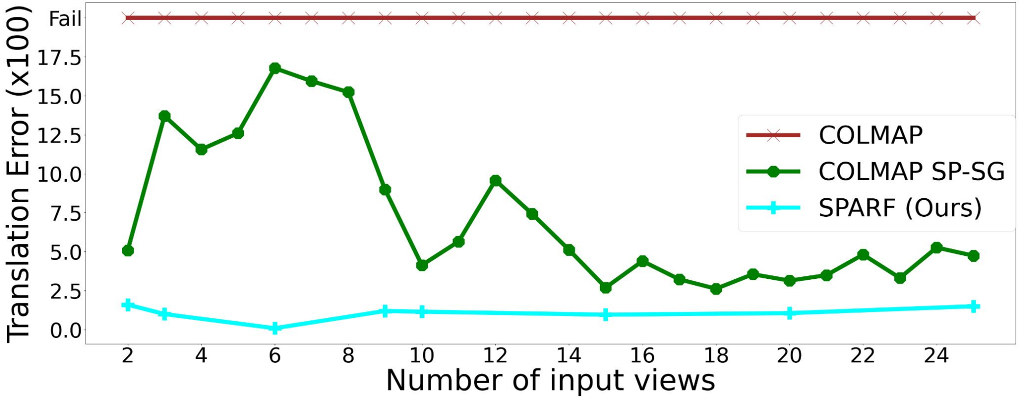
Sparse input views with **fixed ground-truth poses**:

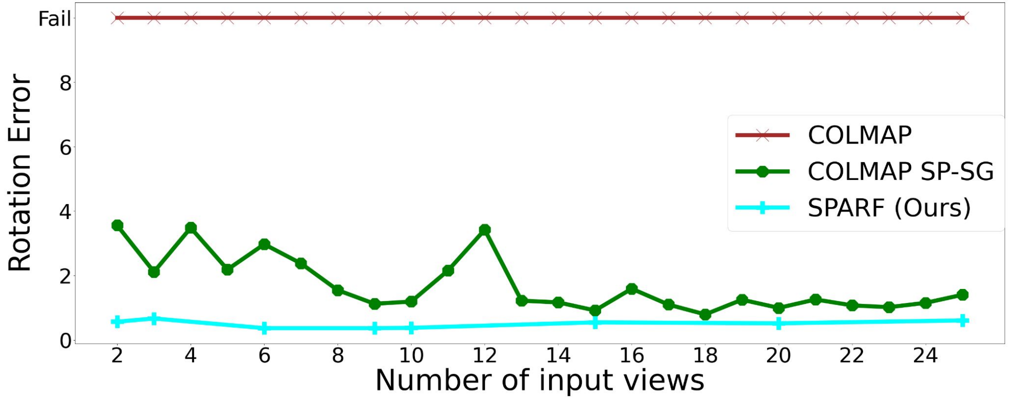
⇒ Overfit to the training views
- Degenerate geometry
- Bad novel view rendering

NeRFs: Challenge in the sparse-view setting

How do we get the poses?

- The standard is to use COLMAP, a structure-from-motion approach
- On few, wide-baseline images, COLMAP will most likely fail
- Even when using better matching, the performance of COLMAP degrades as the number of views decrease
- Pose errors will lead to errors in the learnt scene and therefore in the renderings





Contribution

We propose **Sparse Pose Adjusting Radiance Field** (SPARF), a joint pose-NeRF training strategy.

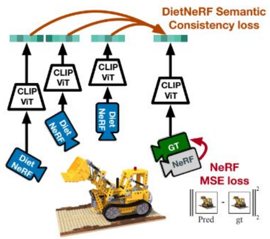
Our approach produces realistic novel-view renderings given

- only few wide-baseline input images (as low as 2 or 3)
- with noisy camera poses.

Related works on NeRF from sparse images

Works which add regularization losses to the NeRF optimization

DietNeRF (ICCV 2021)



Minimizes the CLIP embedding differences between rendered and training images

RegNeRF (CVPR 2022)

Depth-smoothness loss on rendered images,

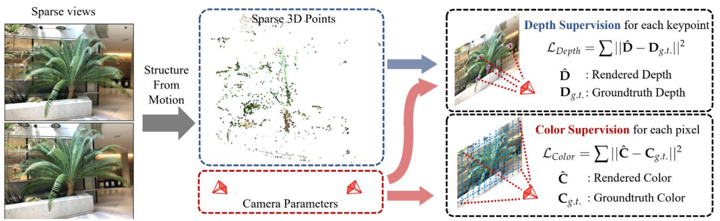
appearance regularization on rendered images

InfoNeRF (CVPR 2022)

Ray entropy regularization to prevent overfitting

Works which add geometric constraint to the NeRF optimization

DS-NeRF



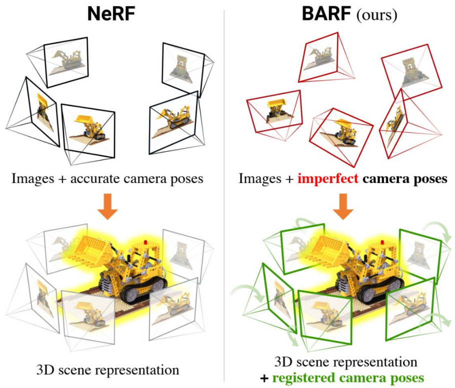
Supervises the rendered depth with sparse depth obtained from triangulation in COLMAP.

But runs COLMAP with ground-truth poses!

They all assume fixed ground-truth poses. ⇒ this is unrealistic!

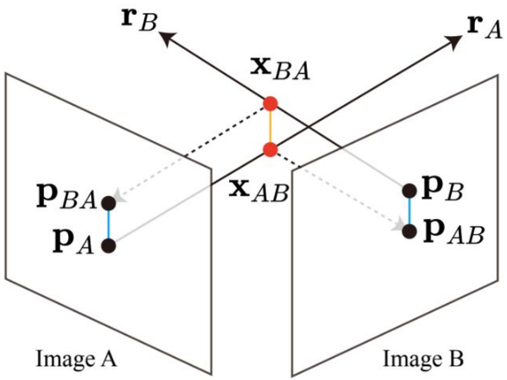
Related works on NeRF from noisy poses

BARF (ICCV 2021) + Follow-ups



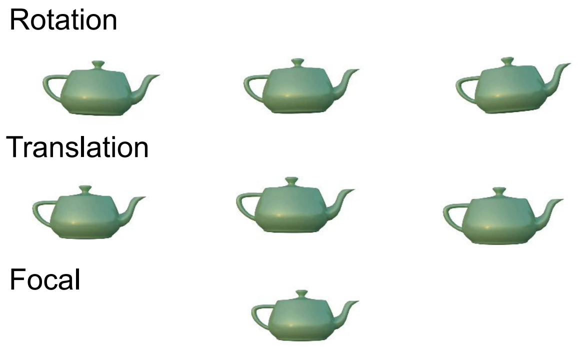
- Proposes to jointly finetune the camera poses with the NeRF
- Follows a coarse to fine strategy to avoid too fast overfitting to a suboptimal solution

SCNeRF (ICCV 2021)



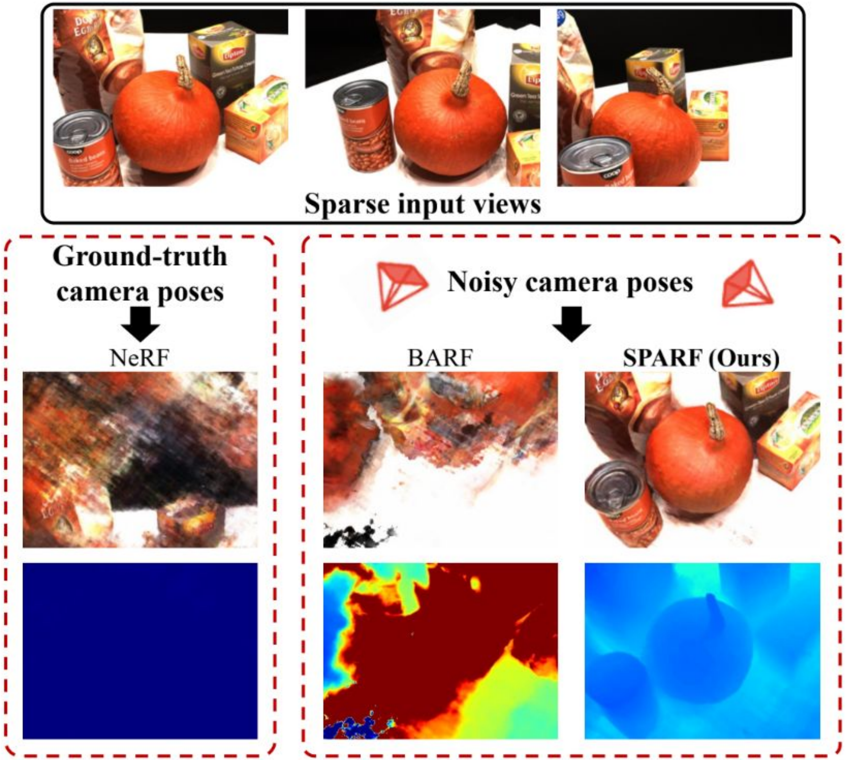
- Proposes the *Projected Ray Distance loss* ⇒ Computes the intersection of the corresponding 2 rays, and measures the re-projection error.
- Proposed loss has a geometric basis but it impacts only the learnt poses

CamP
(SIGGRAPH Asia 2023)



- CamP preconditions camera optimization in camera-optimizing Neural Radiance Field
- Proposes using a proxy problem to compute a whitening transform that eliminates the correlation between camera parameters and normalizes their effects

Our approach: SPARF



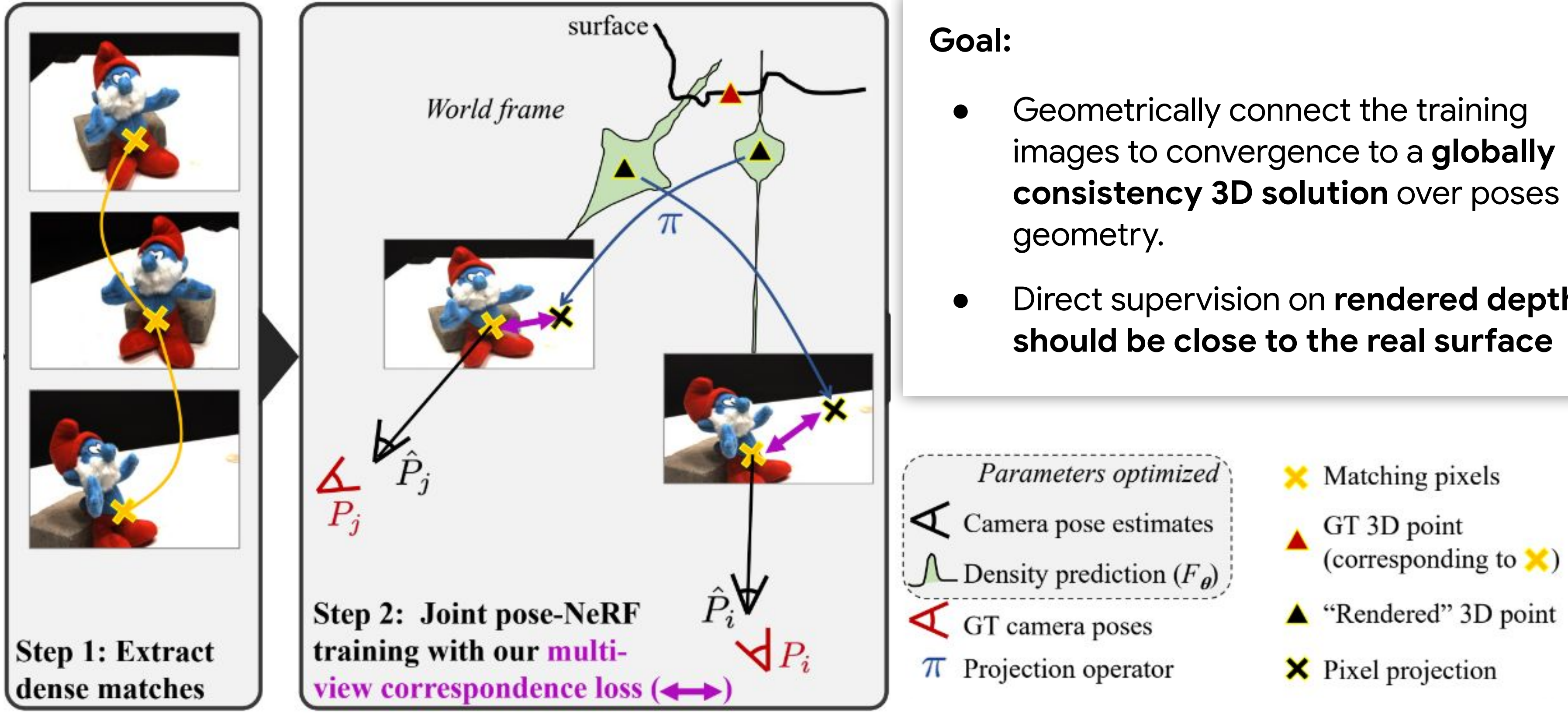
Main challenges:

- NeRF overfit to the few training images without learning a meaningful geometry
- Previous works use the photometric loss. It is applied to each image independently

We propose a joint pose-NeRF training strategy.

We add **two additional constraints into the NeRF optimization, which rely on multi-view geometry principles.**

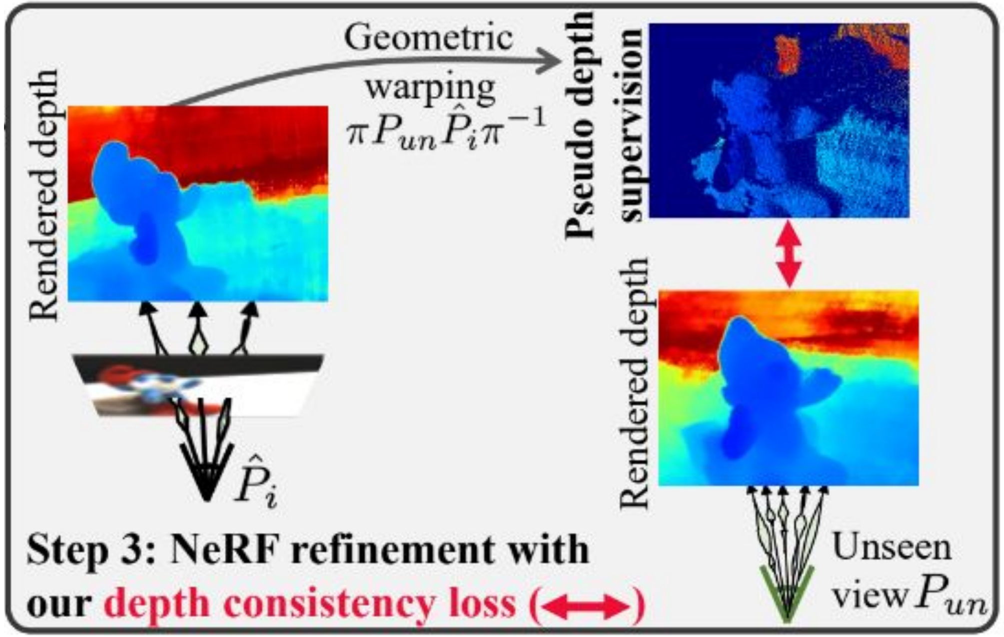
Our approach: Multi-view correspondence loss



Goal:

- Geometrically connect the training images to convergence to a **globally consistency 3D solution** over poses and geometry.
- Direct supervision on **rendered depth ⇒ should be close to the real surface**

Our approach: Depth-consistency loss



Goal: Ensure the reconstructed scene is **consistent from any viewing directions**, including the ones without RGB supervision.

Main idea: Use the rendered depth from the training viewpoints to create pseudo-depth supervision for novel, unseen viewpoints.

We also include a visibility mask, to tackle occlusion.

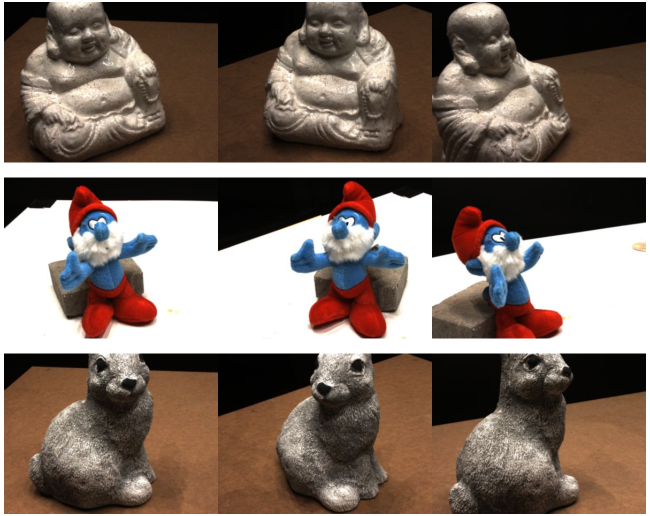
This optimizes over the neural radiance field weights only. The poses are fixed here.

Results: Experimental set-up

- Evaluation on multiple datasets: object-centered, forward-facing scenes, indoor non-forward-facing scenes.
- Sparse-view scenario: only 3 available. Results for 6 or 9 in the paper.
- Different ‘noisy poses’ initializations.
- In the paper, results with fixed ground-truth poses as well.

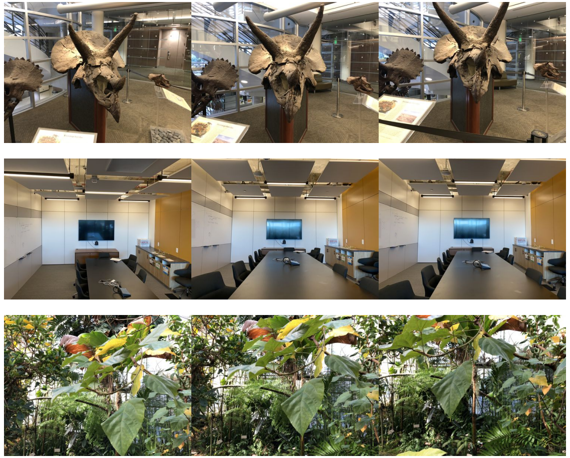
DTU dataset

(DTU Informatics 2010, Aanaes et al)



LLFF dataset

(BMVA 2021, Shafiei et al)



Replica dataset

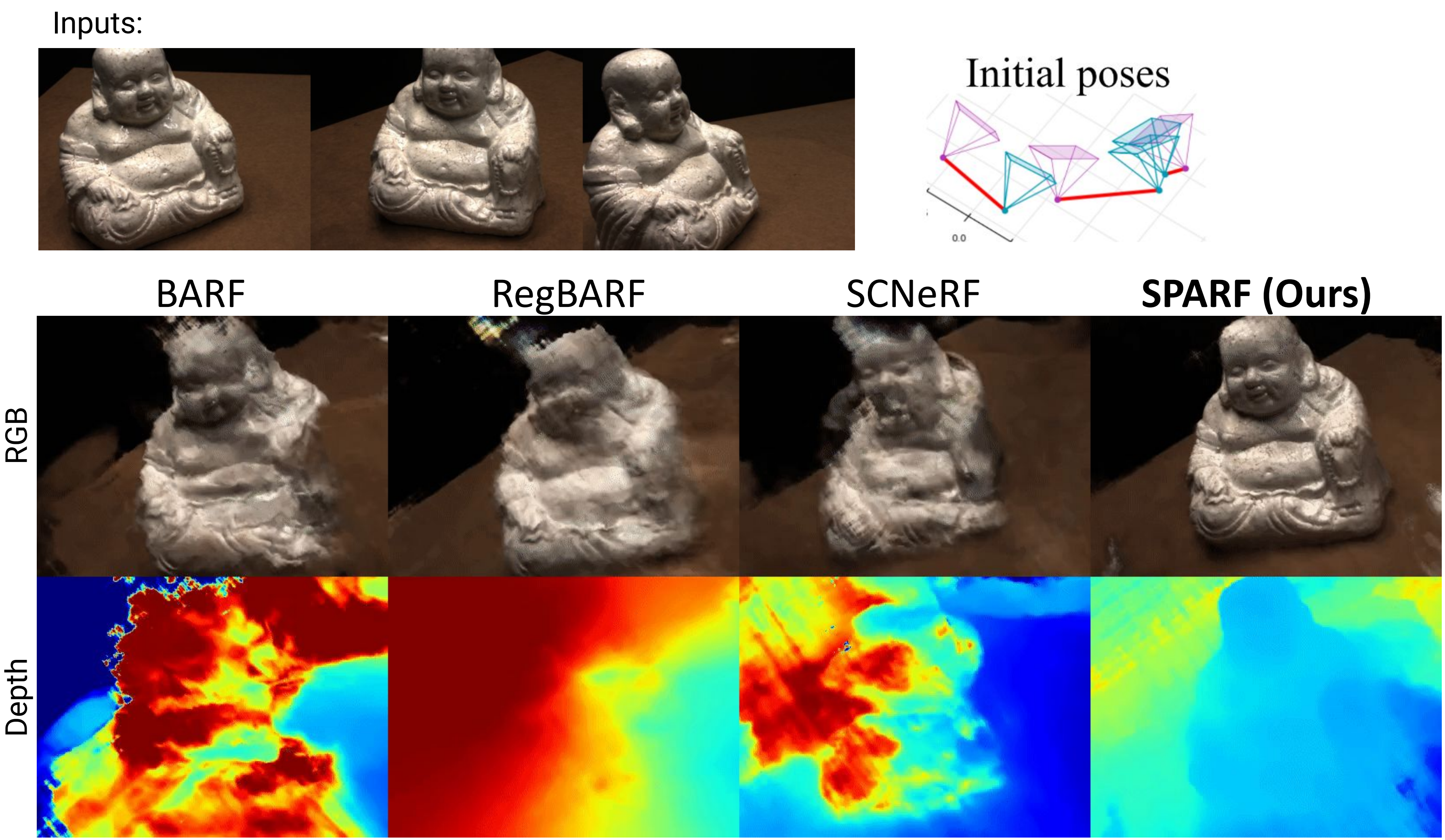
(2019, Straub et al.)

Results: Joint pose-NeRF training on **DTU (3 views) from noisy poses.**

- DTU contains object-level scenes with wide-baseline views spanning a half hemisphere.
- Noisy poses created by synthetically perturbing the ground-truth poses with 15% of Gaussian noise.
- Initial rotation error ≅ 15°
- Initial translation error ≅ 71.0

Method	Rot. ↓	Trans. ↓	PSNR ↑	SSIM ↑	LPIPS ↓	DE ↓
BARF [23]	10.33	51.5	10.71 (9.76)	0.43 (0.62)	0.59 (0.36)	1.90
RegBARF [23,31]	11.20	52.8	10.38 (9.20)	0.45 (0.62)	0.61 (0.38)	2.33
DistBARF [4,23]	11.69	55.7	9.50 (9.15)	0.34 (0.76)	0.67 (0.36)	1.90
SCNeRF [20]	3.44	16.4	12.04 (11.71)	0.45 (0.66)	0.52 (0.30)	0.85
SPARF (Ours)	**1.81**	**5.0**	**17.74 (18.92)**	**0.71 (0.83)**	**0.26 (0.13)**	**0.12**

[23] C.H. Lin, M.W. Ma, A. Torralba, S. Lucey. Barf: Bundle-adjusting neural radiance fields. ICCV 2021
[20] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, J. Park. Self-calibrating neural radiance fields. ICCV 2021
[31] M. Niemeyer, J.T. Barron, B. Mildenhall, M.S. Sajjadi, A. Geiger, N. Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. CVPR 2022.
[4] J.T. Barron, B. Mildenhall, D. Verbin, P.P. Srinivasan, P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR 2022

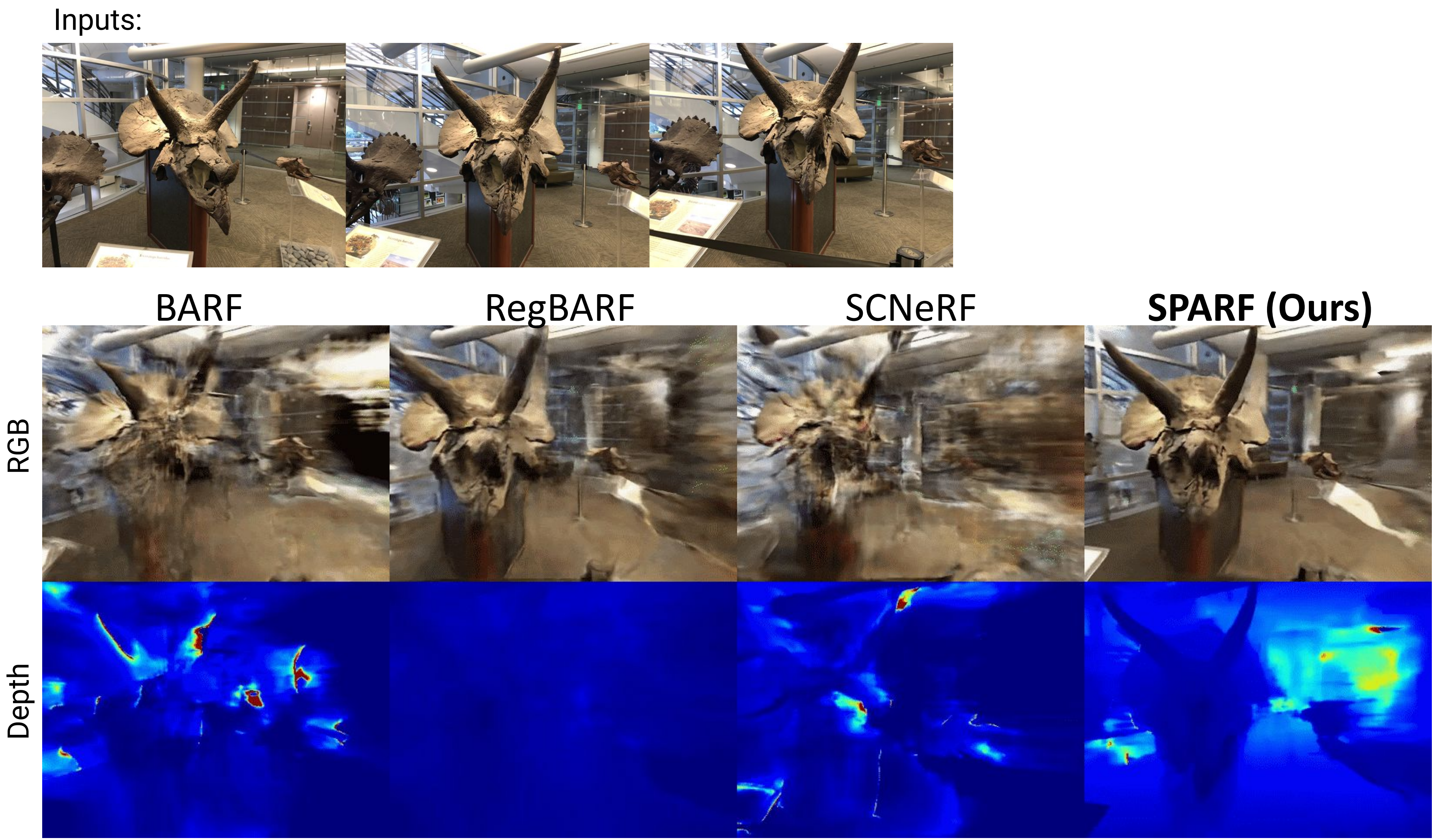

Inputs:
Initial poses
0.0
BARF
RegBARF
SCNeRF
SPARF (Ours)
RGB
Depth

Results: Joint pose-NeRF training on **LLFF (3 views) from identity poses.**

- LLFF contains forward-facing views
- We start from identity poses and refine them along with training the NeRF

	Rot. (°) ↓	Trans. (×100) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BARF [23]	2.04	11.6	17.47	0.48	0.37
RegBARF [23, 31]	1.52	5.0	18.57	0.52	0.36
DistBARF [4, 23]	5.59	26.5	14.69	0.34	0.49
SCNeRF [20]	1.93	11.4	16.52	0.42	0.47
SPARF (Ours)	**1.15**	**4.9**	**19.38**	**0.57**	**0.35**

[23] C.H. Lin, M.W. Ma, A. Torralba, S. Lucey. Barf: Bundle-adjusting neural radiance fields. ICCV 2021
[20] Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, J. Park. Self-calibrating neural radiance fields. ICCV 2021
[31] M. Niemeyer, J.T. Barron, B. Mildenhall, M.S. Sajjadi, A. Geiger, N. Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. CVPR 2022.
[4] J.T. Barron, B. Mildenhall, D. Verbin, P.P. Srinivasan, P. Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR 2022

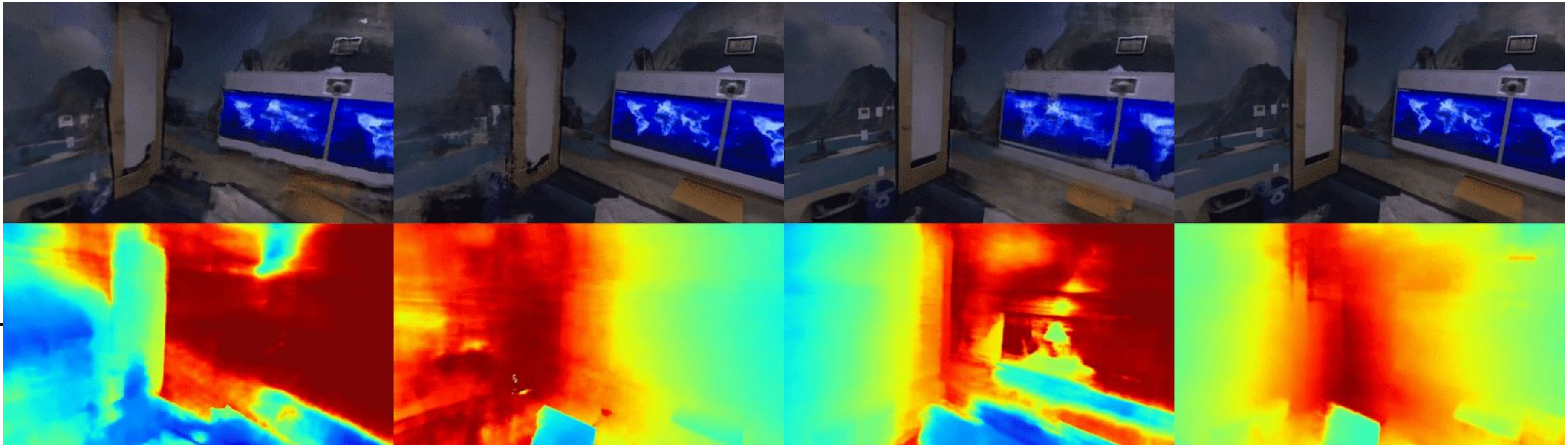

Inputs:
BARF
RegBARF
SCNeRF
SPARF (Ours)
RGB
Depth

REPLICA Dataset. Inputs:

BARF | DS-NeRF | SCNeRF | **SPARF (Ours)**

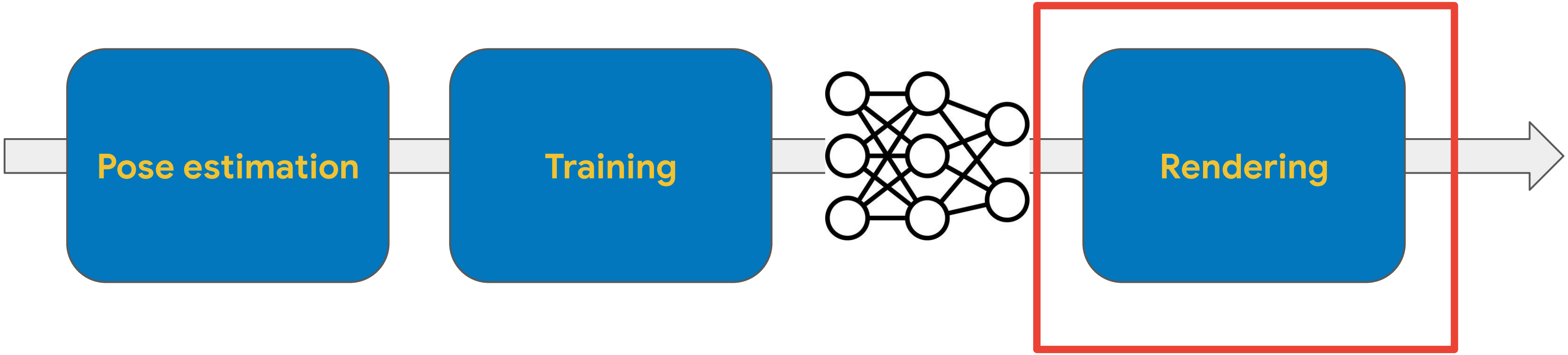
RGB

Depth



Limitations for real applications

- **Pose estimation** is a critical step. Nerf requires dense pose coverage with accurate pose estimation. **Noise and sparsity** highly affect quality.
- **Rendering in real time is still a problem**, especially on embedded/mobile settings

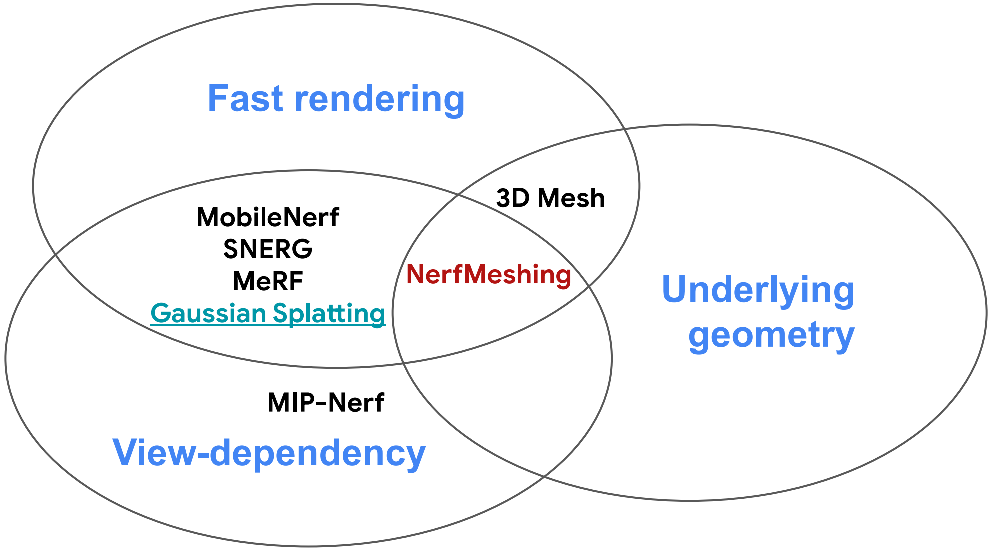


NeRFMeshing: Distilling Neural Radiance Fields into Geometrically-Accurate 3D Meshes

Marie-Julie Rakotosaona   Fabian Manhardt   Diego Martin Arroyo
Michael Niemeyer   Abhijit Kundu   Federico Tombari

Nerf rendering dilemma

NeRFs are optimized exclusively for visual consistency -> lack of accurate underlying geometry



1. NerfMeshing extracts a **neural mesh** from Nerf, so that **geometry** can be used for shape relighting, physics-based simulation, geometry-based compositionality, ..
2. Enables **fast rendering**, since meshes are much faster to render than radiance fields
3. While preserving **view dependency**

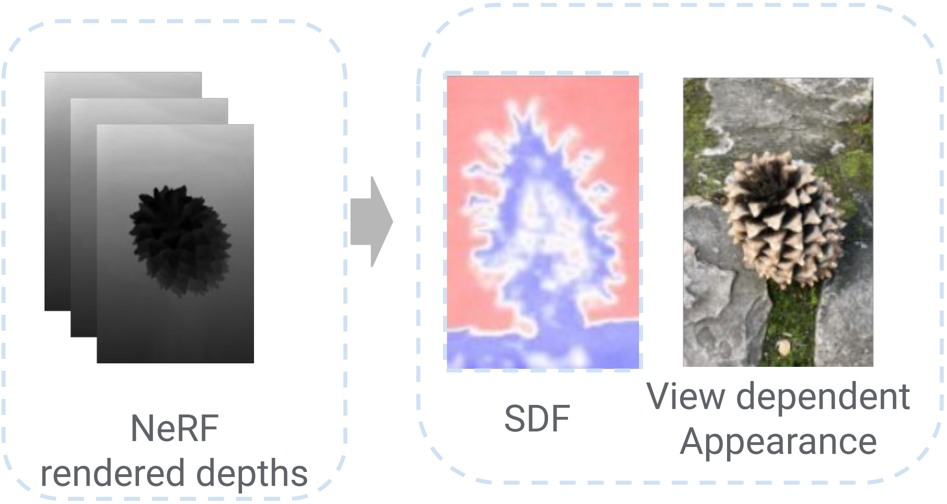
Method

Main challenges:

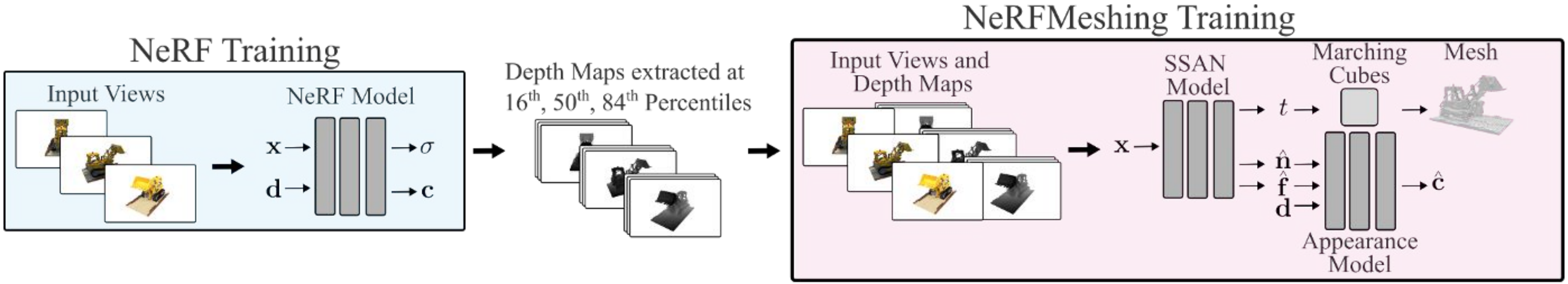
- NeRF density field does not represent a unique surface
- How to extract a surface from a pre-trained NeRF?

Main idea:

- Use NeRF rendered depth maps to infer:
  - A SDF
  - A set of view-dependent appearance features



NeRFMeshing



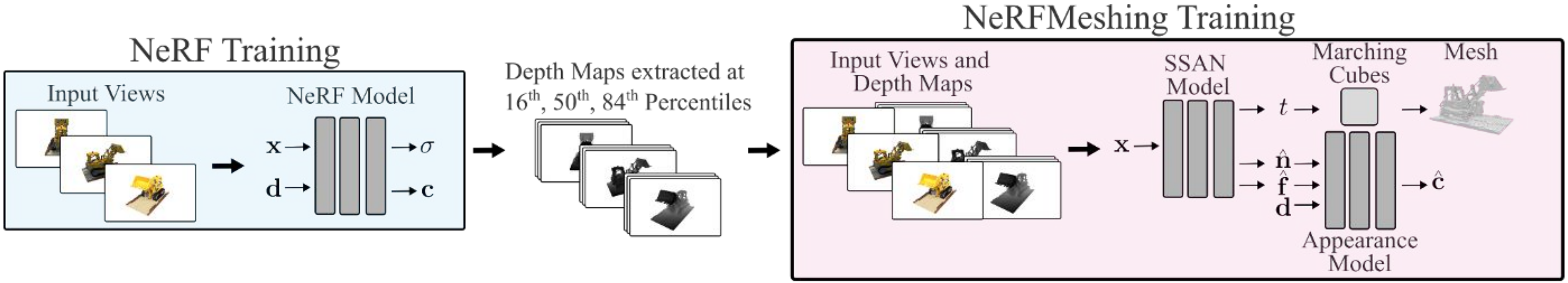
We train two additional networks:

- **Signed Surface Approximation Network (SSAN)**: trained to regress, from a given position **x,** a TSDF value **t**, normal **n** and appearance features **f**
- **Appearance model:** takes predicted normal **n** and appearance features **f** together with the viewing direction **d** to regress a view-dependant color **c**

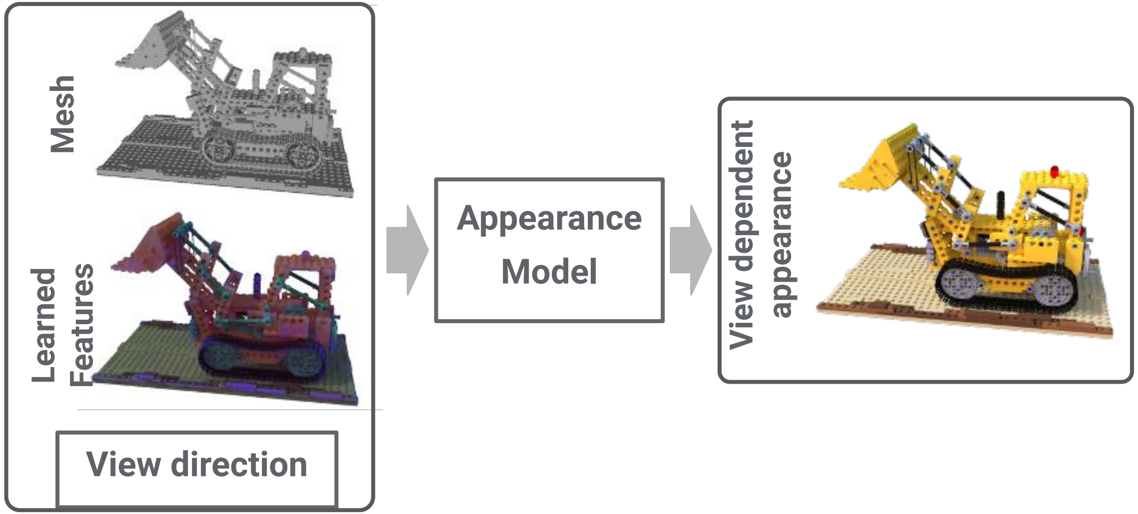
Advantages:

- Small and modular method that can extract a **mesh from any pre-trained NeRF**
- Mesh representation can be **used in computer graphics rendering pipelines with minimal changes** to include the view dependent network
- Appearance model allows faster rendering than Nerf

NeRFMeshing: Rendering Appearance

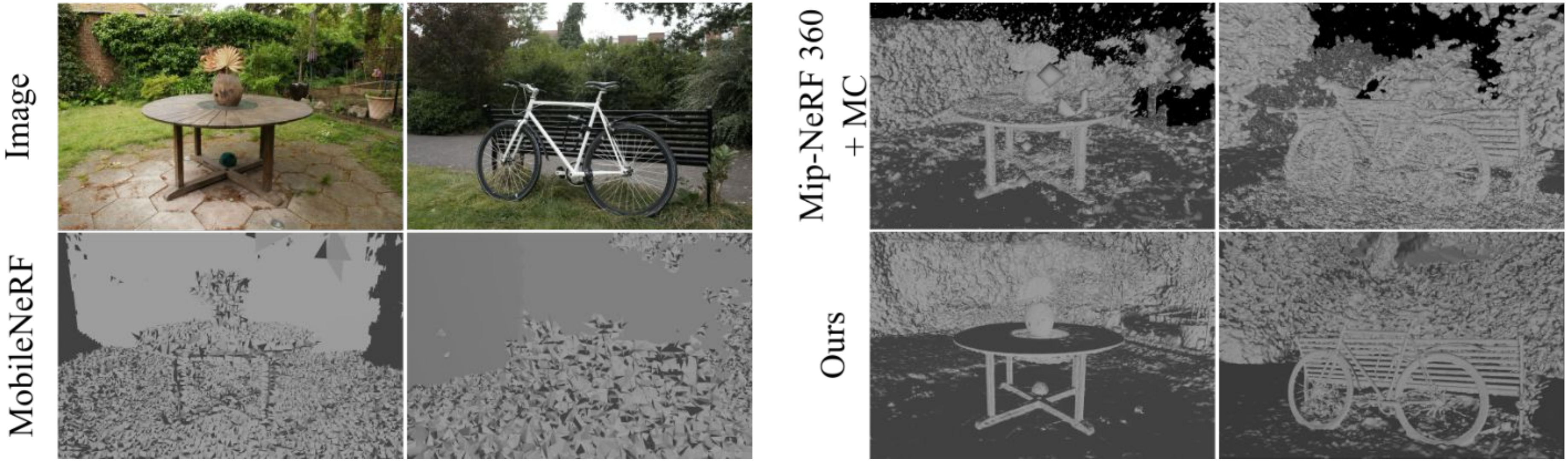


We use learned features on the mesh and view directions to infer the appearance at rendering time



Results

Results: Geometry Comparison



Results: Unbounded Scene Rendering

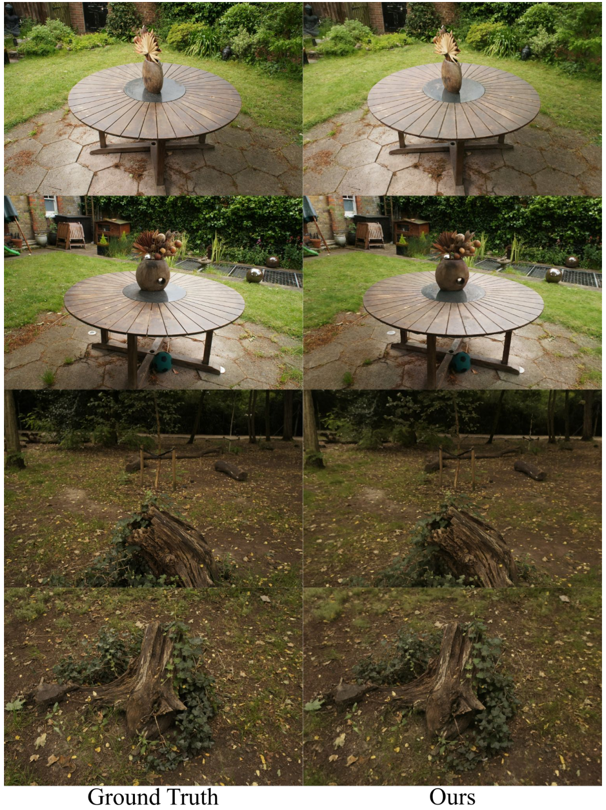
Results: Object Rendering

Results: Physics Based Simulations

Conclusion, future directions

- Modular method that can be trained on any pretrained NeRF
- Mesh format enables training in computer graphics pipelines with minimal changes for the view dependance
- Accurate mesh enables physics based simulations

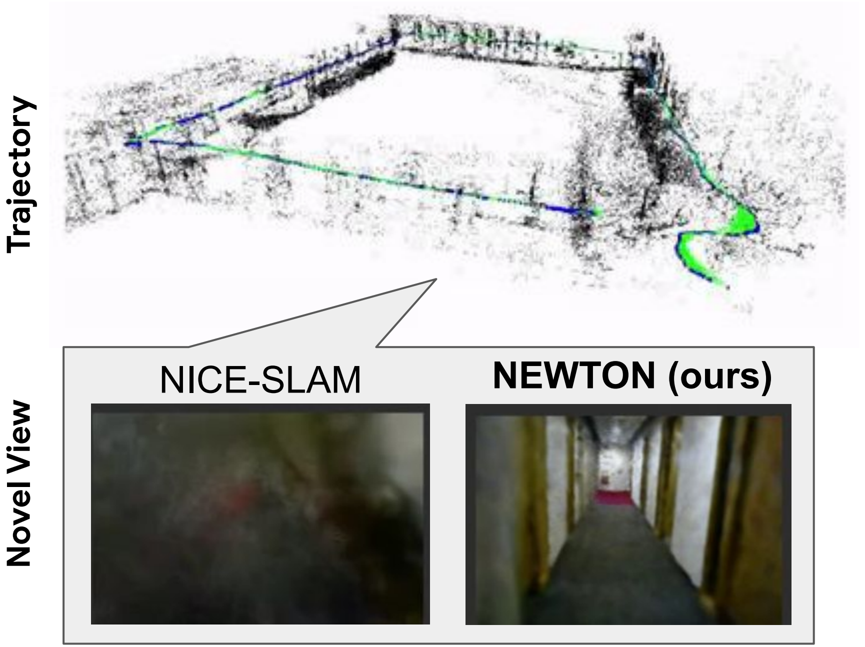
- **Mesh are still limited in the way they can represent e.g. thin structures and specular/reflective surfaces**



Future direction: online NeRF reconstruction/SLAM

- **NEWTON: Neural View-Centric Mapping for On-the-Fly Large-Scale SLAM** ([paper]())
    - Neural Field mapping method which works with an dynamic loop-closing SLAM system
    - Dynamically allocate, train and render multiple local NeRFs
    - Strong robustness to large pose updates.

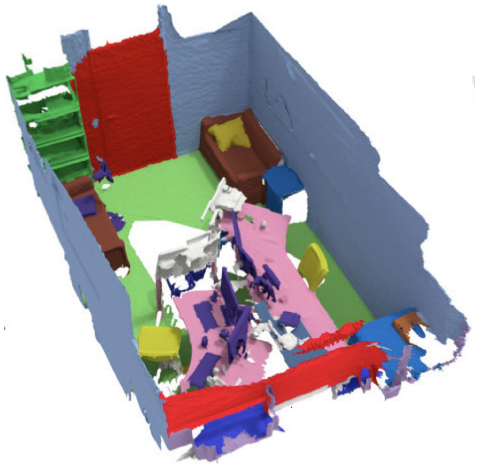
Live NeRF reconstruction in real-time



(2/3)
Open Set 3D Semantic Segmentation

Semantics for 3D Scene Understanding

Typical Tasks



3D Semantic Segmentation

Assign a semantic class to each point in a given 3D scene.

3D Instance Segmentation

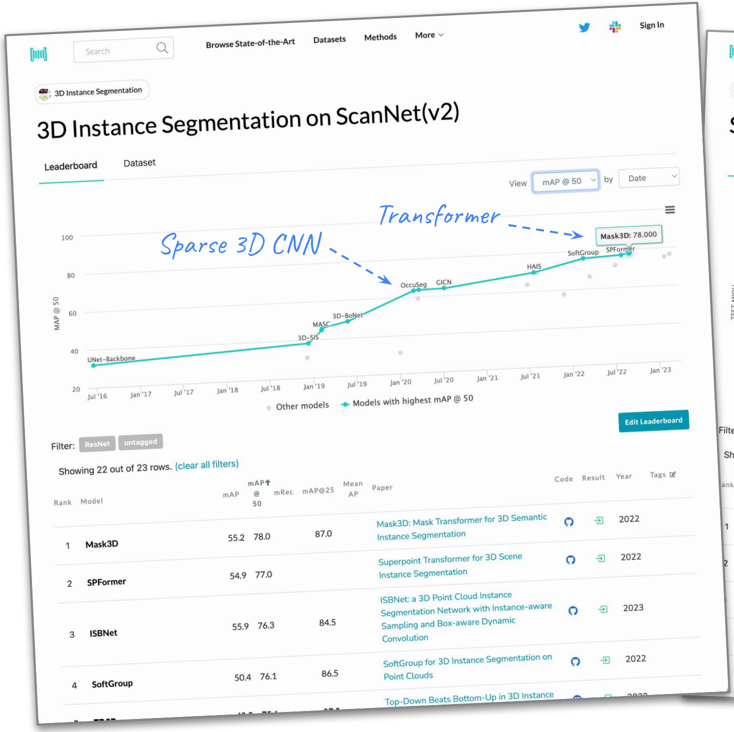
Predict instance masks and semantic labels for each object in a given 3D scene.

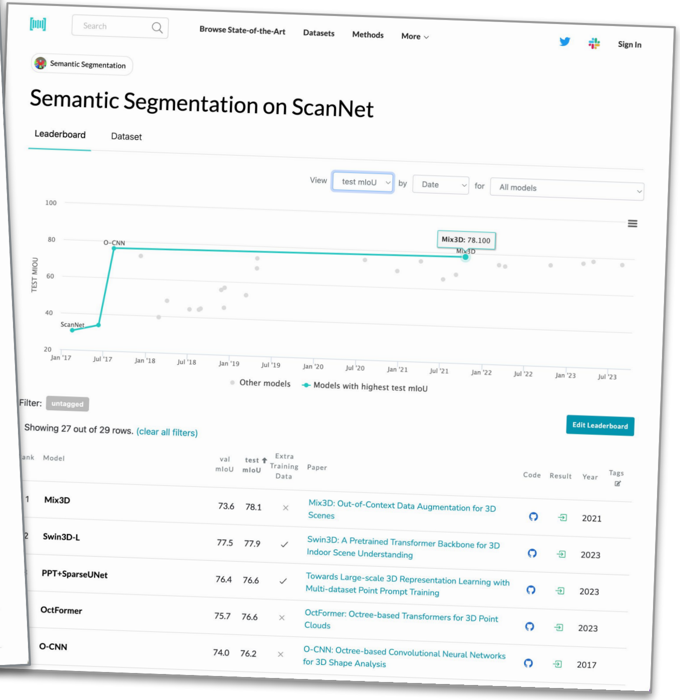
3D Object Detection

Detect the 3D bounding box of each object in a given 3D scene.

3D Scene Understanding

Active field of research with significant progress over the last years

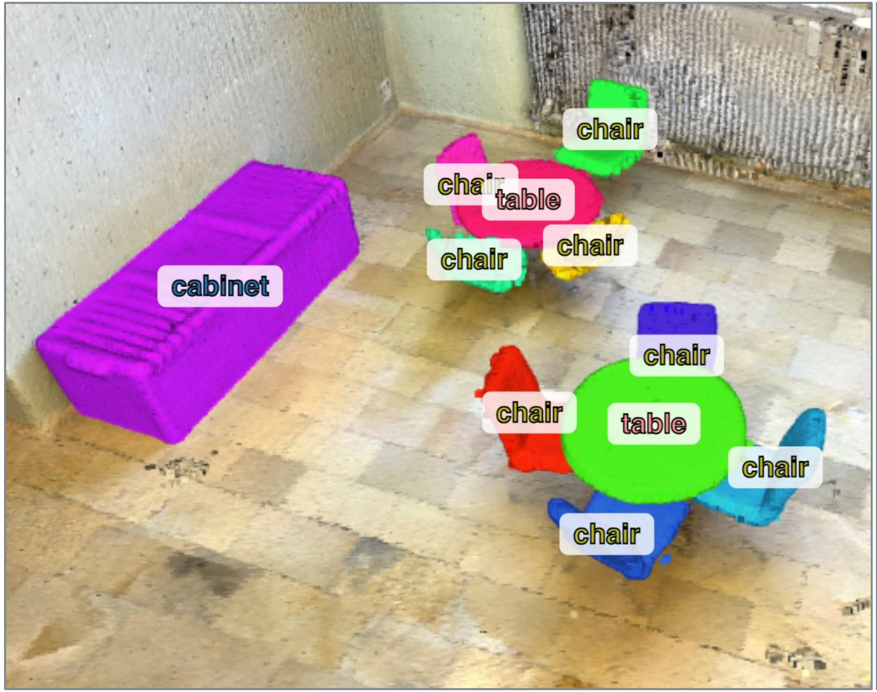


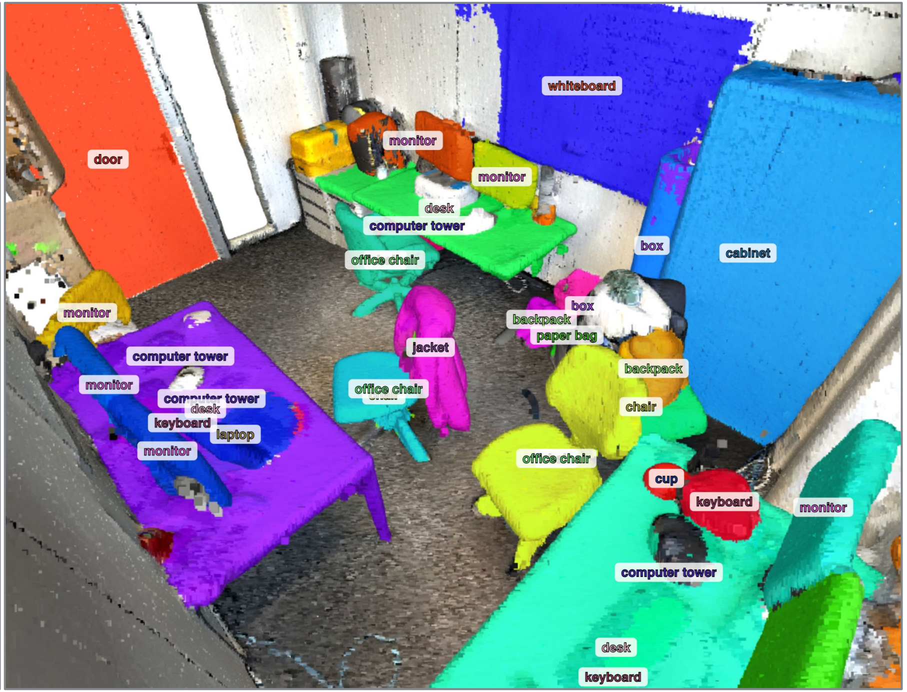


3D Semantic Segmentation: https://paperswithcode.com/sota/semantic-segmentation-on-scannet?metric=test%20mIoU
3D Instance Segmentation: https://paperswithcode.com/sota/3d-instance-segmentation-on-scannetv2?metric=mAP%20%40%2050

3D Scene Understanding

Example 3D Instance Segmentations from **Mask3D** 🎭 [1]

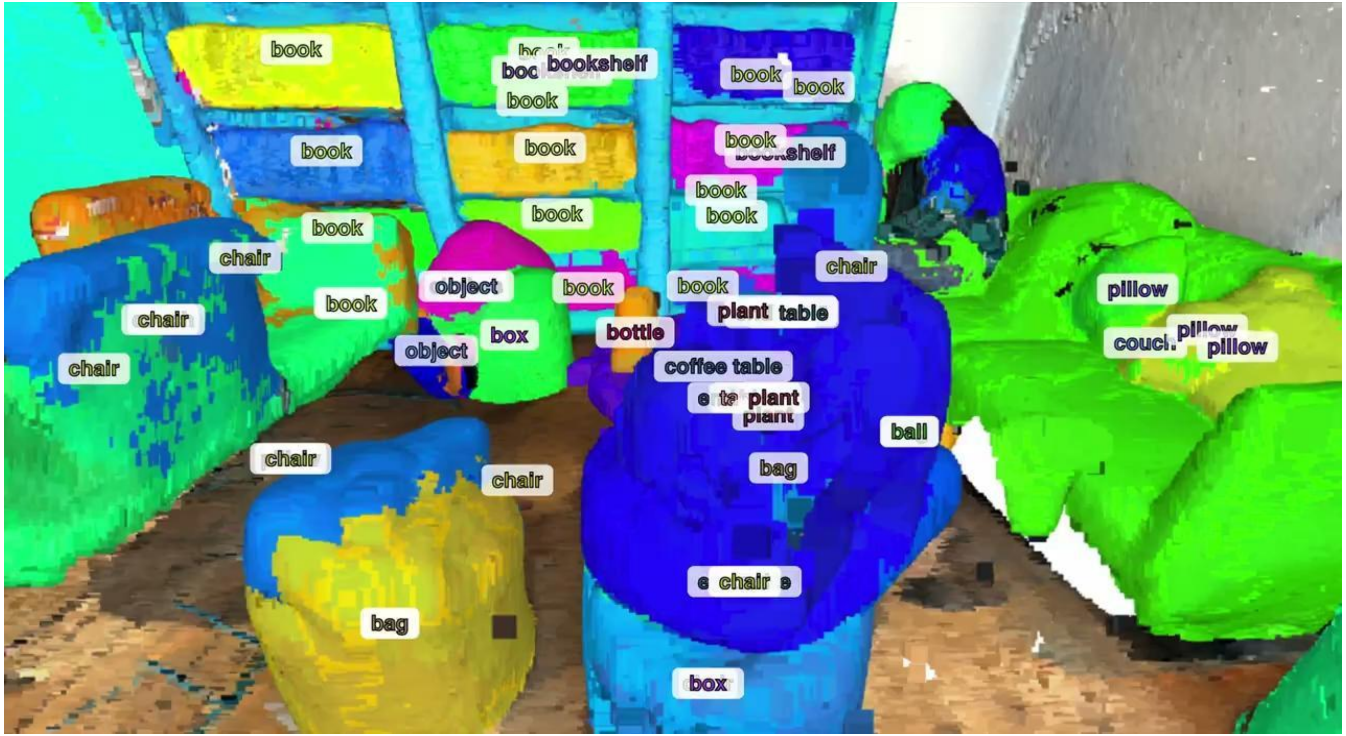




[1] Schult et al. "*Mask3D: Mask Transformer for 3D Instance Segmentation*" ICRA'23

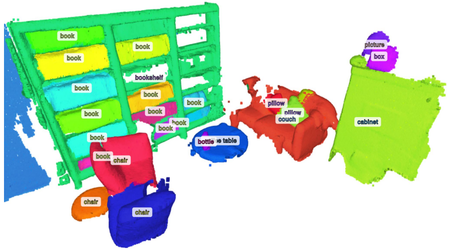
3D Scene Understanding

Example 3D Instance Segmentations from **Mask3D** 🎭 [1]



Input: 3D Point Cloud

Output: 3D Semantic Instances

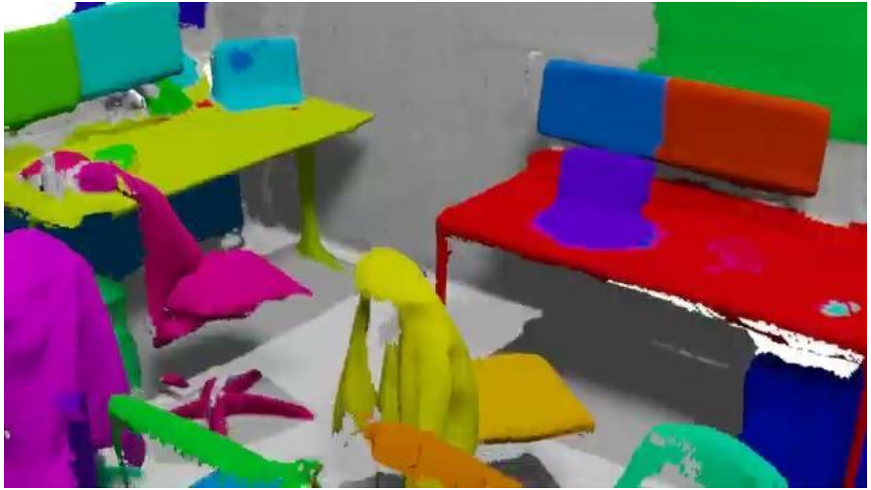


[1] Schult et al. "*Mask3D: Mask Transformer for 3D Instance Segmentation*" ICRA'23

3D Scene Understanding

Works well for semantic classes seen during training (closed-world setting)

Input 3D Scene

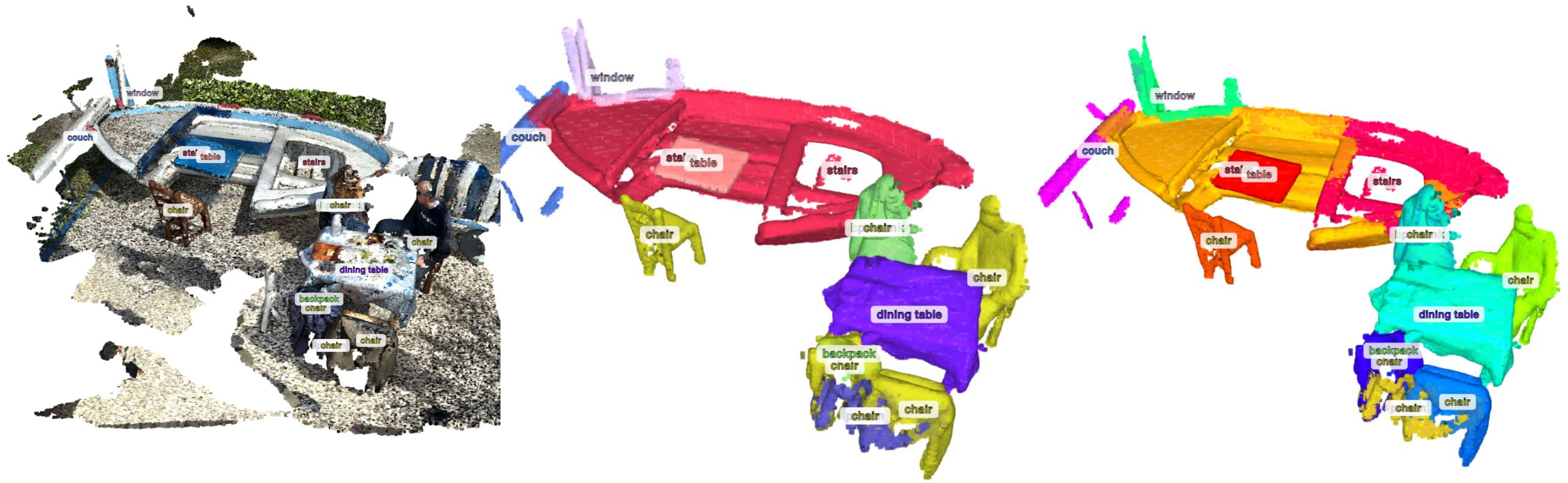


Predicted 3D Instance Masks

[1] Schult et al. "*Mask3D: Mask Transformer for 3D Instance Segmentation*" ICRA'23

3D Scene Understanding: *Limitations of **Closed-Set** Assumption*

Example "*in-the-wild*" scene (link)



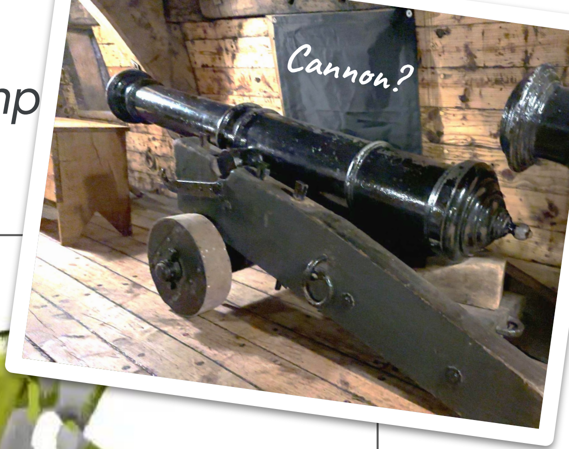
Input 3D Scene **3D Semantics** **3D Instance Masks**

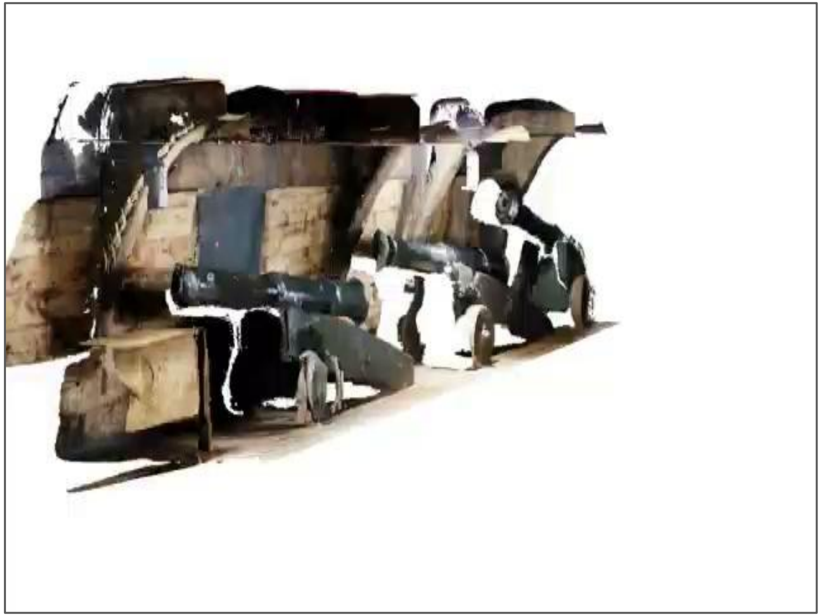
Couch Window Stairs Chair

Table Dining Table Backpack

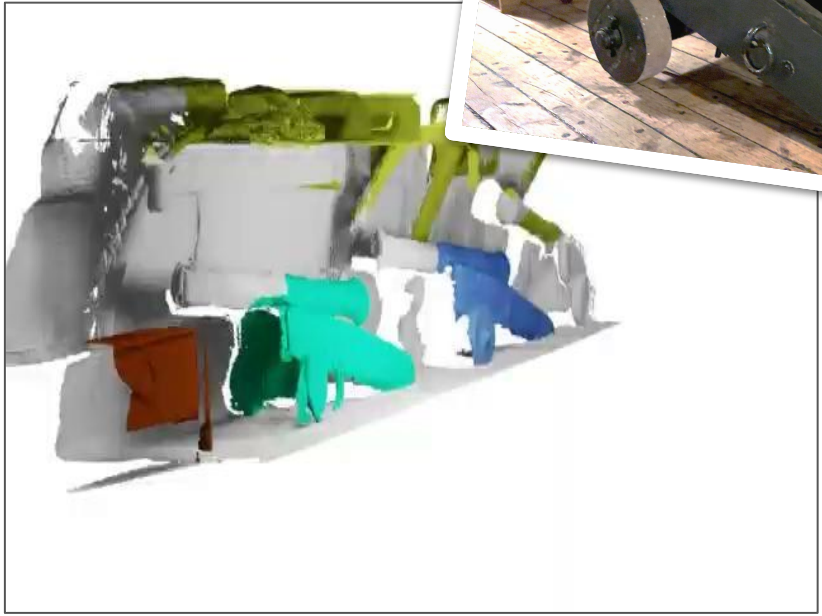
3D Scene Understanding: *Limitations of **Closed-Set** Assump*

Example "*in-the-wild*" scene (link)





Input 3D Scene



3D Semantics

Towards **Open-Set** 3D Scene Understanding

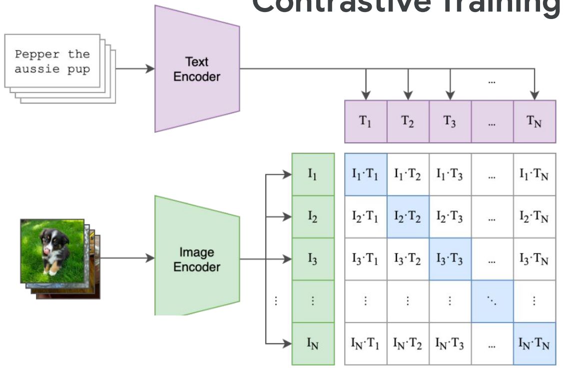
using Visual-Language Models (VLM)

Visual-language models such as CLIP or ALIGN [1,2] consist of an ***image***- and ***text***-encoder. They are trained:

- on internet-scale image-caption pairs
- in a contrastive manner

If the text caption describes the image, then the encodings of both modalities (text and image) correlate, otherwise they do not.

Contrastive Training



[1] Radford et al. "*Learning Transferable Visual Models From Natural Language Supervision*" ICML'21
[2] Jia et al. "*Scaling Up Visual and Vision-Language Representation*" ICML'21

Towards **Open-Set** 3D Scene Understanding (cont.)

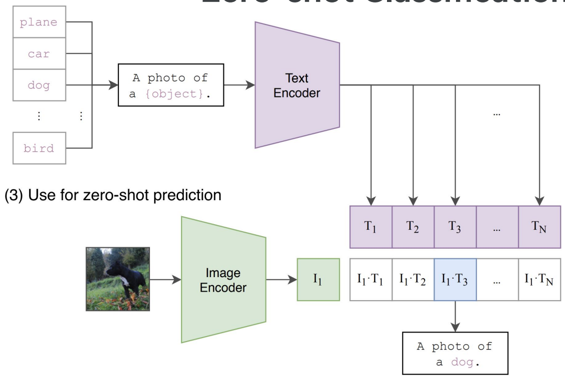
using Visual-Language Models (VLM)

We can use this mechanism for zero-short image classification:

1. Compute encoding of both the text and image
2. Take dot-product of normalized encodings
3. Image class corresponds to maximum response

Since VLMs are "*trained on the internet*", they have seen numerous and rare concepts which makes them great candidates for **open-set** scene understanding.

Zero-shot Classification



[1] Radford et al. "*Learning Transferable Visual Models From Natural Language Supervision*" ICML'21
[2] Jia et al. "*Scaling Up Visual and Vision-Language Representation*" ICML'21

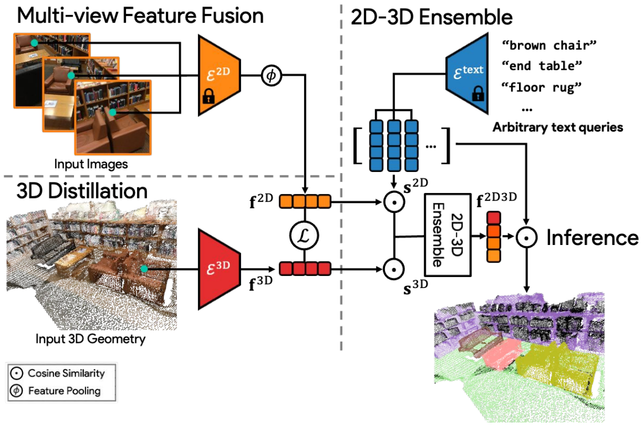
Open-Set 3D Semantic Segmentation

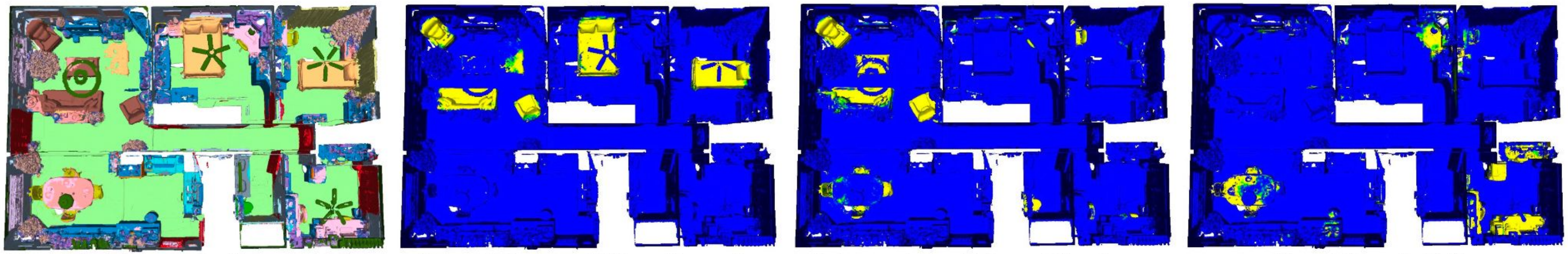
OpenScene: 3D Scene Understanding with Open Vocabularies

How do we transfer **open-set** scene understanding to **3D** scenes?

OpenScene [3] obtains **per-pixel** multi-view open-set features from LSeg [1] or OpenSeg [2] and projects them onto **3D points** of the scene point cloud.

A sparse 3D CNN is then trained to predict **per-point** open-set features distilled from 2D ones. 2D and 3D features are ensembled via CLIP supervision, resulting in a scene with associated per-point open-set features.





Zero-shot Semantic Segmentation | **"anything soft"** - Property | **"where to sit"** - Affordance | **"work"** - Activity

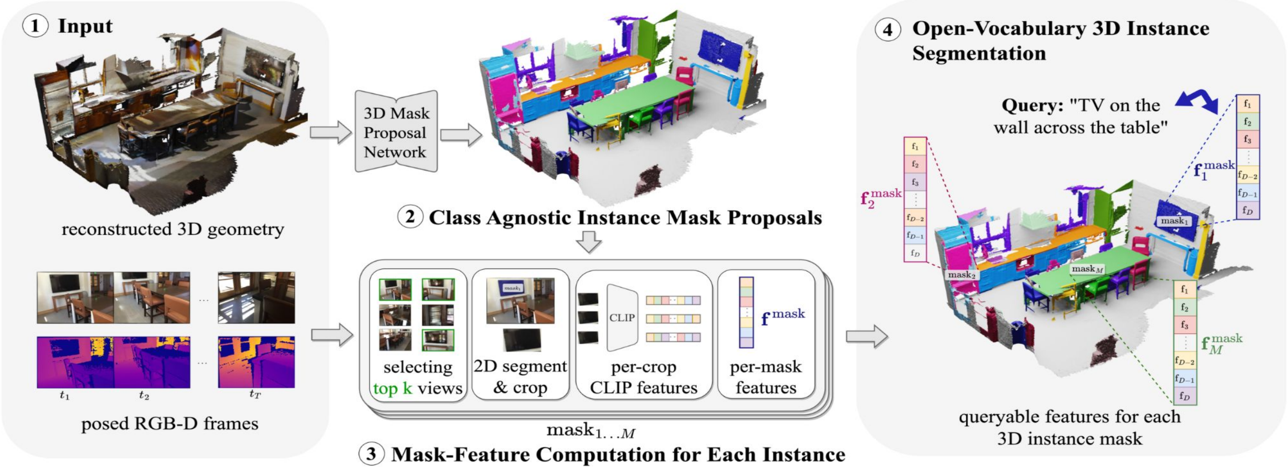
[1] Li et al. "*Language-driven Semantic Segmentation*" ICLR'22
[2] Ghiasi et al. "*Scaling open-vocabulary image segmentation with image-level labels*" ECCV'22
[3] Peng et al. "*OpenScene: 3D Scene Understanding with Open Vocabularies*" CVPR'23

Open-Set 3D **Instance** Segmentation

OpenMask3D: Open-Vocabulary 3D Instance Segmentation

For many applications it is important to differentiate between **multiple** instances of the **same** class.

OpenMask3D [1] obtains **per-segment** open-set features by first segmenting the 3D scene into class-agnostic segments using Mask3D [2],
then the projected segments are cropped in 2D at multiple scales to obtain CLIP features for each 3D segment.

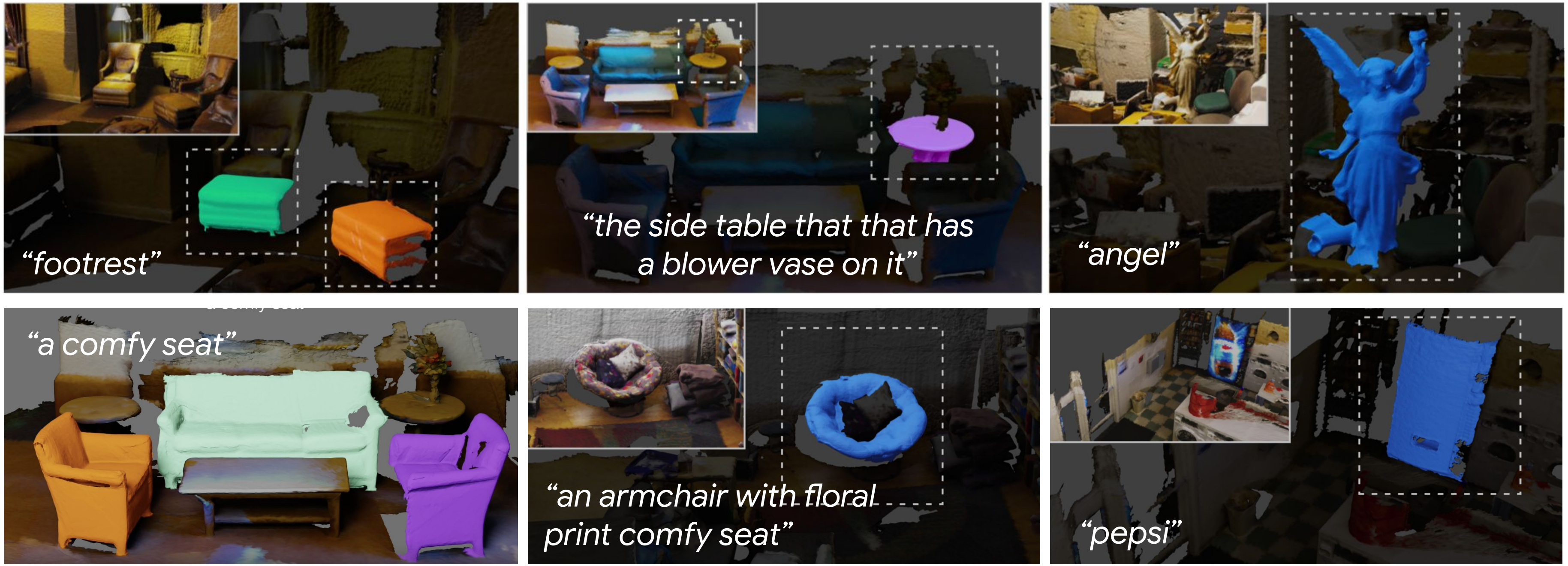


[1] Takmaz et al. "OpenMask3D: Open-Vocabulary 3D Instance Segmentation" arXiv'23
[2] Schult et al. "*Mask3D: Mask Transformer for 3D Instance Segmentation*" ICRA'23

Open-Set 3D **Instance** Segmentation

OpenMask3D: Open-Vocabulary 3D Instance Segmentation

Multiple object instances | **Rich query descriptions** | **Long-tail objects**



[1] Takmaz et al. "OpenMask3D: Open-Vocabulary 3D Instance Segmentation" arXiv'23

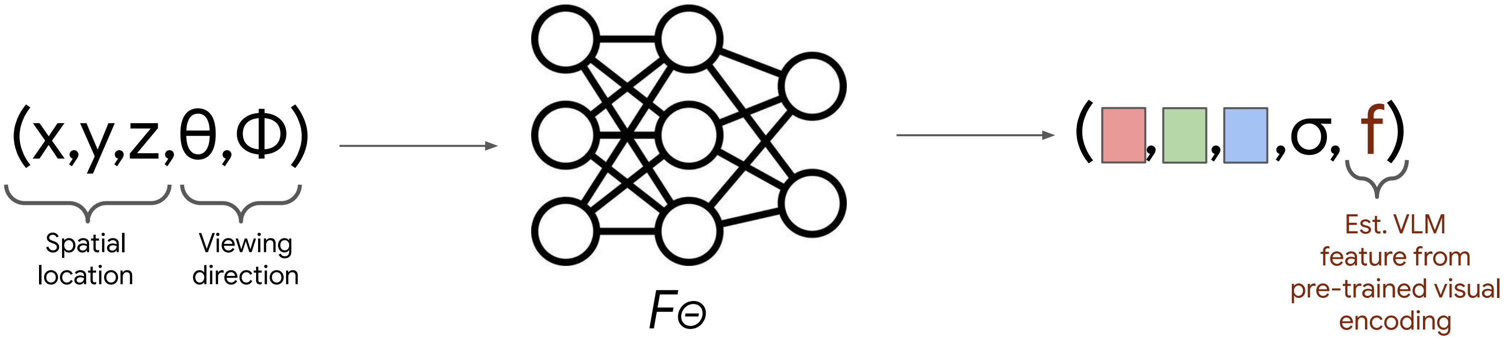
What about open set with implicit representations?

Open-Set 3D Scene Understanding using **Implicit** Representations

Explicit v.s. Implicit Representations: Polygon Meshes and Point Clouds or NeRF Representations?

Can we use Implicit NeRF representations for Open-Set 3D Scene Understanding?

Idea: Ground CLIP features (or any other features from a pre-trained visual encoding aligned with language) volumetrically inside NeRFs (in addition to color and density).



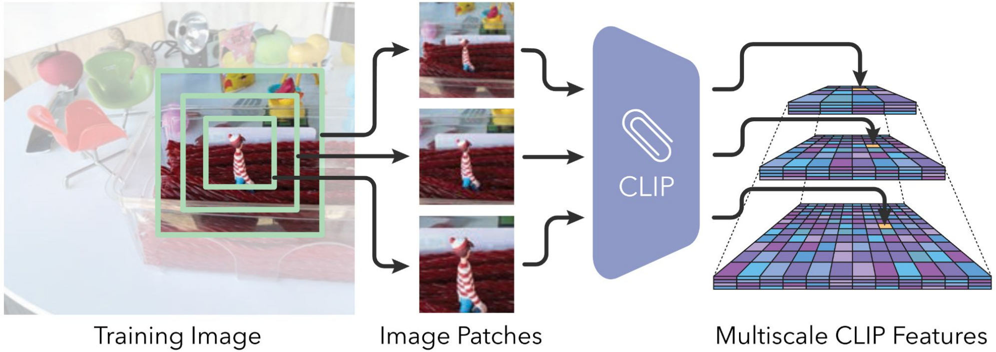
[1] Kerr et al. "LERF: Language Embedded Radiance Fields" ICCV'23
[2] Engelmann et al. "*Open-Set 3D Scene Segmentation with Rendered Novel Views*", arXiv'23

Open-Set 3D Scene Understanding using **Implicit** Representations

LERF: Language Embedded Radiance Fields

LERF [1] distills CLIP features into a NeRF representation, based on CLIP encodings of multi-scale image patches.

- Using the original CLIP image-encoder does not require fine-tuning
- Since CLIP is a global feature (per patch/image), it includes a strategy to efficiently compute CLIP at the “right” object scale for each scene component
- Similar to DFF [2], LERF regresses also DINO [3] features to spatially regularize the CLIP space (DINO is sensitive to location)



[1] Kerr et al. "LERF: Language Embedded Radiance Fields" ICCV’23
[2] Kobayashi et al. *“Decomposing NeRF for Editing via Feature Field Distillation”* NeuRIPS’22
[3] Caron et al. *“Emerging Properties in Self-Supervised Vision Transformers” ICCV’21*

Open-Set 3D Scene Understanding using **Implicit** Representations

LERF: Language Embedded Radiance Fields

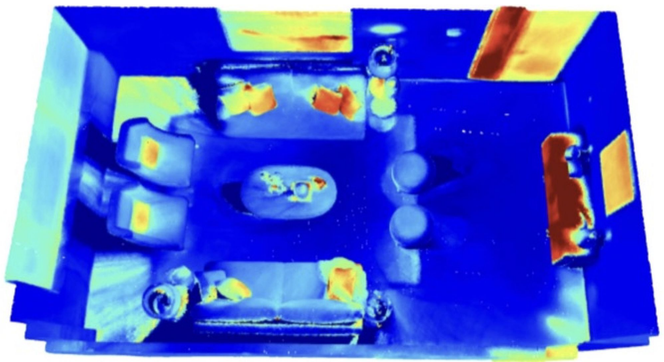
[1] Kerr et al. "LERF: Language Embedded Radiance Fields" ICCV’23

Open-Set 3D Scene Understanding using **Implicit** Representations

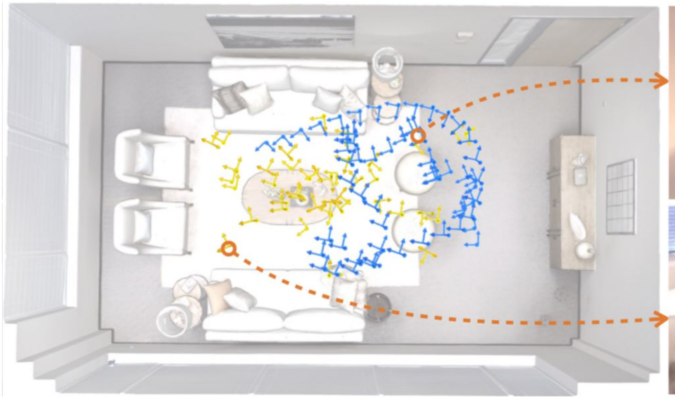
Open-Set 3D Scene Segmentation with Rendered Novel Views

Contributions:

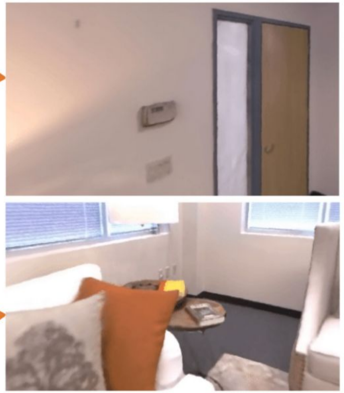
- Replace the global visual embedding (like CLIP, Align) with a **pixel-wise embedding (LSeg, OpenSeg)**, achieving a **significant simplification of the framework** (no more multiscale needed) and **better quality** on segment borders
- Use Nerf's NVS capabilities to **render new views** where the original camera trajectory missed important scene parts, based on where the extracted open-set features disagree



Uncertainty



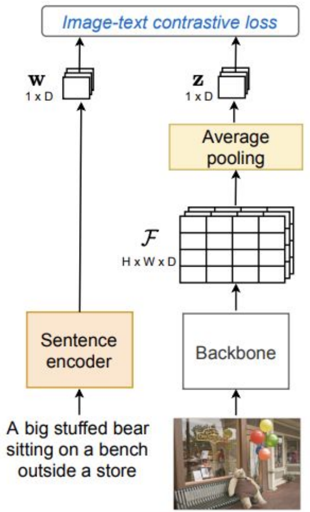
Existing and novel poses



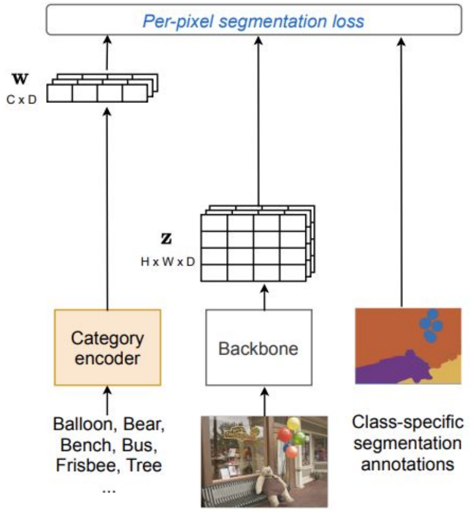
Novel views

[1] Engelmann et al. "*Open-Set 3D Scene Segmentation with Rendered Novel Views*" arXiv'23

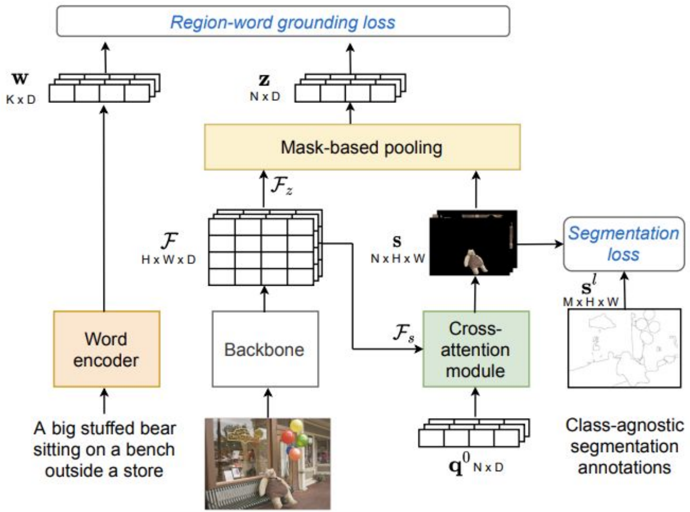
Pre-trained VLM encoding: image/patch-, region/mask- and pixel-level



CLIP[1] / ALIGN[2]
(Figure taken from [3])



LSeg [3]

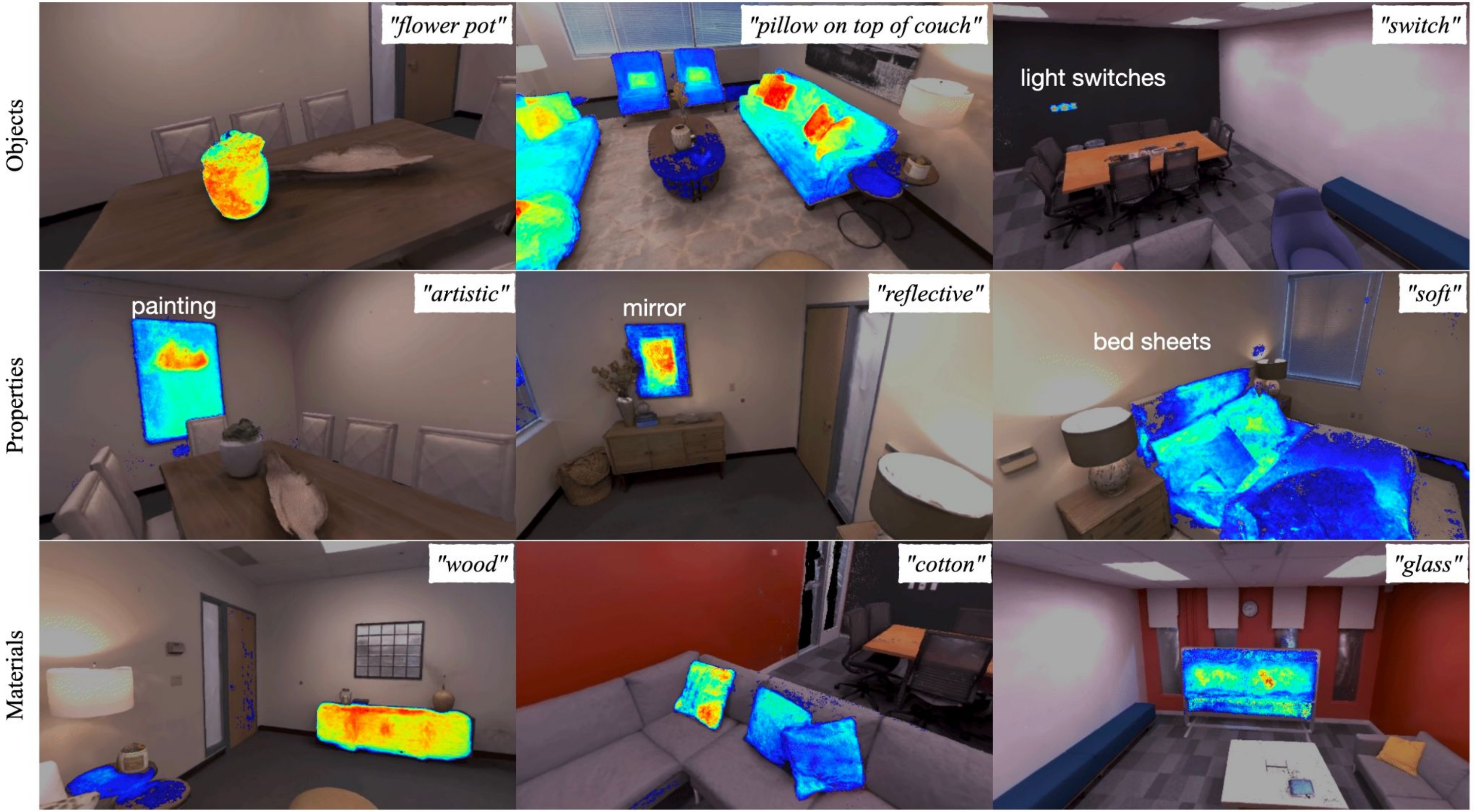


OpenSeg[4]

[1] Radford et al. "*Learning Transferable Visual Models From Natural Language Supervision*" ICML'21
[2] Jia et al. "*Scaling Up Visual and Vision-Language Representation*" ICML'21
[3] Li et al, Language-driven Semantic Segmentation, ICLR 22
[4] Ghiasi et al, Scaling Open-Vocabulary Image Segmentation with Image-Level Labels, ECCV 22

OpenSet 3D Scene Understanding: *Object, Properties, Materials*

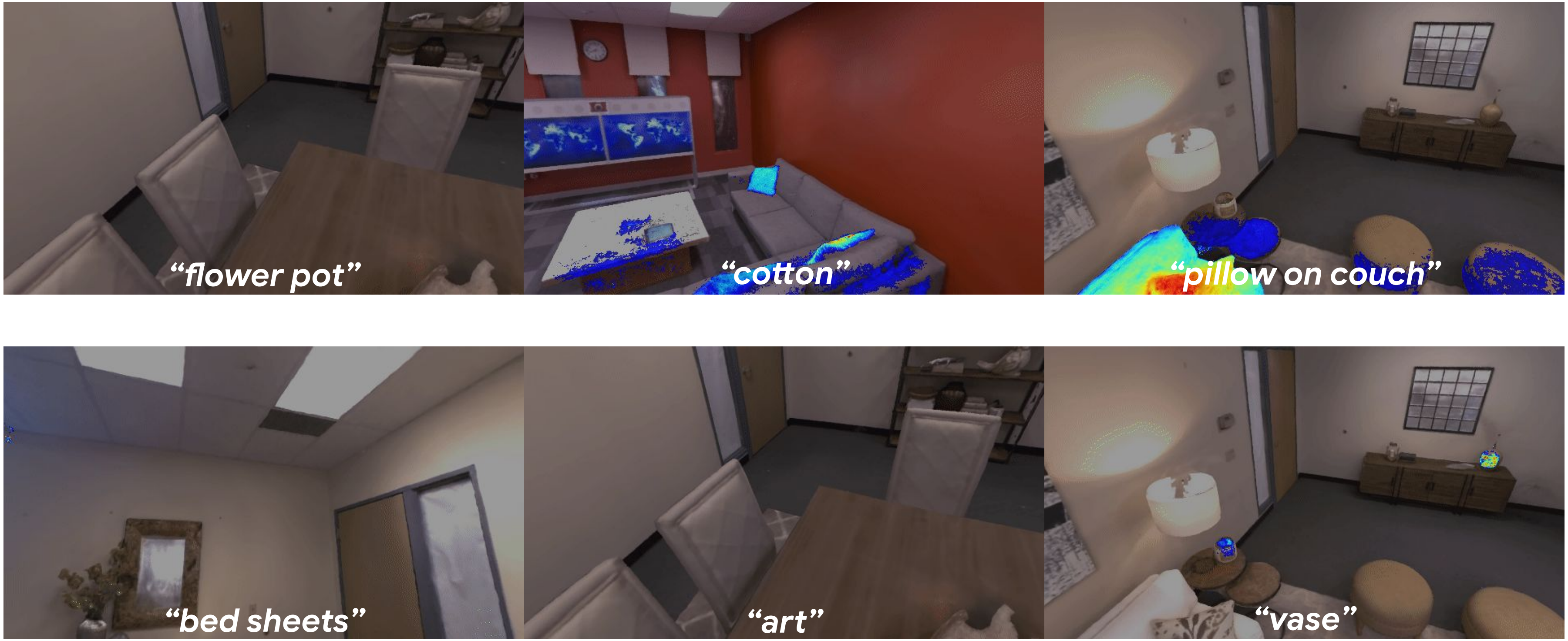
Localize arbitrary objects, properties or materials using open-vocabulary text queries.



Engelmann et al. "*Open-Set 3D Scene Segmentation with Rendered Novel Views*" arXiv'23

OpenSet 3D Scene Understanding: *Object, Properties, Materials*

Localize arbitrary objects, properties or materials using open-vocabulary text queries.



Engelmann et al. “*Open-Set 3D Scene Segmentation with Rendered Novel Views*” arXiv’23

OpenSet 3D Scene Understanding: *Object, Properties, Materials*

Localize arbitrary objects, properties or materials using open-vocabulary text queries.

Engelmann et al. "*Open-Set 3D Scene Segmentation with Rendered Novel Views*" arXiv'23

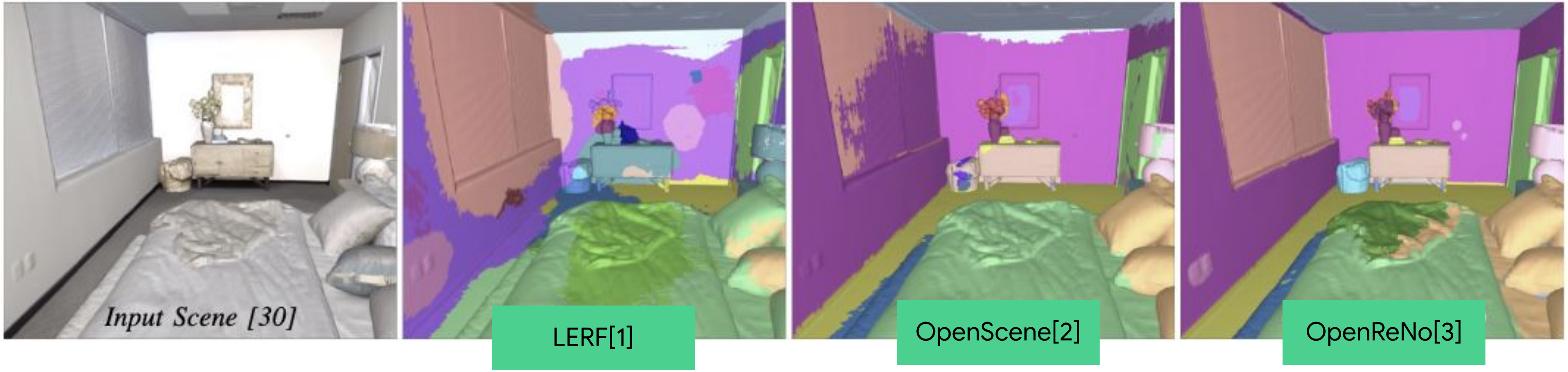
OpenSet 3D Scene Understanding: *Object, Properties, Materials*

Localize arbitrary objects, properties or materials using open-vocabulary text queries.

Engelmann et al. "*Open-Set 3D Scene Segmentation with Rendered Novel Views*" arXiv'23

OpenSet 3D Scene Understanding: *Zero-Shot 3D Semantic Segmentation*

Evaluation on Replica



[1] Kerr et al. "LERF: Language Embedded Radiance Fields" ICCV'23
[2] Peng et al. "*OpenScene: 3D Scene Understanding with Open Vocabularies*" CVPR'23
[3] Engelmann et al. "*Open-Set 3D Scene Segmentation with Rendered Novel Views*" arXiv'23

(3/3)
3D Scene Graphs

3D Scene Understanding with Scene Graphs

Definition of semantic scene graphs (SSG)

SSG with images

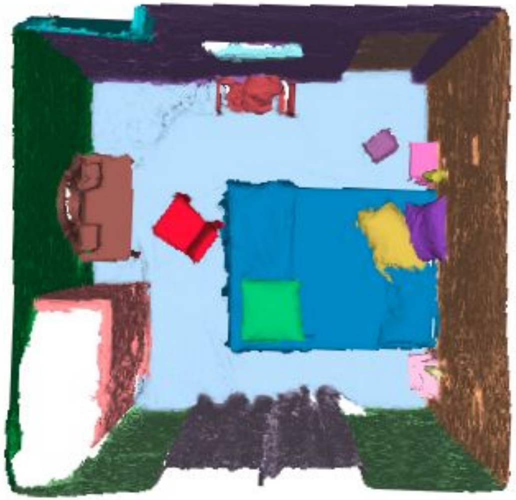
SSG with 3D scenes: inference

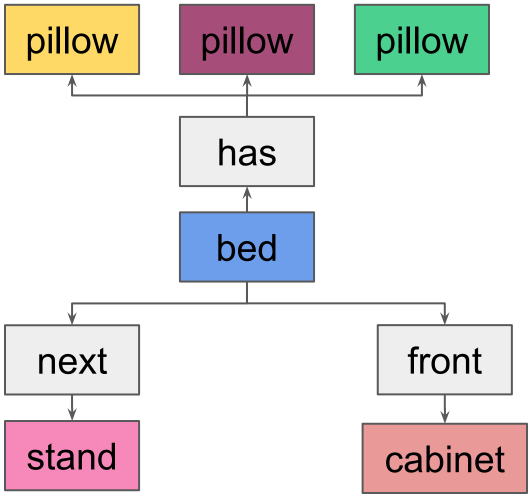
Definition of semantic scene graphs

3D Scene Representations

Text	3D Reconstruction	Scene Graph

A **Bedroom** consisting of a bed with **3** pillows and a desk with an office chair. **Next** to the bed there is a little nightstand and in **front** of the bad is a cabinet standing.





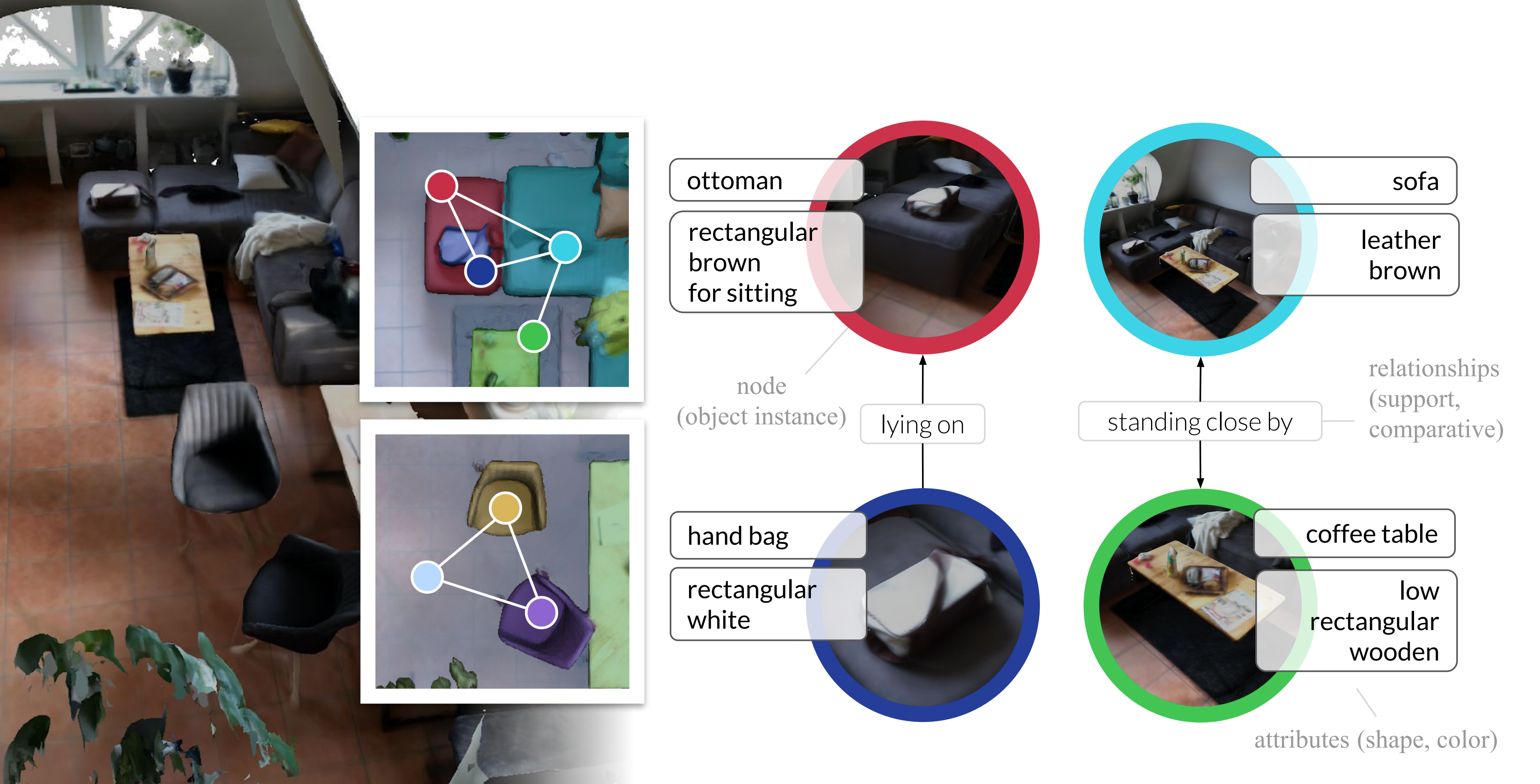
Semantic scene graphs

Graph that relates the components of a scene

- **Nodes**: objects
- **Edges**: relationships between objects
  - Action (holding, eating, riding, sitting, ...)
  - Proximity (near, left of, front of, above, ...)
  - Support (on, hanging on, ...)
  - Comparison (same as, smaller than, ...)
- **Attributes**: object properties
  - color, shape, material, ...

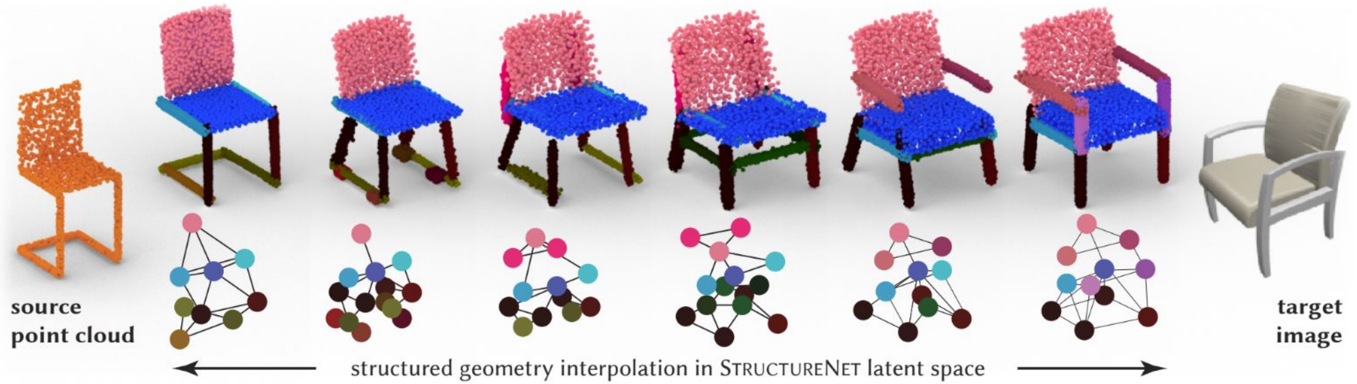
- **Recent trend:** enrich nodes with
  - learned features
  - geometric features (bounding box)

Semantic scene graphs in 3D



From 3D scenes to 3D objects/shapes

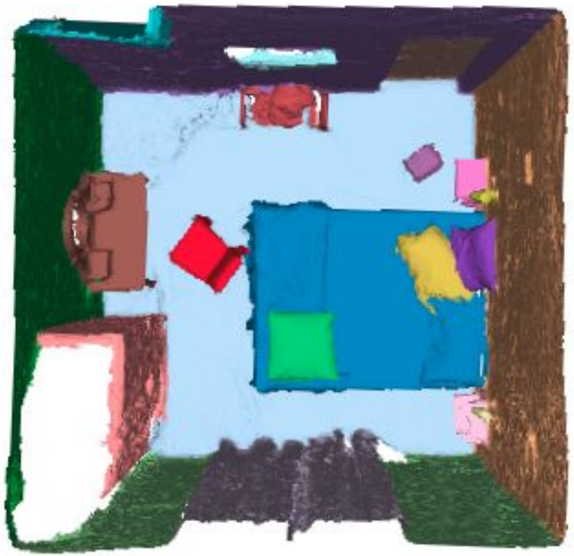
- Rather than representing components of a scene, a graph is associated with a single object or 3D shape
- Each node is typically a semantic component of the shape itself (e.g. armrests and legs for a chair), while edges can represent geometric adjacency or semantic relationships
- This can help generating different shapes from a given category in a semantically coherent way, or interpolate between two given shapes (e.g. for retrieval applications)

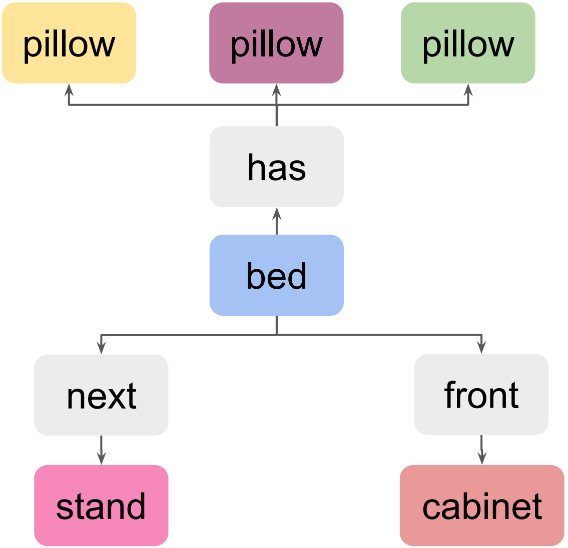


K. Mo et al, StructureNet: Hierarchical Graph Networks for 3D Shape Generation, Siggraph Asia 2019

3D Scenes

Semantic scene graphs





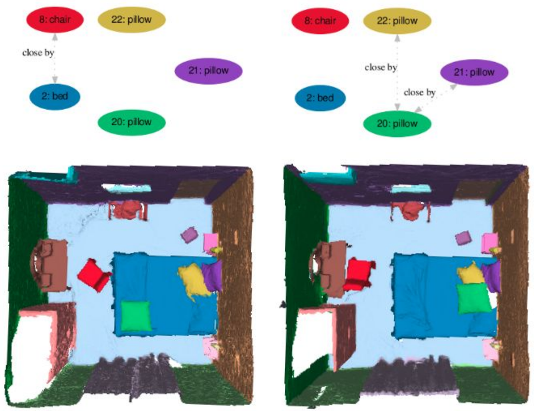
Memory Expensive

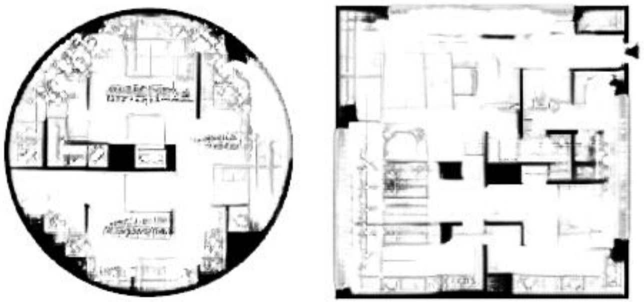
Typically Rigid

Memory Lightweight

Why scene graphs?

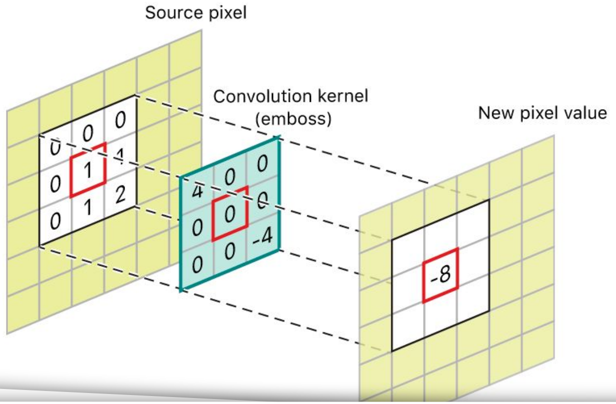
Inference	Generation
Multimodal scene retrieval	Synthetic data generation
Scene change detection	Indoor design / work placement
Compression	Scene editing / manipulation





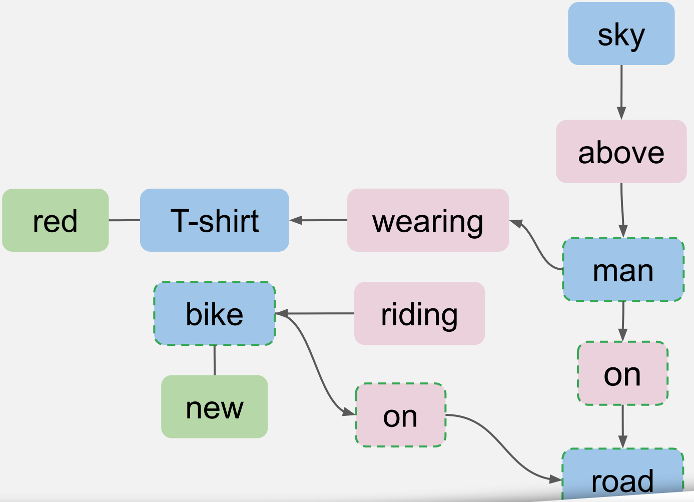
S. Chaillou, ArchiGAN: a Generative Stack for Apartment Building Design, nVidia, 2019

Scene Graph Processing



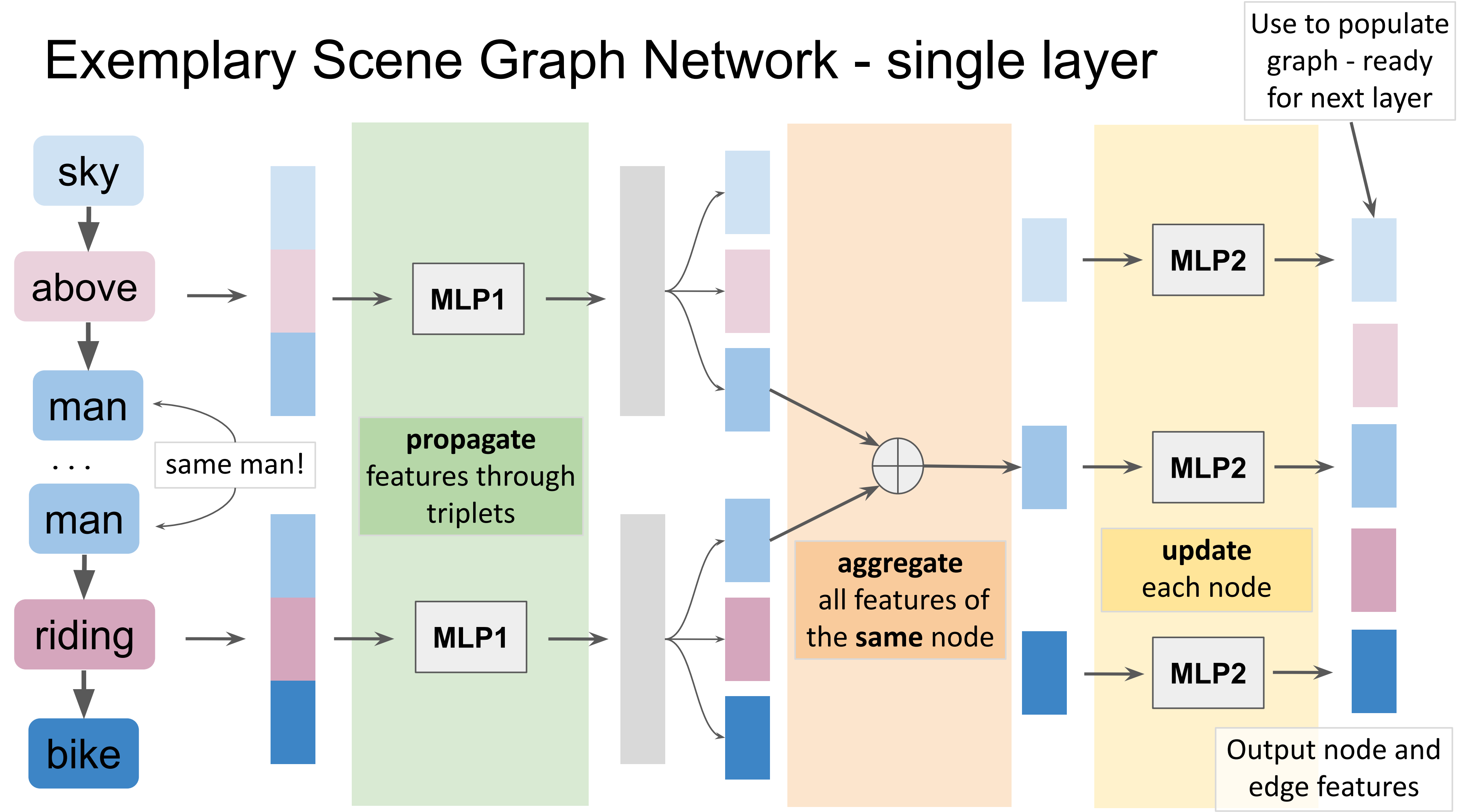
2D Convolutional Networks:

- Locality and Receptive Field
- Hierarchical Learning
- Parameter Sharing

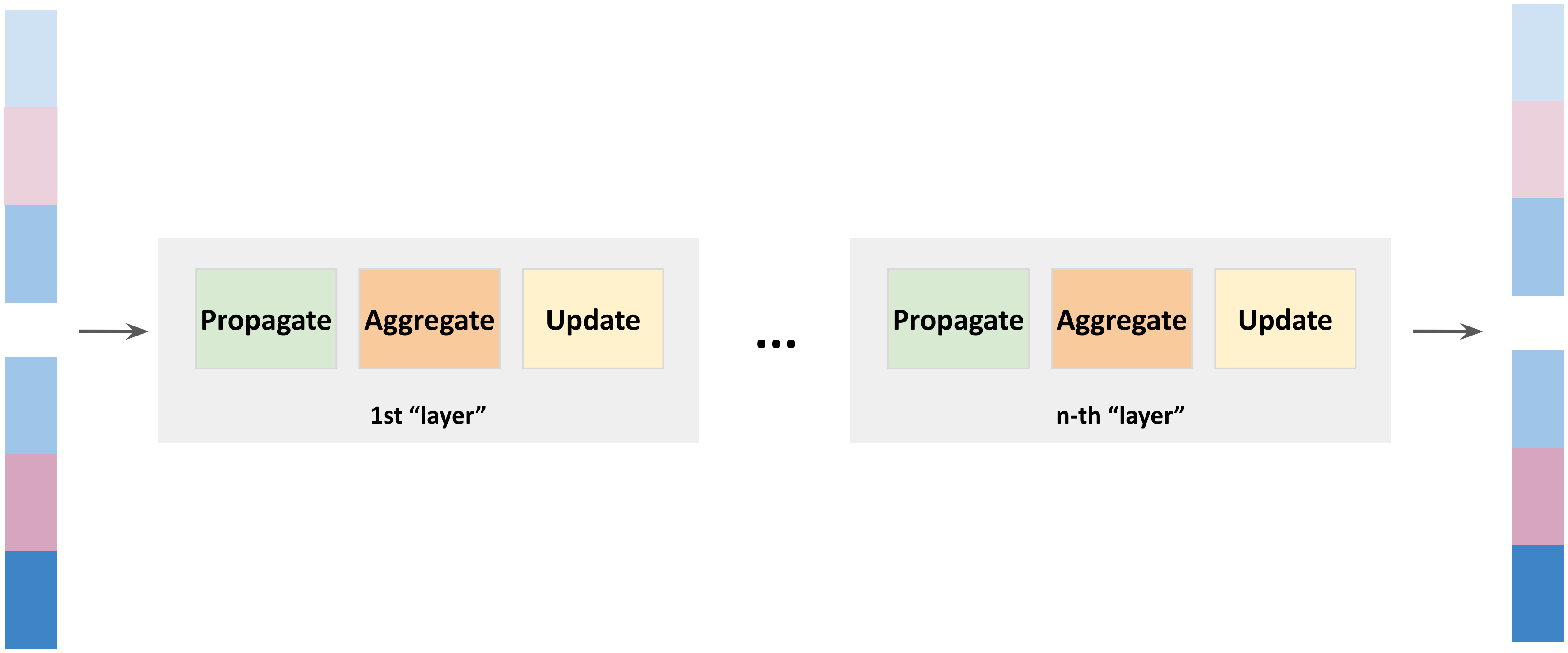


Can we bring these nice properties from 2D convolutions also to Scene Graphs?

Exemplary Scene Graph Network - single layer



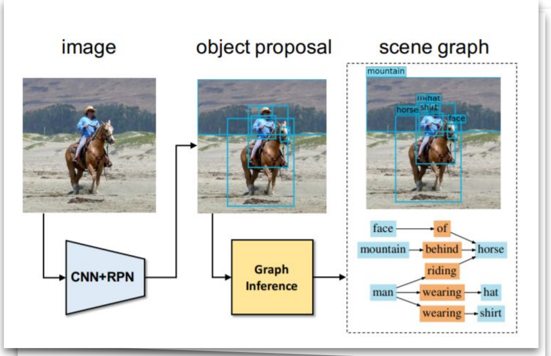
Scene Graph Network - general architecture



Semantic scene graphs in images

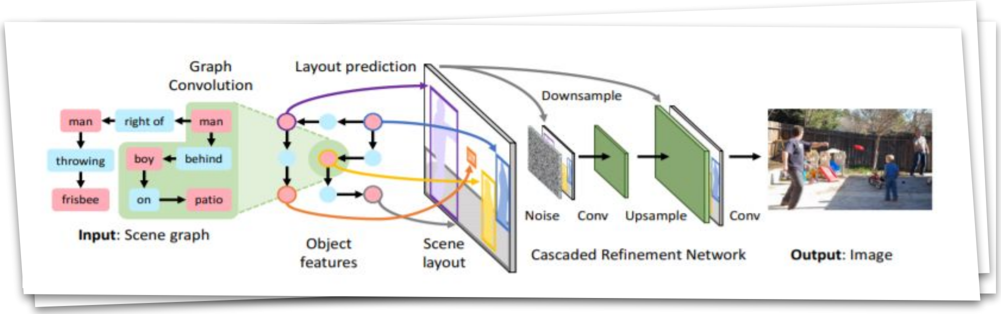
2D Semantic scene graphs

From **image** to **scene graph**



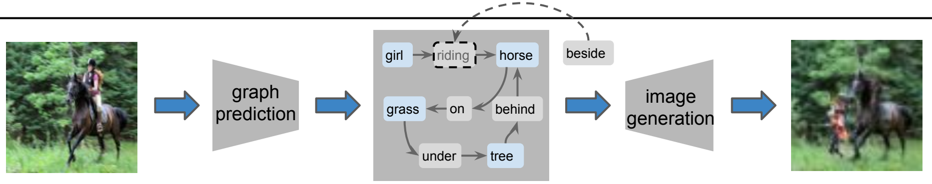
[Xu CVPR'17] Use region proposals to compute 3D scene graph via an iterative message passing network

From **scene graph** to **image**



[Johnson CVPR'18] Computes images from semantic graphs (object class)

[Ashual ICCV'19] Additionally employs visual features for objects



source image

interactive graph modification

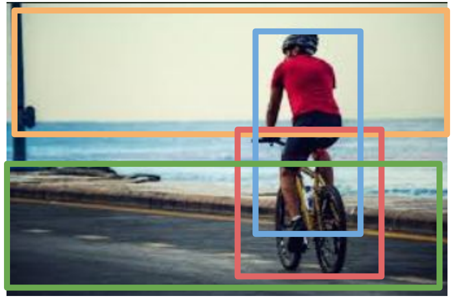
modified image

Closing the loop - From image to scene graph and back [Dhamo20]

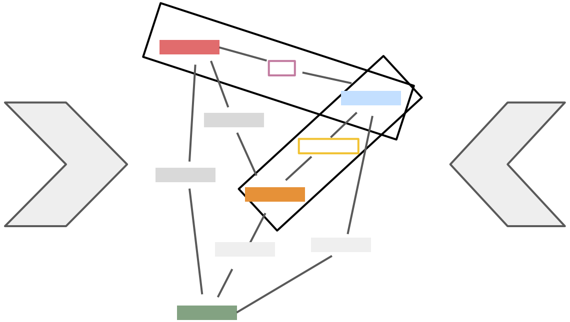
- Leveraging the predicted semantic scene graph as abstract input representation for the user
- User can apply changes which are then reflected in the input image

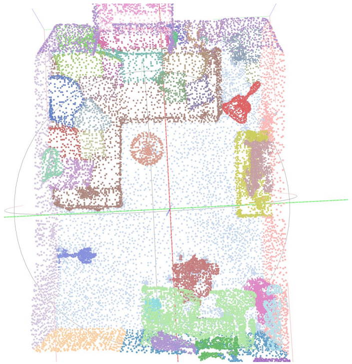
Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. ICCV 2019.
Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. CVPR 2018.
Danfei Xu, Yuke Zhu, Christopher Choi, Li Fei-Fei. Scene Graph Generation by Iterative Message Passing. CVPR 2017
H. Dhamo, A. Farshad, I. Laina, N. Navab, G. D. Hager, F. Tombari, C. Rupprecht. Semantic image manipulation using scene graphs. CVPR 2020

Can we use scene graphs as abstract representation for 3D scenes, similarly to what we do with images?



Image



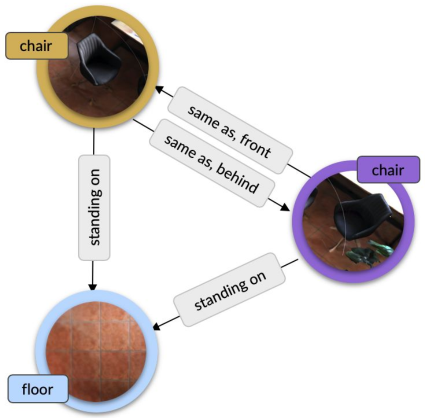


3D scene scan

Learning 3D Semantic Scene Graphs from 3D scenes

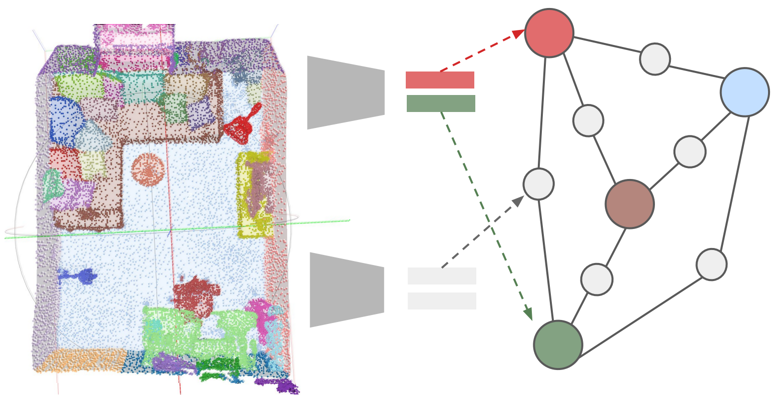
3DSSG Dataset

3D Semantic Scene Graph Dataset available at 3DSSG.github.io



3D Graph Prediction Network

Learned Method for Semantic Scene Graph Prediction based on PointNet and GCNs



Scene Retrieval

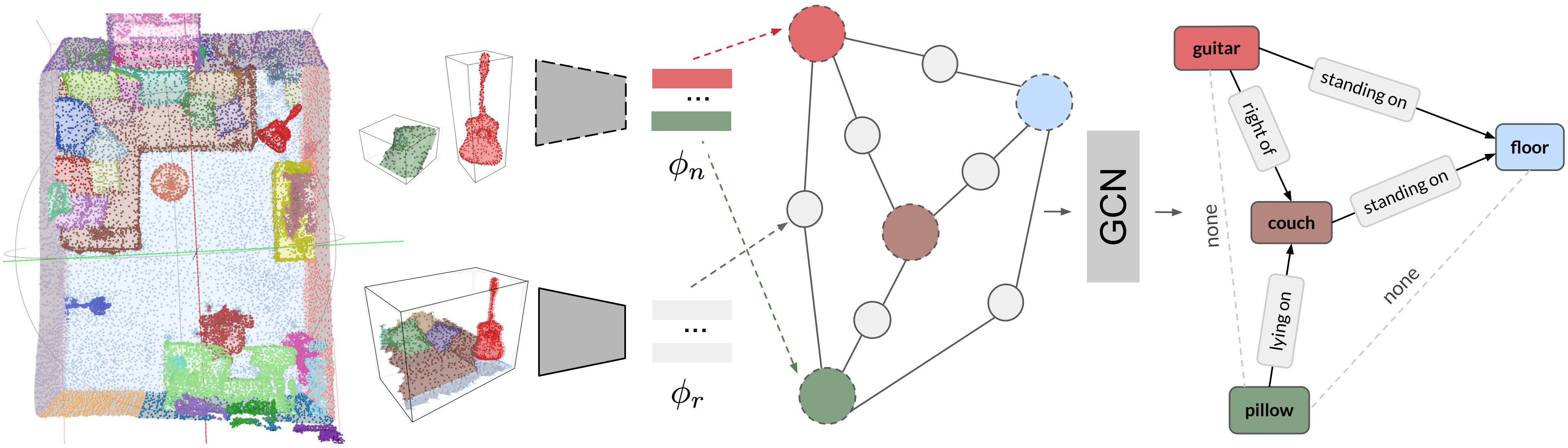
Application: 3D and 2D-3D Scene Retrieval in Changing Environments

J. Wald, H. Dhamo, N. Navab, F. Tombari, Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions, CVPR 2020


3DSSG
3D Semantic Scene Graphs

Learning 3D Semantic Scene Graphs

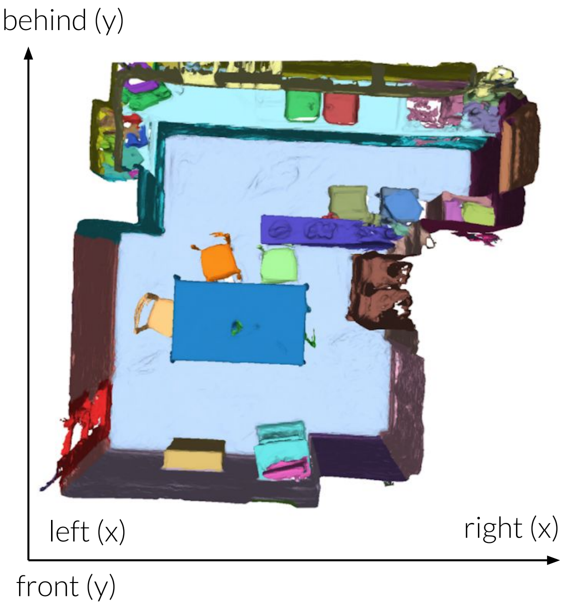
Input Point Set of a Scene | Fully-Connected Graph of Features | **Output** 3D Scene Graph

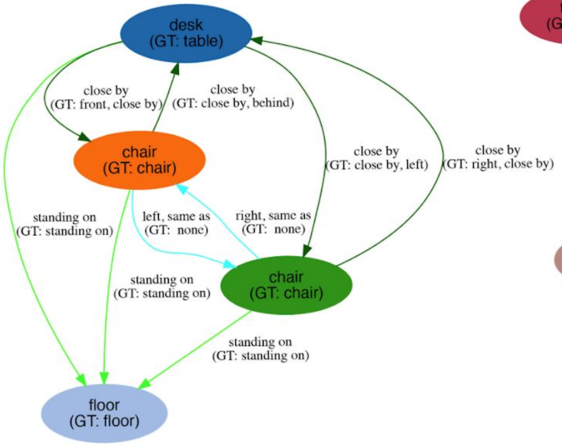


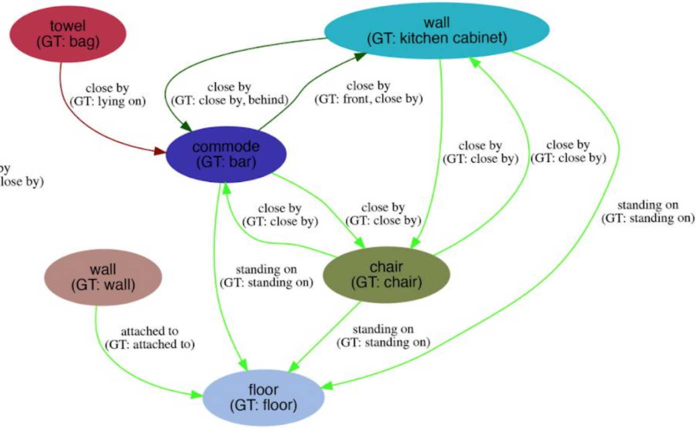
$$\mathcal{L}_{\text{total}} = \lambda_{obj}\mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{pred}} \qquad \mathcal{L} = -\alpha_t(1-p_t)^\gamma \log p_t$$

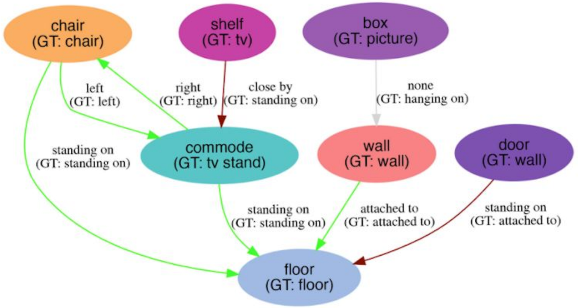
none or multiple predicate predictions per edge

Results Learning 3D Semantic Scene Graphs







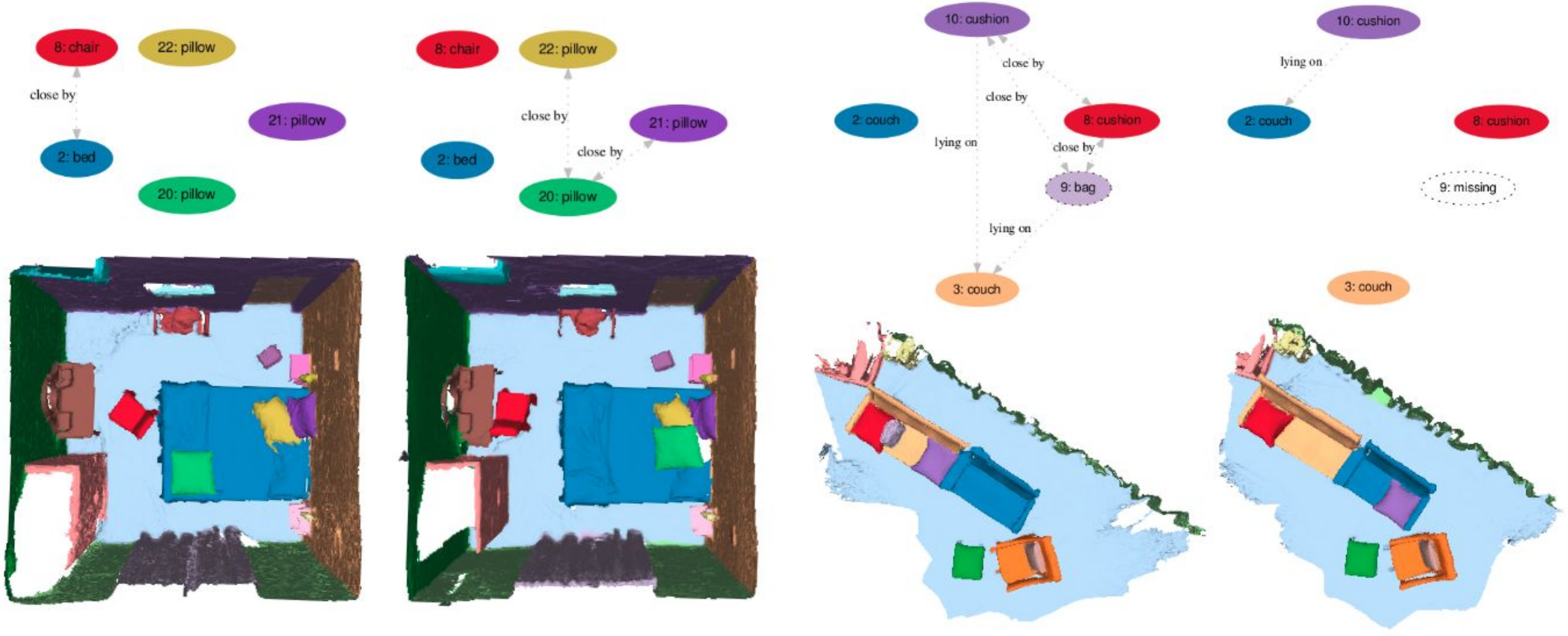


	Relationship Prediction		Object Class Prediction		Predicate Prediction	
Method	R@50	R@100	R@5	R@10	R@3	R@5
① Relation Prediction Baseline	0.39	0.45	0.66	0.77	0.62	0.88
Single Predicate, ObjCls from PointNet Features	0.37	0.43	**0.68**	**0.78**	0.42	0.58
② Multi Predicate, ObjCls from PointNet Features	**0.40**	**0.66**	**0.68**	**0.78**	**0.89**	**0.93**
Multi Predicate, ObjCls from GCN Features	0.30	0.60	0.60	0.73	0.79	0.91

2D-3D Scene Retrieval

Towards Persistent Scene Understanding: Graph-based Change Detection

A byproduct of scene retrieval is semantic change detection (including changed relationships and objects).



Project page (including dataset): https://3DSSG.github.io

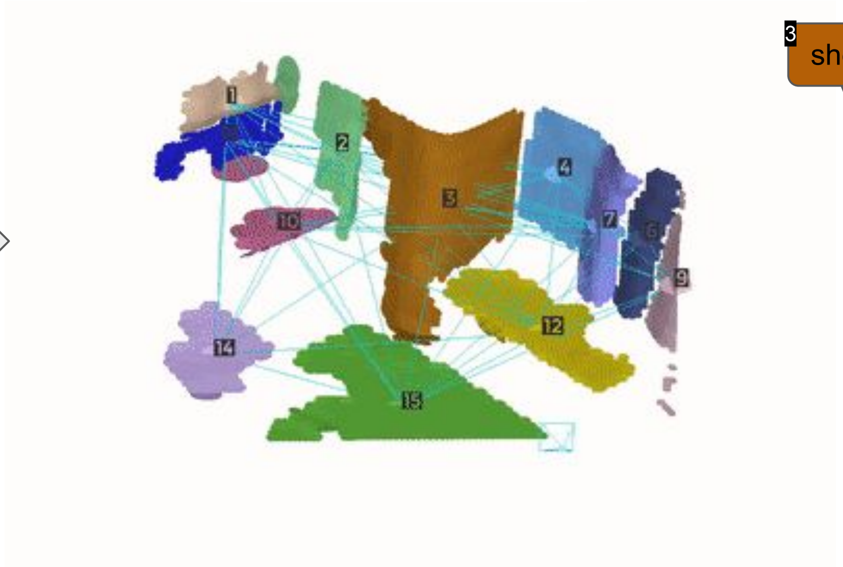
We have seen how scene graphs can be predicted offline from a full 3D scan/partial view of a scene

Can we instead predict them **incrementally** and **online**, as part of a SLAM pipeline?

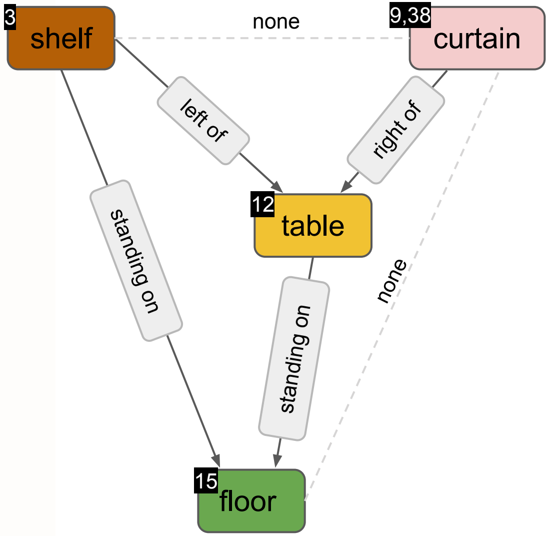
Online Semantic Scene Graphs

RGB-D Input

3D Scene Structure



3D Scene Graph



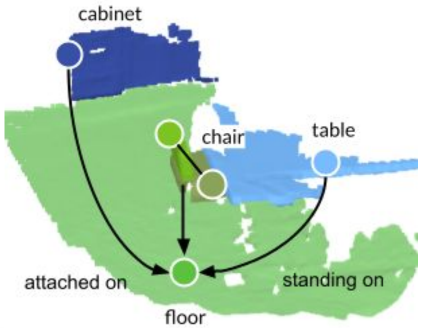
SceneGraphFusion: Incremental 3D Scene Graph Prediction from RGB-D Sequences

Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, Federico Tombari

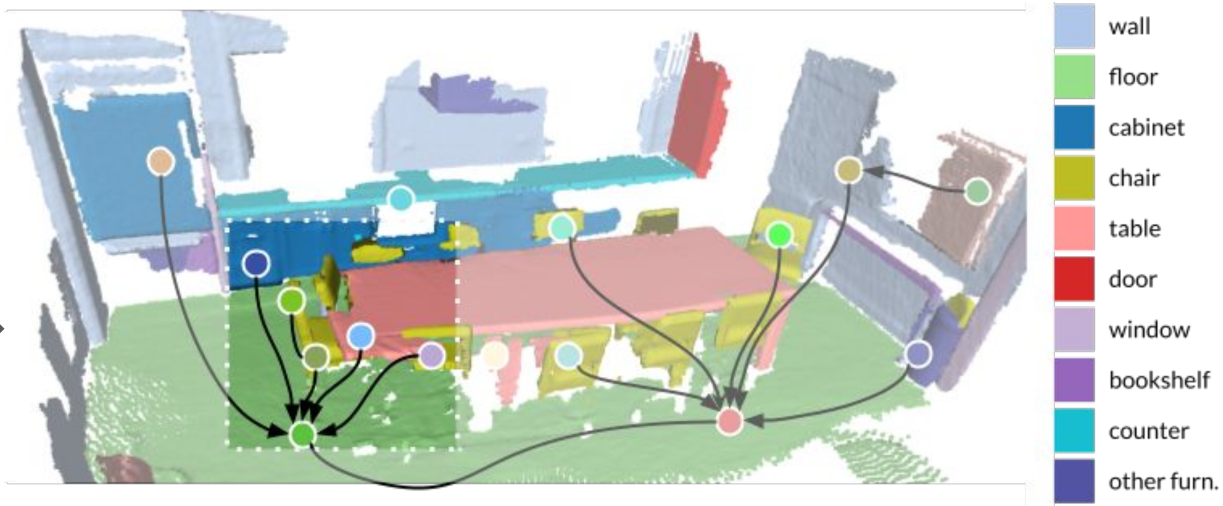
CVPR 2021

Input

Frame-wise Online Prediction



3D Semantic Reconstruction + Scene Graph



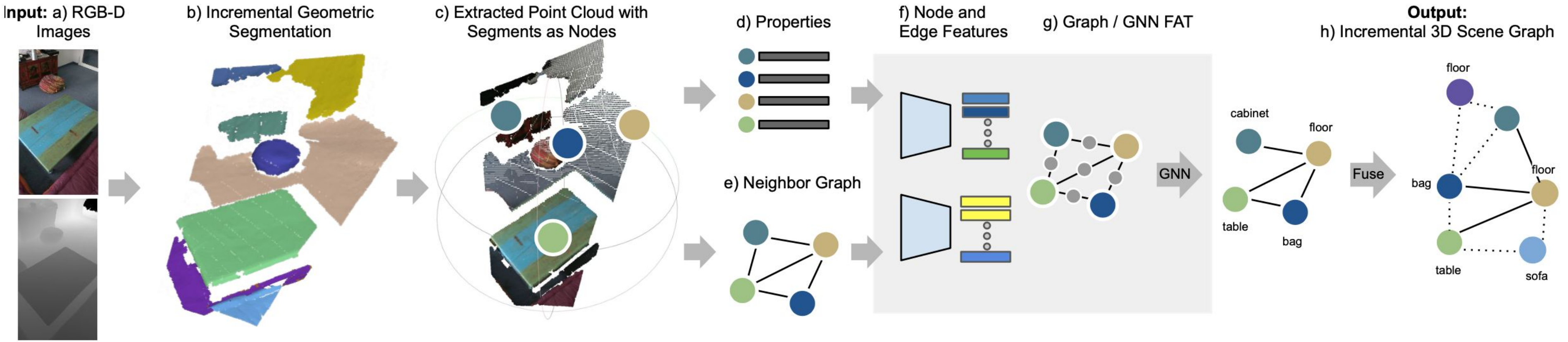
Semantic Scene Graphs model high-level semantics of objects and their relationships

Nodes: object classes

Edges: relationships between the connected nodes

Goal: incrementally predict the semantic scene graph with SLAM in real-time from an RGB-D sequence

Proposed framework



- Incrementally build globally consistent 3D geometric segmentation from RGB-D sequence using [1]
- Extracts neighbor segments in view
- Compute
  - a node feature on each segment using PointNet
  - an edge features on each pair based on stat indicators between the two segments (centroid distance, std of point cloud, ..)
- A GNN predicts object classes on each node and relationship on edges
- Each partial graph is **fused into a global 3D scene graph**

[1] Tateno, Keisuke, Federico Tombari, and Nassir Navab. "Real-time and scalable incremental segmentation on dense slam." *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015.

Results

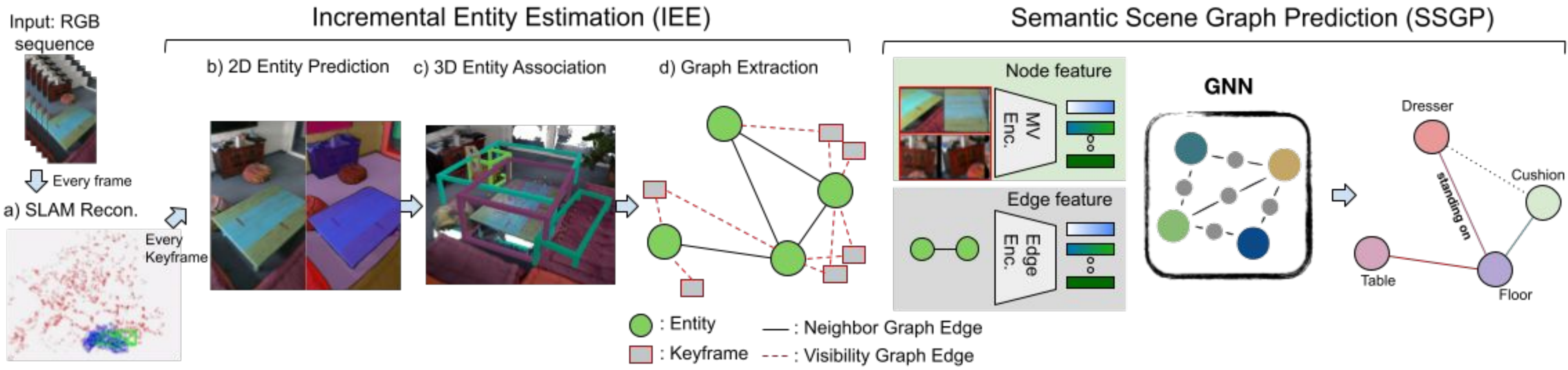
Input: RGB-D -> Monocular ?

Incremental 3D Semantic Scene Graph Prediction from RGB Sequences

Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, Federico Tombari

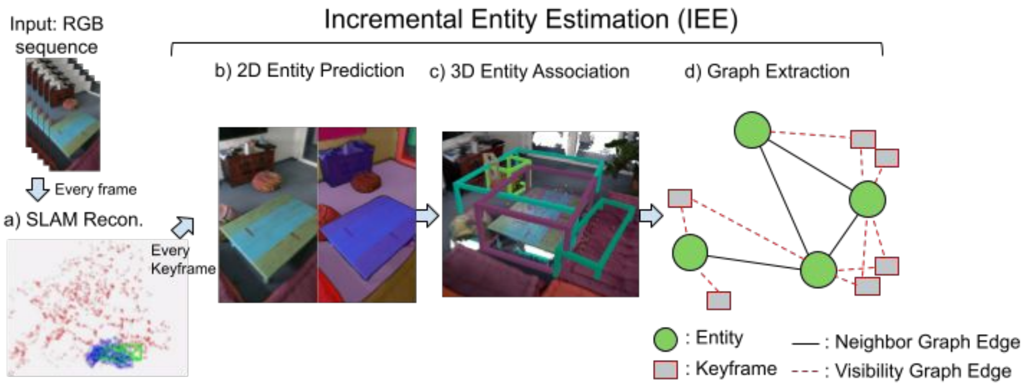
CVPR 2023

Objective: estimate the 3D scene graph incrementally from a **RGB sequence.**



1) Incremental Entity Estimation (IEE)

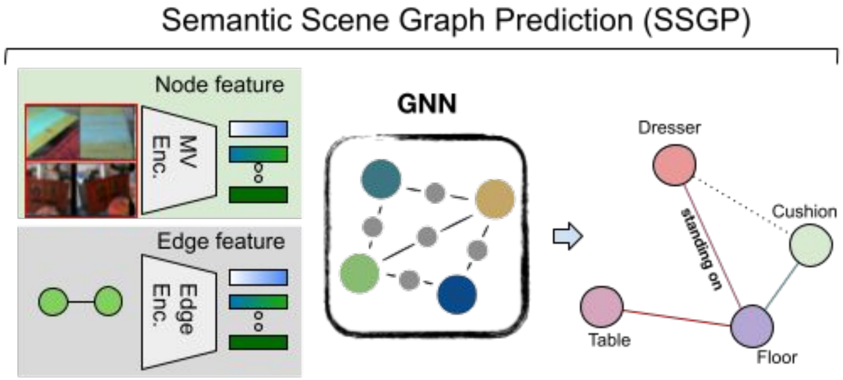
- a) Sparse Mapping: ORB-SLAM3 [1]
- b) 2D class agnostic instance segmentation via EntitySegmentation Network [2]
- c) Extract 3D bounding boxes via ApproxMVBB [3]
- d) Compute Neighbor Graph, where nodes are bounding boxes and connected to multiple keyframes



[1] Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., & Tardós, J. D. (2021). Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, *37*(6), 1874-1890.
[2] Qi, L., Kuen, J., Wang, Y., Gu, J., Zhao, H., Lin, Z., ... & Jia, J. (2021). Open-world entity segmentation. arXiv preprint arXiv:2107.14228.
[3] Chang, C. T., Gorissen, B., & Melchior, S. (2011). Fast oriented bounding box optimization on the rotation group SO (3, ℝ). *ACM Transactions on Graphics (TOG)*, *30*(5), 1-16.

2) Semantic Scene Graph Predictor (SSGP)

- Node Feature: MVCNN [1] with Res18 (concatenated features from the ROIs of all associated keyframes)
- Edge Feature: similar to SceneGraphFusion (various heuristics between two bboxes) + relative pose encoding
- GNN: same as in SceneGraphFusion



[1] Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 945-953).

Results

GT segmentation

InSeg from depth data

EntityNet from RGB data

	Method	Recall(%)			mIoU(%)		mPrec(%)		mRecall(%)	
		Rel.	Obj.	Pred.	Obj.	Pred.	Obj.	Pred.	Obj.	Pred.
GT	IMP [61]	8.1	31.6	95.6	19.7	24.3	42.8	43.8	30.6	28.2
	VGfM [15]	11.1	38.1	95.4	25.6	26.9	47.2	43.8	39.1	33.6
	Wald et al. [56]	26.5	52.3	91.3	28.1	19.4	39.0	30.4	47.6	28.3
	Wu et al. [59]	31.4	58.4	92.0	32.6	32.5	45.7	32.9	48.2	65.6
	Ours	**54.5**	**75.8**	**95.9**	**55.1**	**45.2**	**66.6**	**51.4**	**79.4**	**70.3**
Dense	IMP [61]	28.7	58.8	69.4	23.9	27.7	33.3	36.5	39.7	46.0
	VGfM [15]	39.9	**68.6**	73.4	37.5	31.9	48.5	41.8	57.2	49.6
	Wald et al. [56]	18.2	42.2	**93.4**	19.4	23.9	33.8	37.7	33.0	28.3
	Wu et al. [59]	39.3	67.3	82.8	**41.7**	31.1	52.1	34.7	**59.3**	**61.6**
	Ours	**42.2**	67.9	89.6	41.3	**37.1**	**52.9**	**43.9**	59.1	56.7
Sparse	IMP [61]	26.8	52.9	72.2	23.1	18.2	33.3	26.9	45.0	31.4
	VGfM [15]	29.9	57.6	74.3	26.6	24.0	39.1	31.5	41.9	46.7
	Wald et al. [56]	12.3	31.0	**81.6**	9.1	21.4	17.4	31.9	16.8	34.9
	Wu et al. [59]	13.6	35.9	81.5	6.3	12.9	9.38	30.7	10.8	15.2
	Ours	29.5	58.0	80.4	30.4	**27.0**	40.1	**38.7**	52.9	**51.3**
	Ours (i)	**31.2**	**59.0**	80.6	**30.6**	26.4	**41.8**	37.9	**54.9**	50.5

Table 2. Evaluation of scene graph prediction task on 3RScan/3DSSG [56] with 20 objects and 8 predicate classes. We evaluate all methods on 3RScan with different input types.

Incremental 3D Semantic Scene Graph Prediction from RGB Sequences

ECCV22-166

Thanks for the attention!

Credits and collaborators (possibly incomplete):

Francis Engelmann (Google, ETH)

Helisa Dhamo (TUM)

Abhijit Kundu (Google)

Diego Martin Arroyo (Google)

Ferjad Naeem (Google, ETH)

Nassir Navab (TUM)

Michael Niemeyer (Google)

Evin Pinar Oernek (TUM)

Fabian Manhardt (Google)

Marie-Julie Rakotosaona (Google)

David J. Tan (Google)

Prune Truong (ETH)

Keisuke Tateno (Google)

Luc Van Gool (ETH)

Johanna Wald (Google)

Shun-Cheng Wu (TUM)

We offer internships/research position and research collaborations across Google, ETH and TUM around these topics.