# Foundation Models Meet 3D Vision

*Toward Open-World 3D Scene Understanding and Controllable 3D Generation*

**Francis Engelmann** PostDoc Stanford

Guest Lecture CS231A | June 4th, 2025

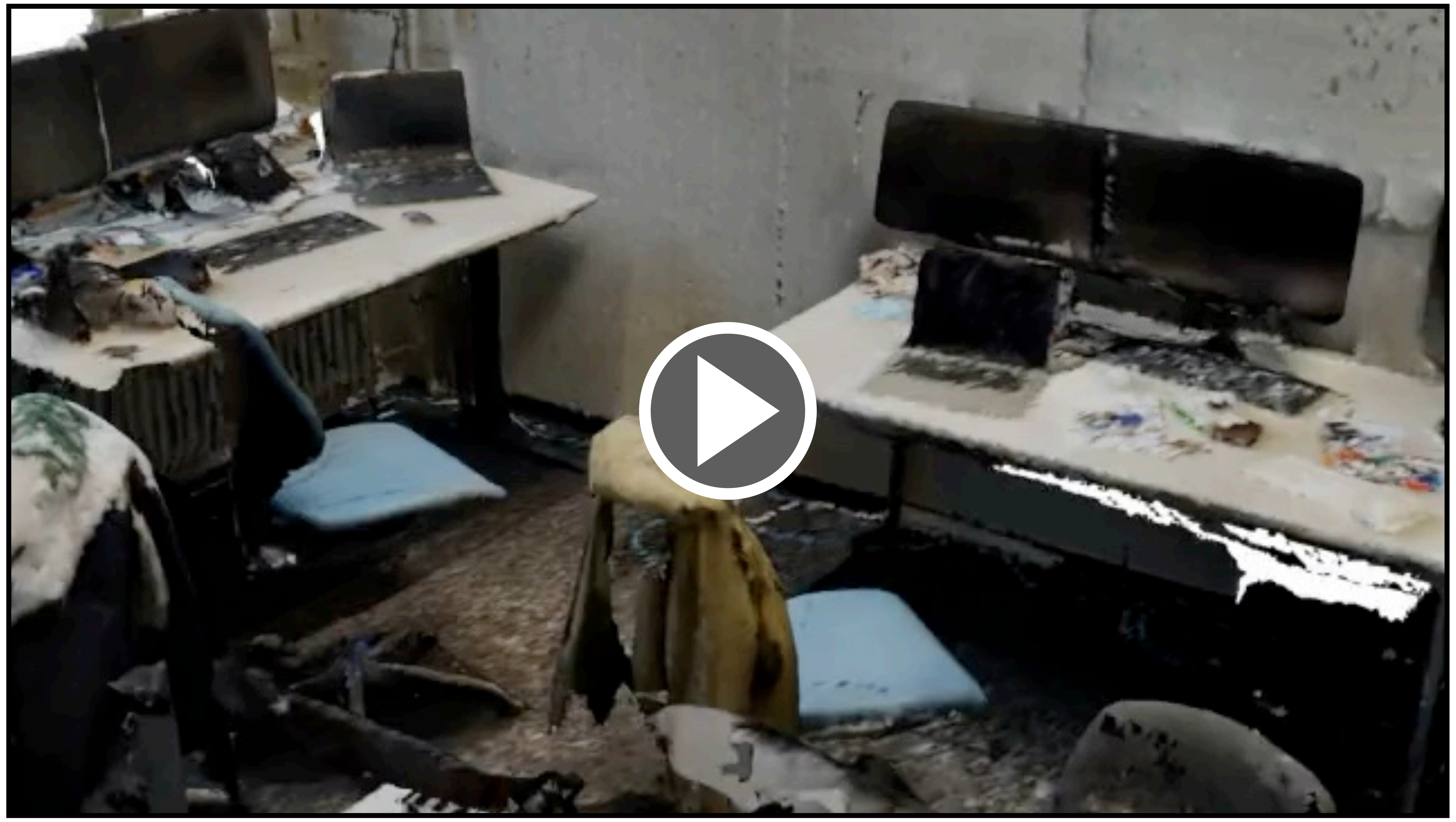
# *Toward Open-World 3D Scene Understanding and Controllable 3D Generation*

# 3D Scene Understanding

# What is 3D Scene Understanding?

Input: 3D scan of a scene...

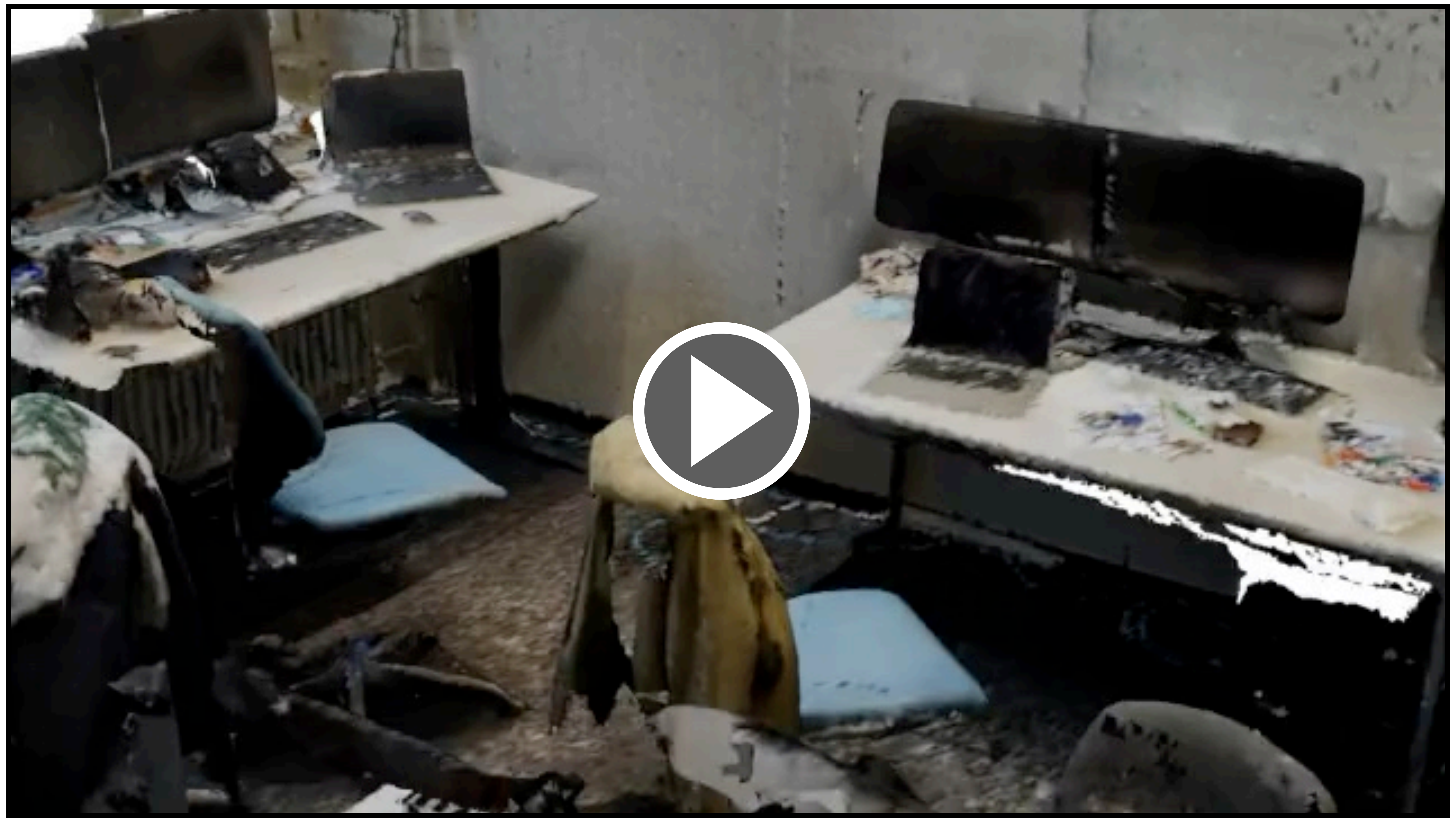
Mobile 3D Scanner



3D Scan / Reconstruction

# What is 3D Scene Understanding?

**Exemplary Task:** 3D Semantic Instance Segmentation



Input: **3D Scan**

Mobile Scanner

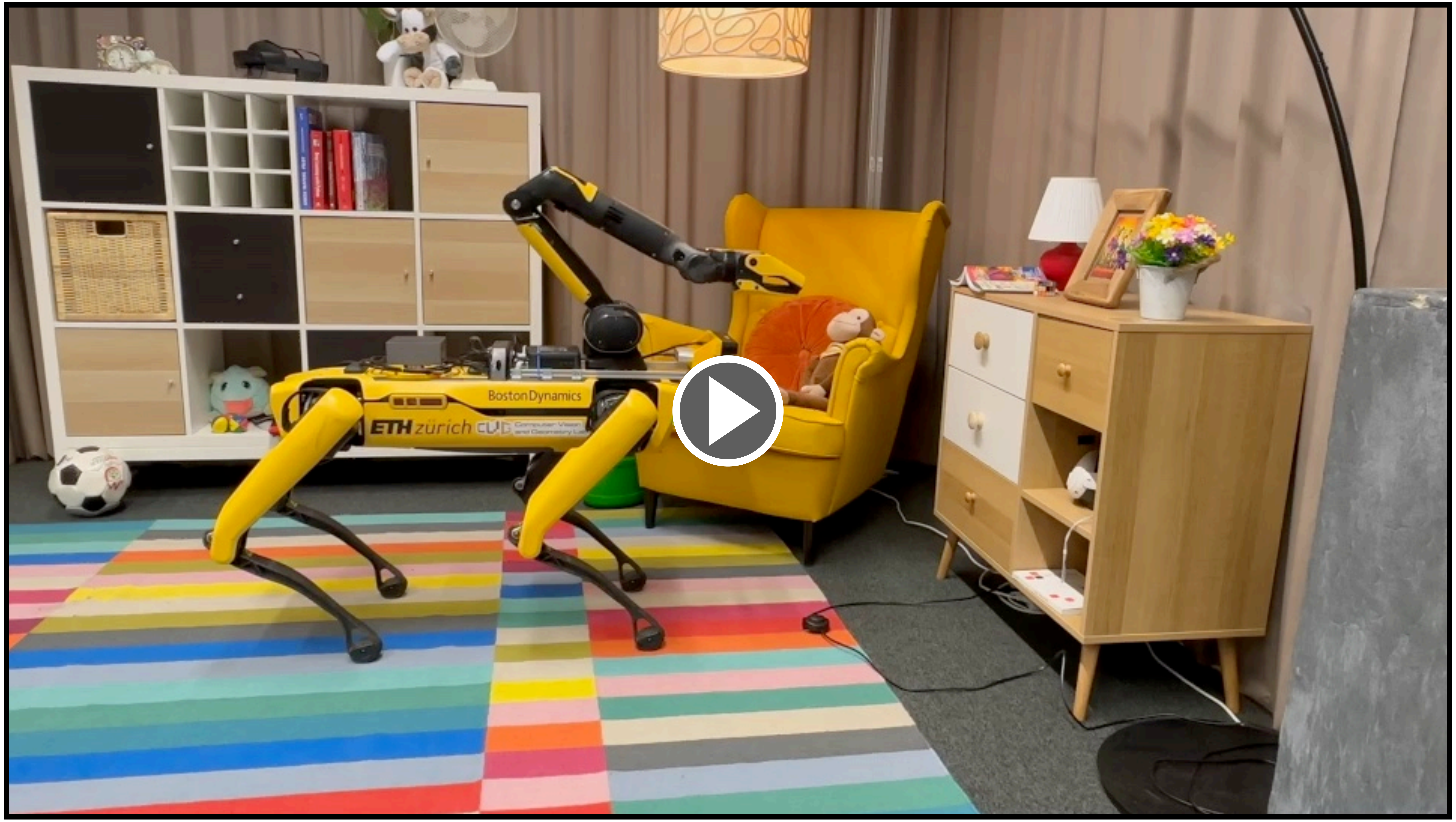
Output: **Semantic Instance Masks**



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

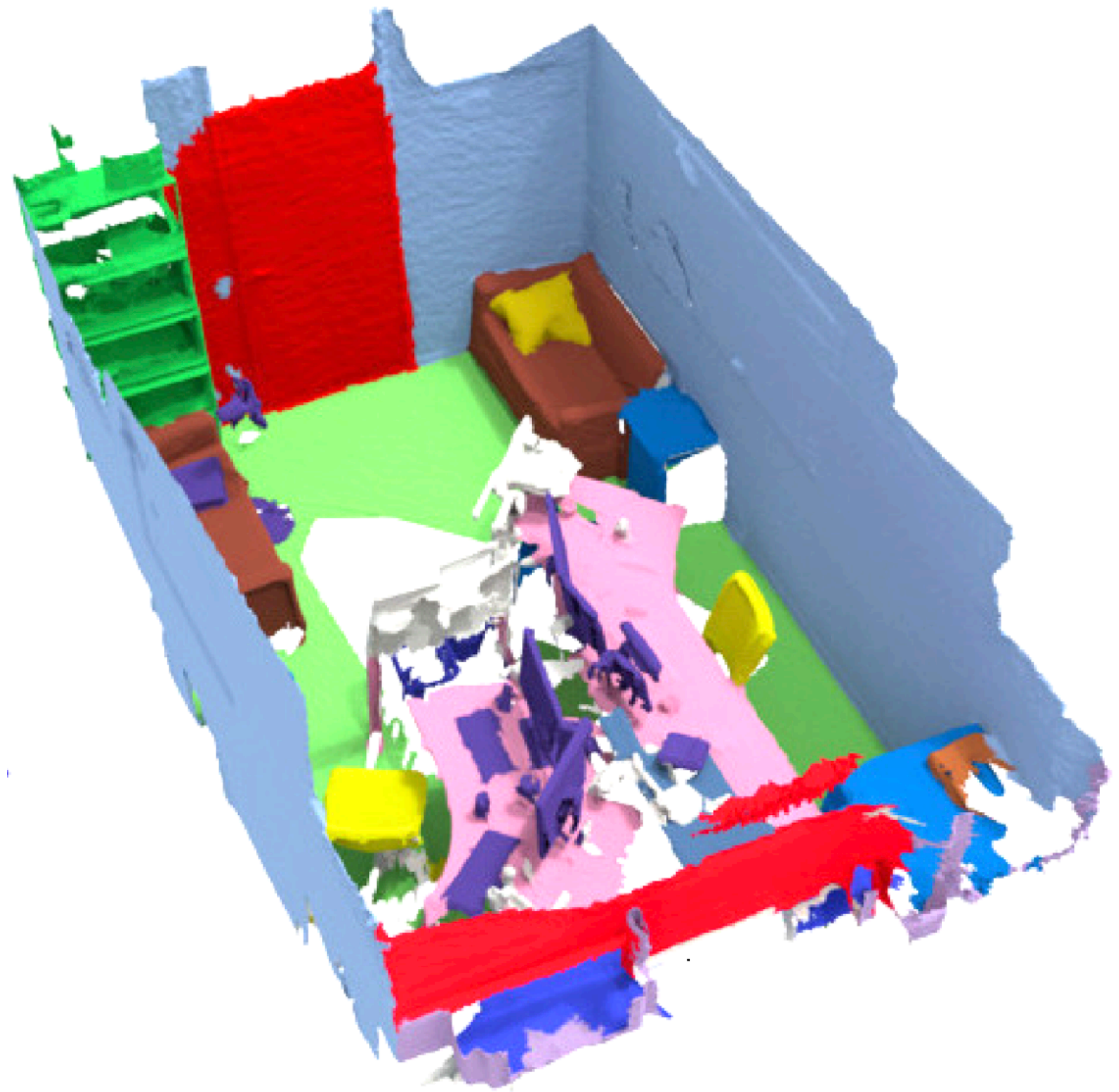
# What is 3D Scene Understanding?

**Towards human-centric AI:** e.g., Household robots making our lives easier



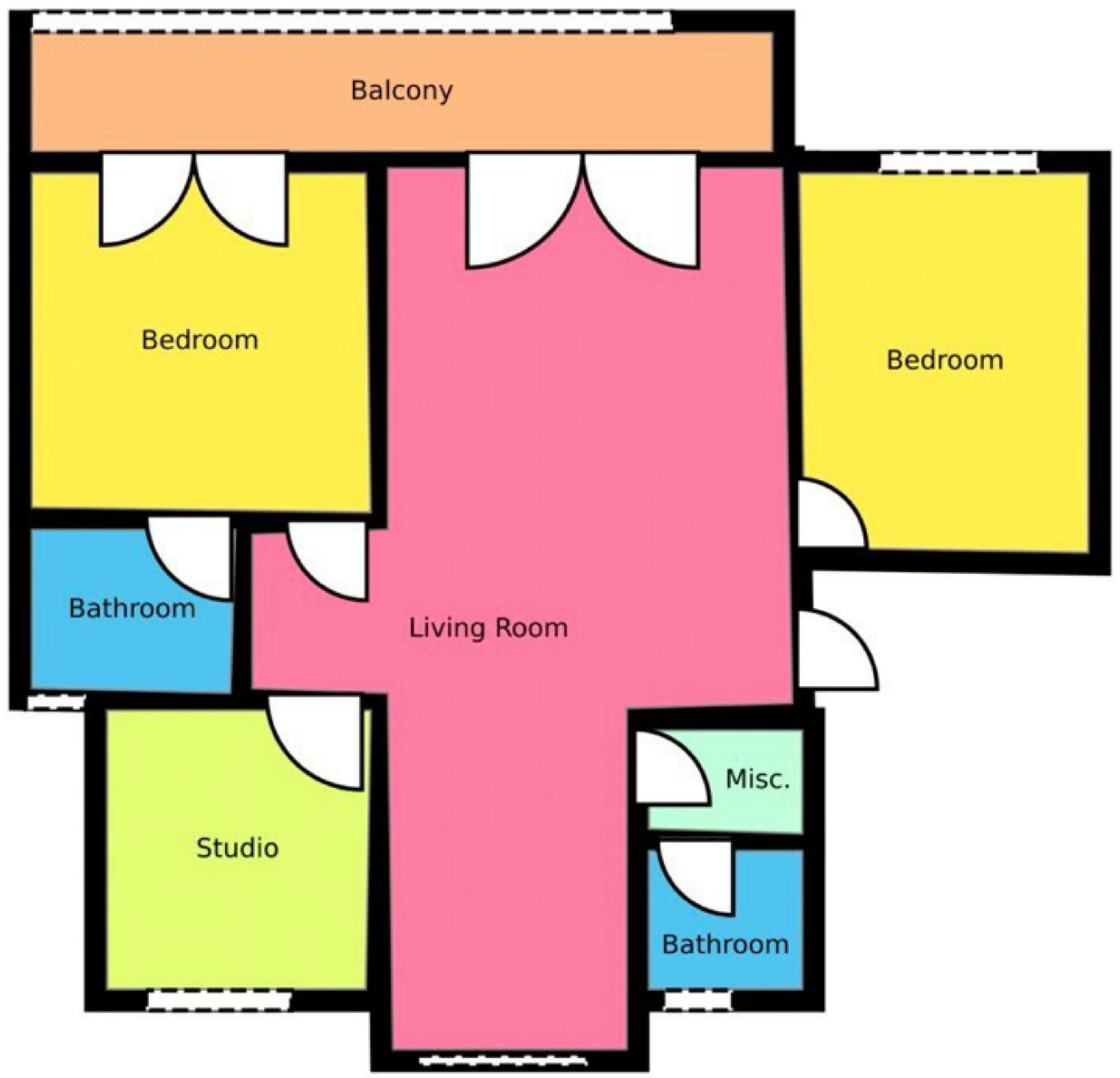
# 3D Scene Understanding

**Tasks:** From an input 3D scan, we predict ...



**3D Scene Segmentation**

*"Object instances"*




**Vectorized Floorplans**

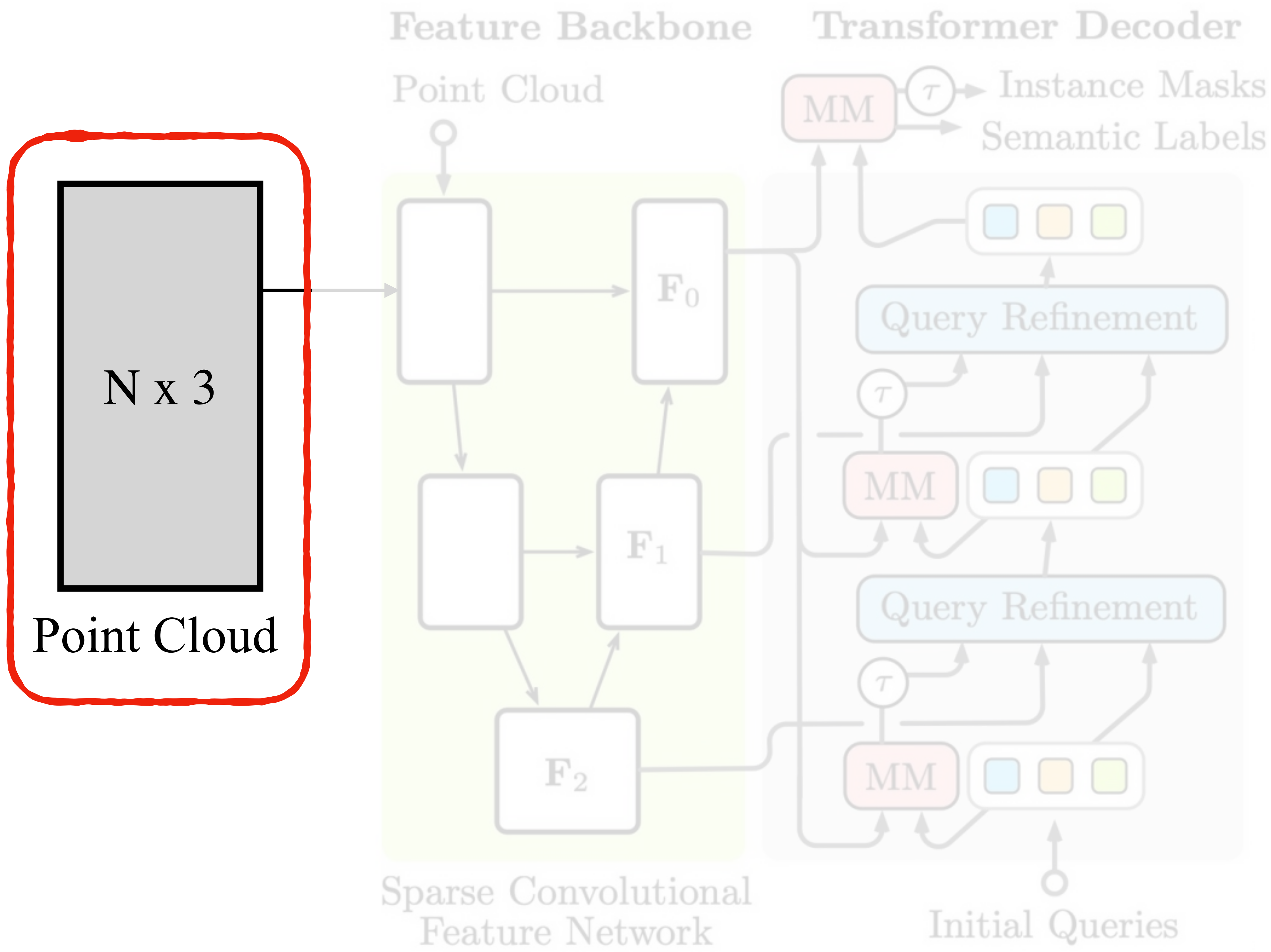
*"Structural elements"*

**Human Part Segmentation**

*"Human-scene interactions"*

# 3D Semantic Instance Segmentation

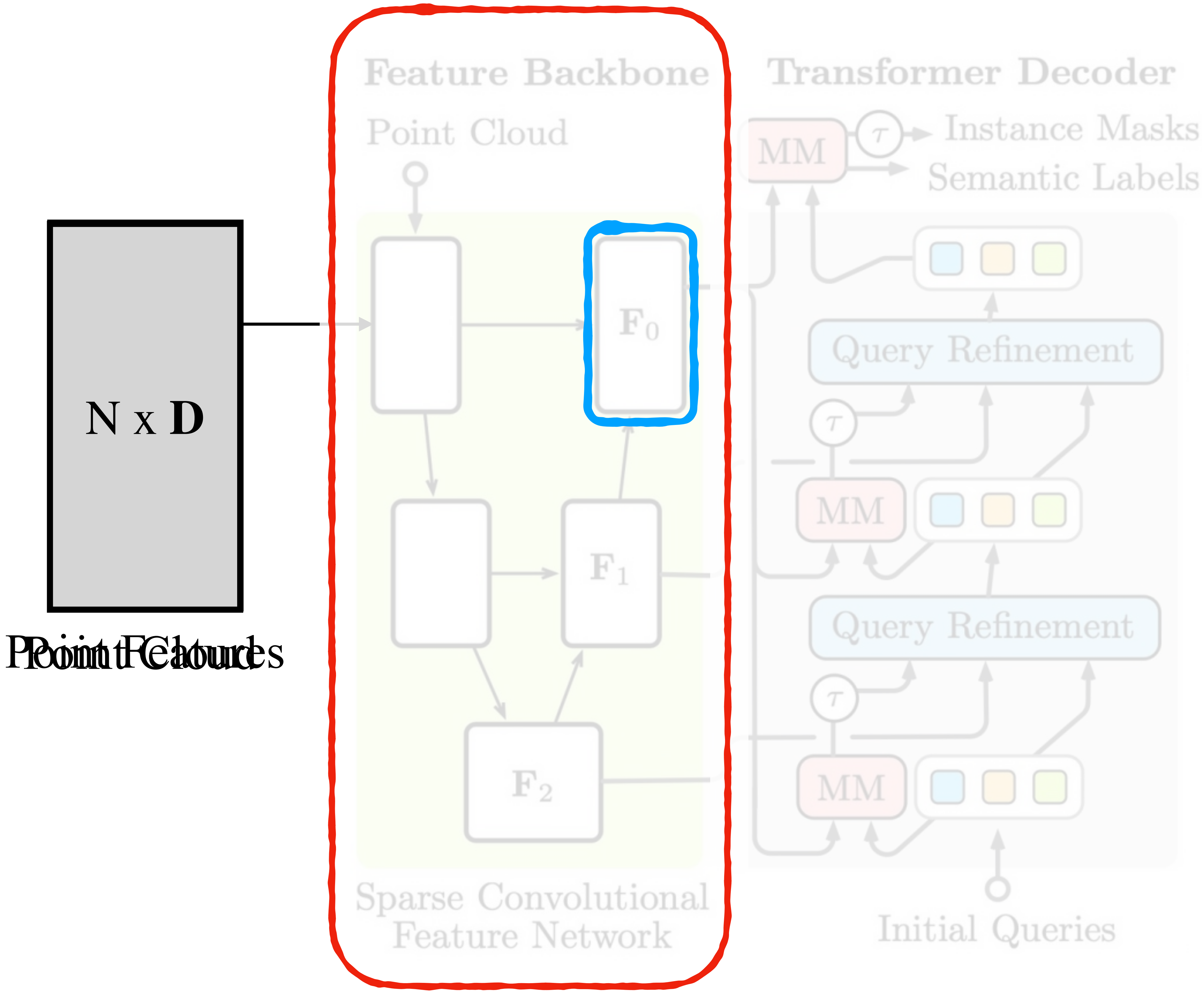
## Mask Transformer for 3D Instance Segmentation [1]




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

# 3D Semantic Instance Segmentation

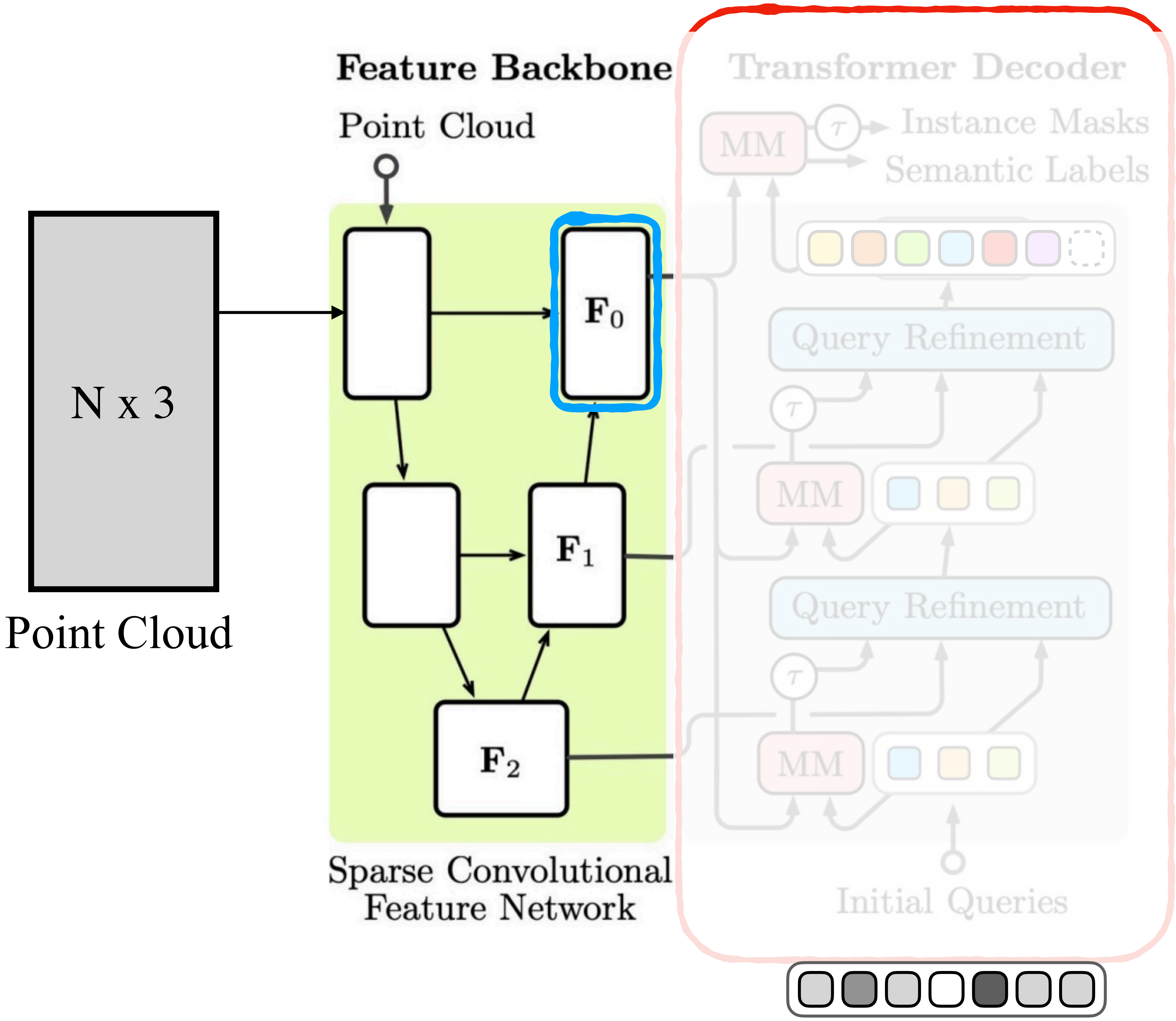
## Mask Transformer for 3D Instance Segmentation [1]




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

# 3D Semantic Instance Segmentation

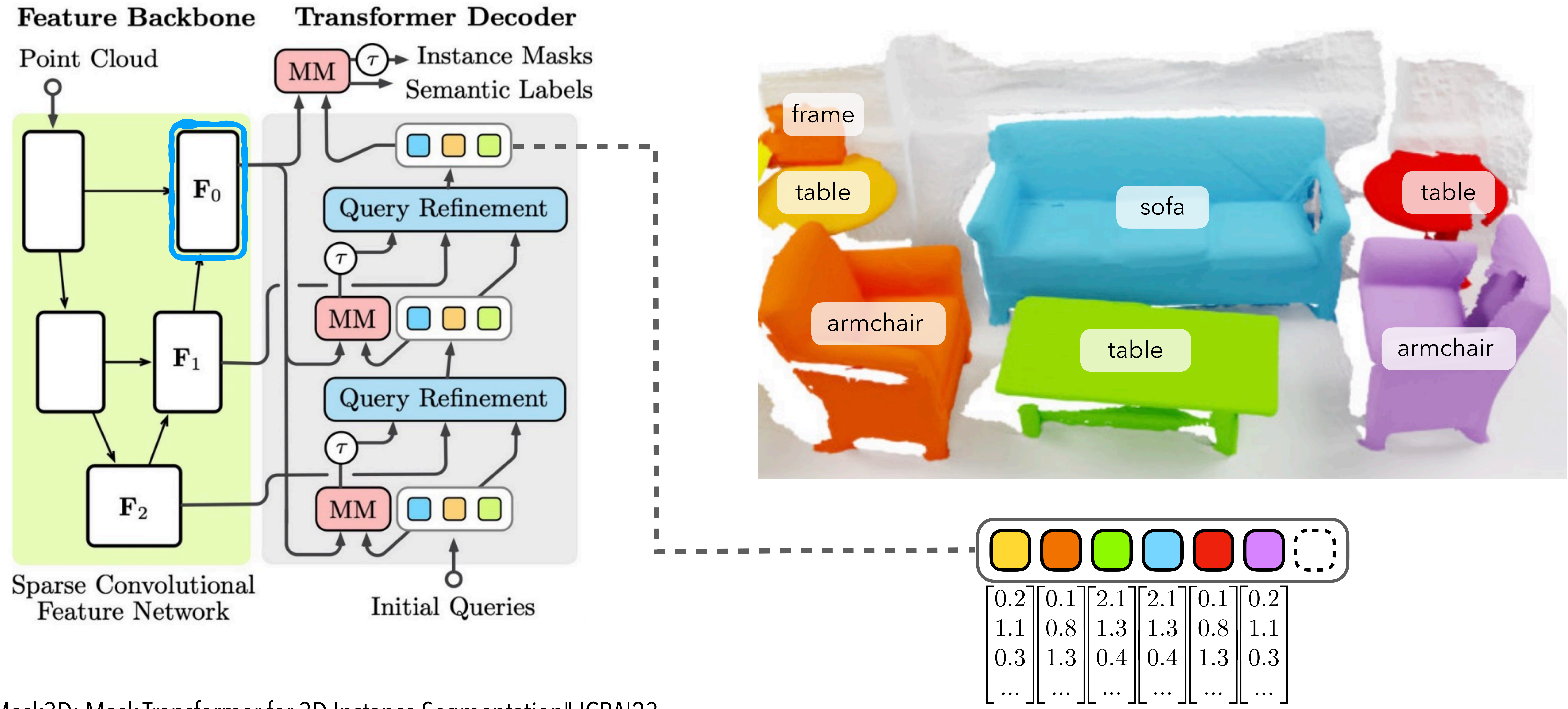
Mask Transformer for 3D Instance Segmentation [1]




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

# 3D Semantic Instance Segmentation

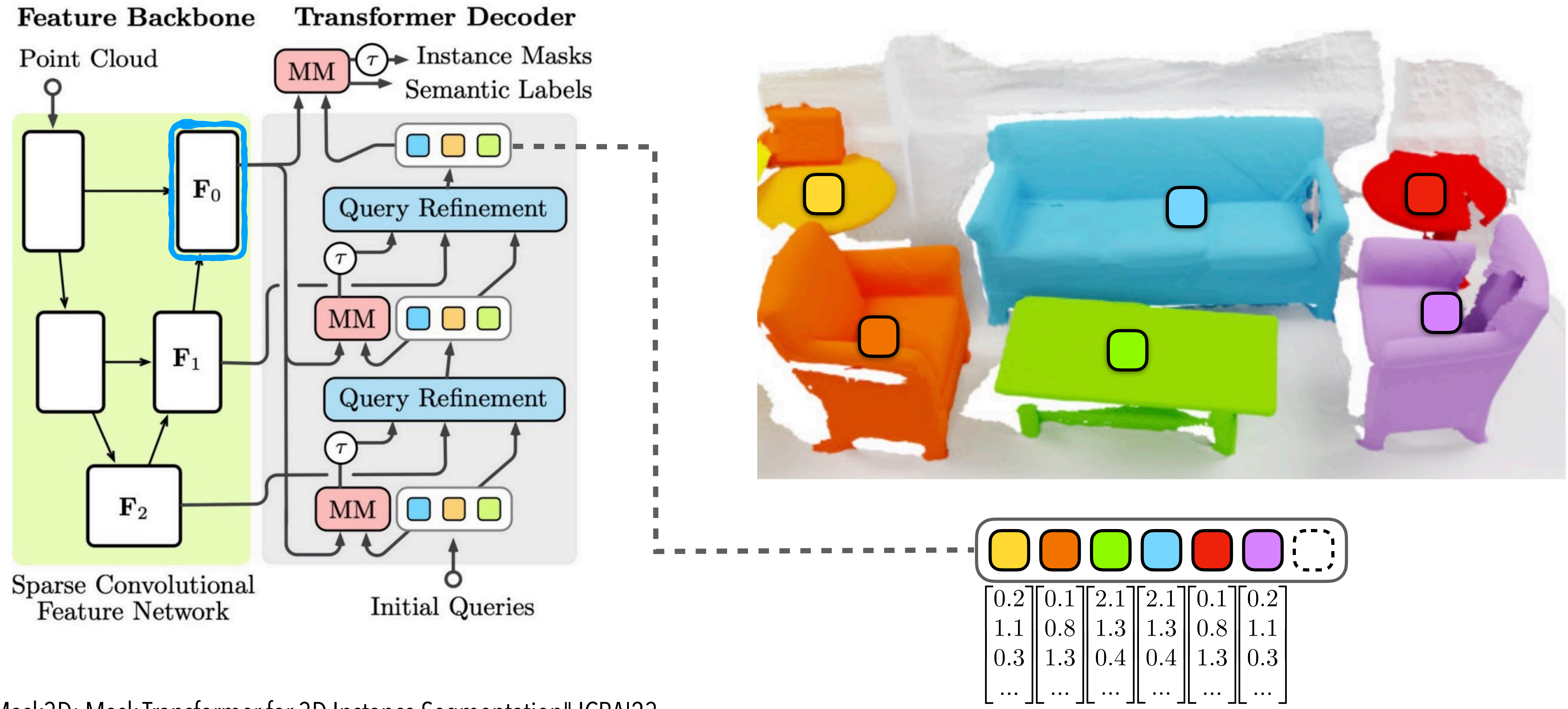
## Mask Transformer for 3D Instance Segmentation [1]




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

# 3D Semantic Instance Segmentation

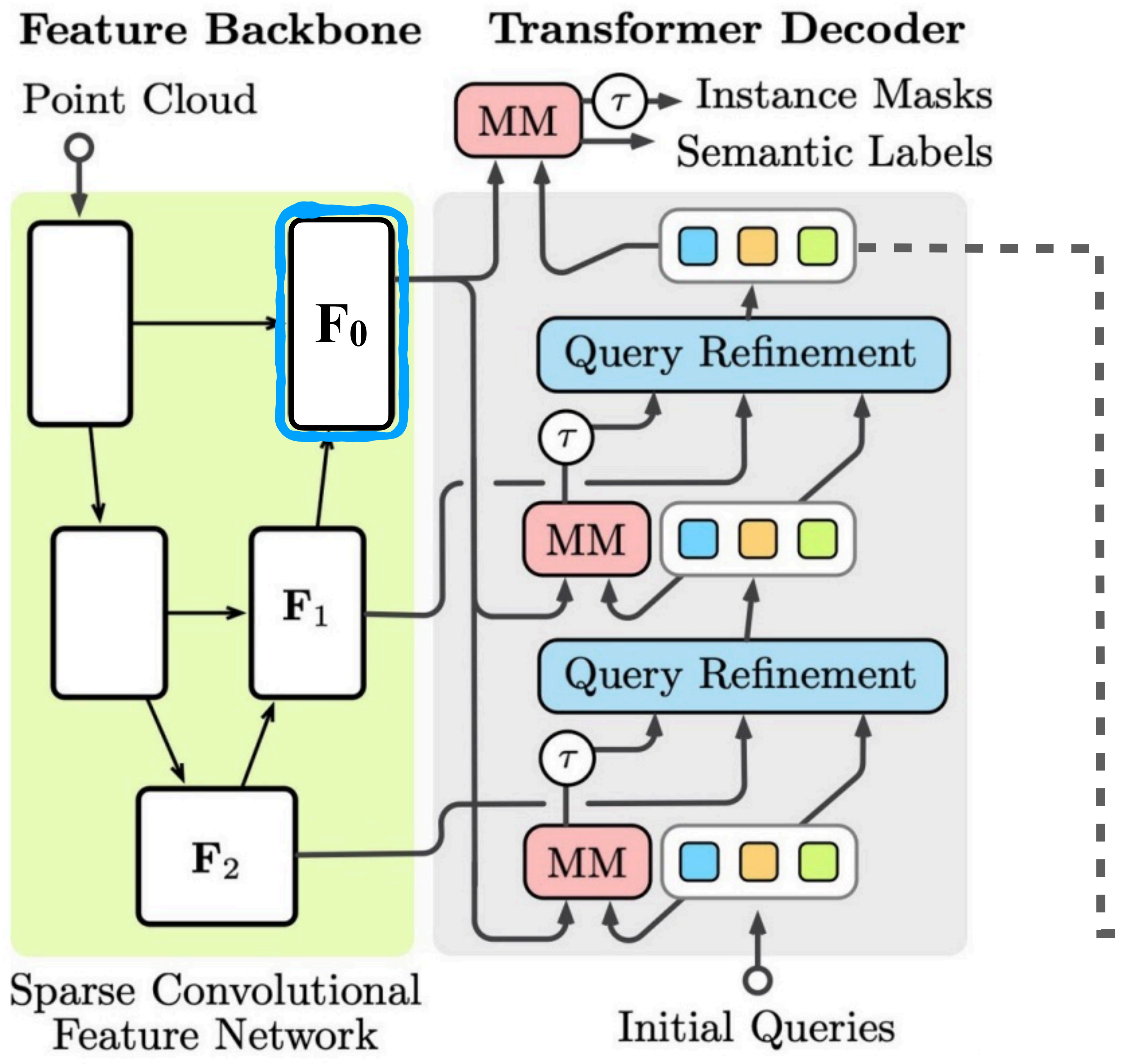
## Mask Transformer for 3D Instance Segmentation [1]

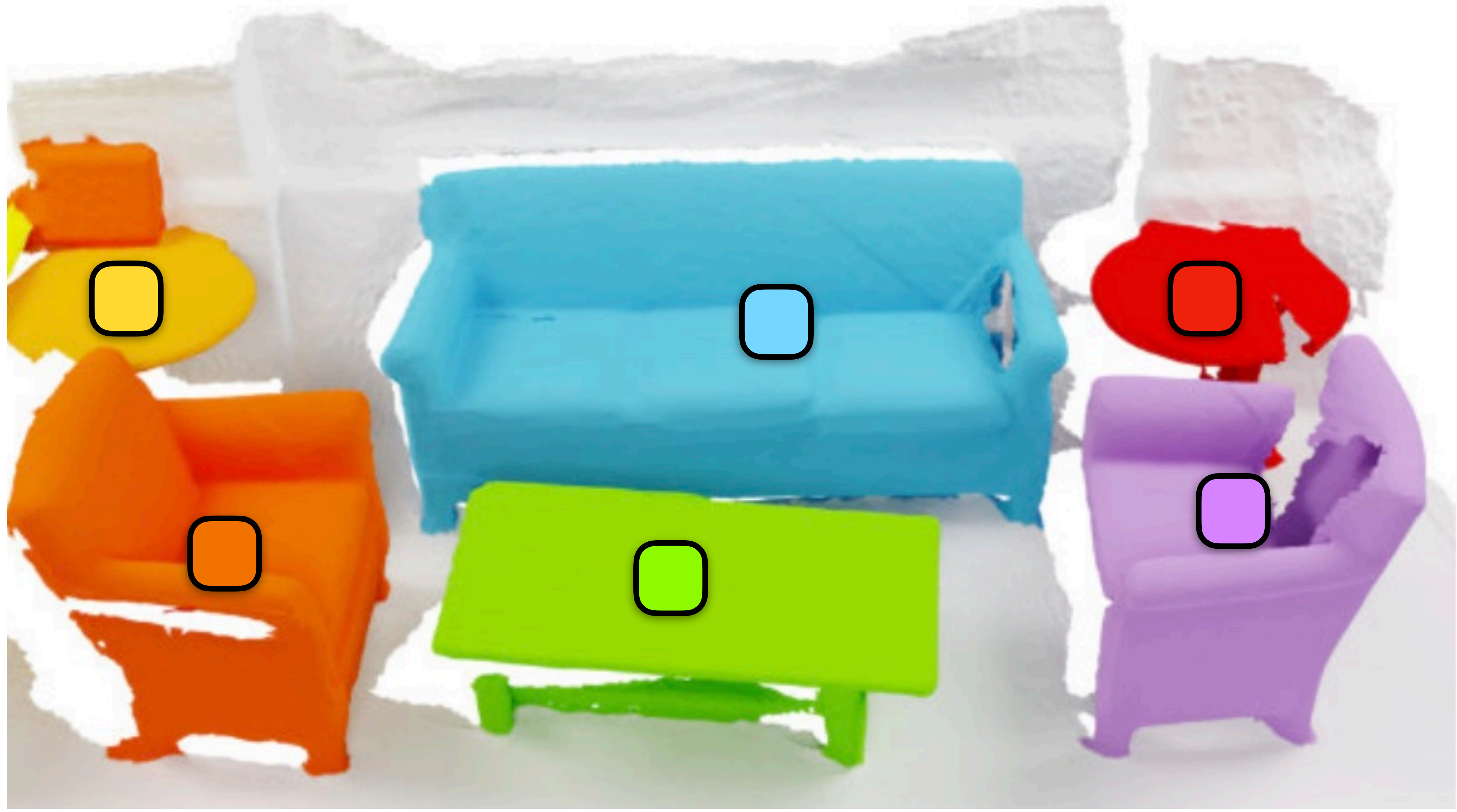


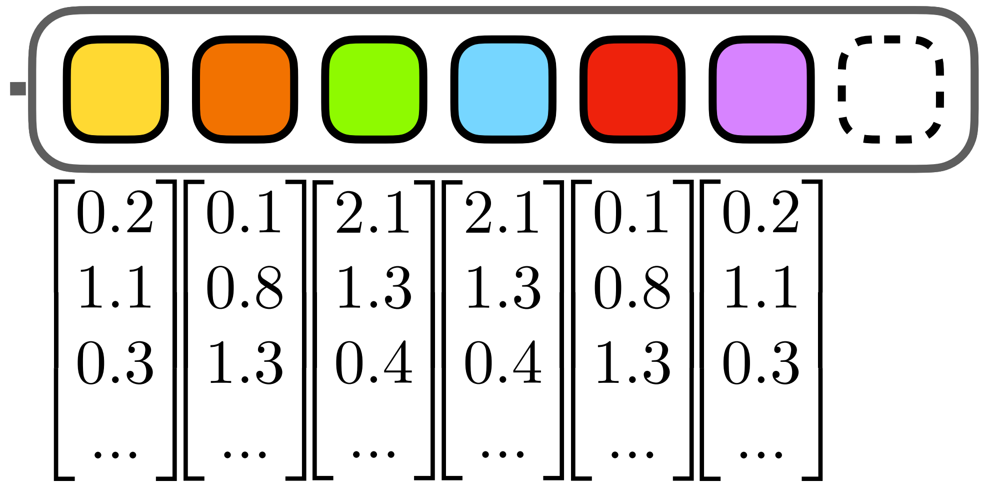

[1] Schult et al. "Mask3D: Mask Transformer for 3D Instance Segmentation" ICRA'23

# 3D Semantic Instance Segmentation

## Mask Transformer for 3D Instance Segmentation [1]



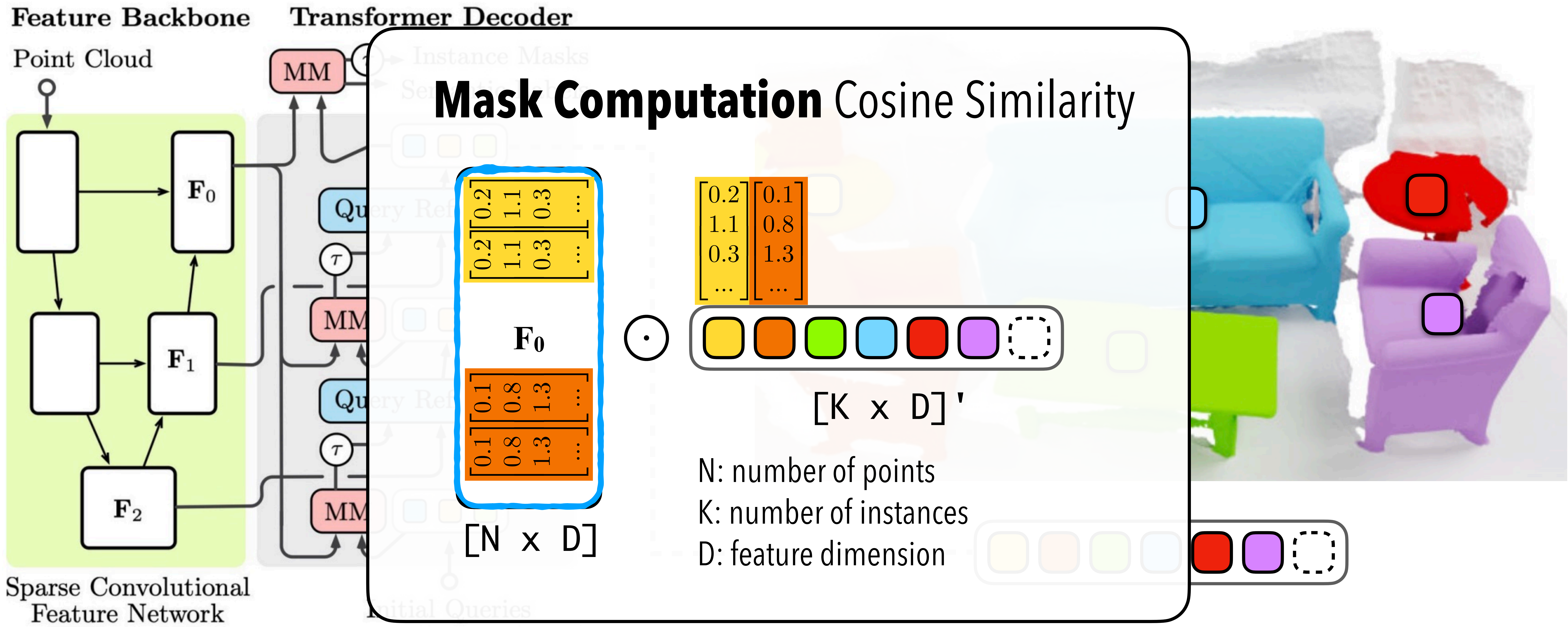






[1] Schult et al. "Mask3D: Mask Transformer for 3D Instance Segmentation" ICRA'23

# 3D Semantic Instance Segmentation

## Mask Transformer for 3D Instance Segmentation [1]




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

# 3D Semantic Instance Segmentation

Mask3D: Mask Transformer for 3D Instance Segmentation

Online Demo

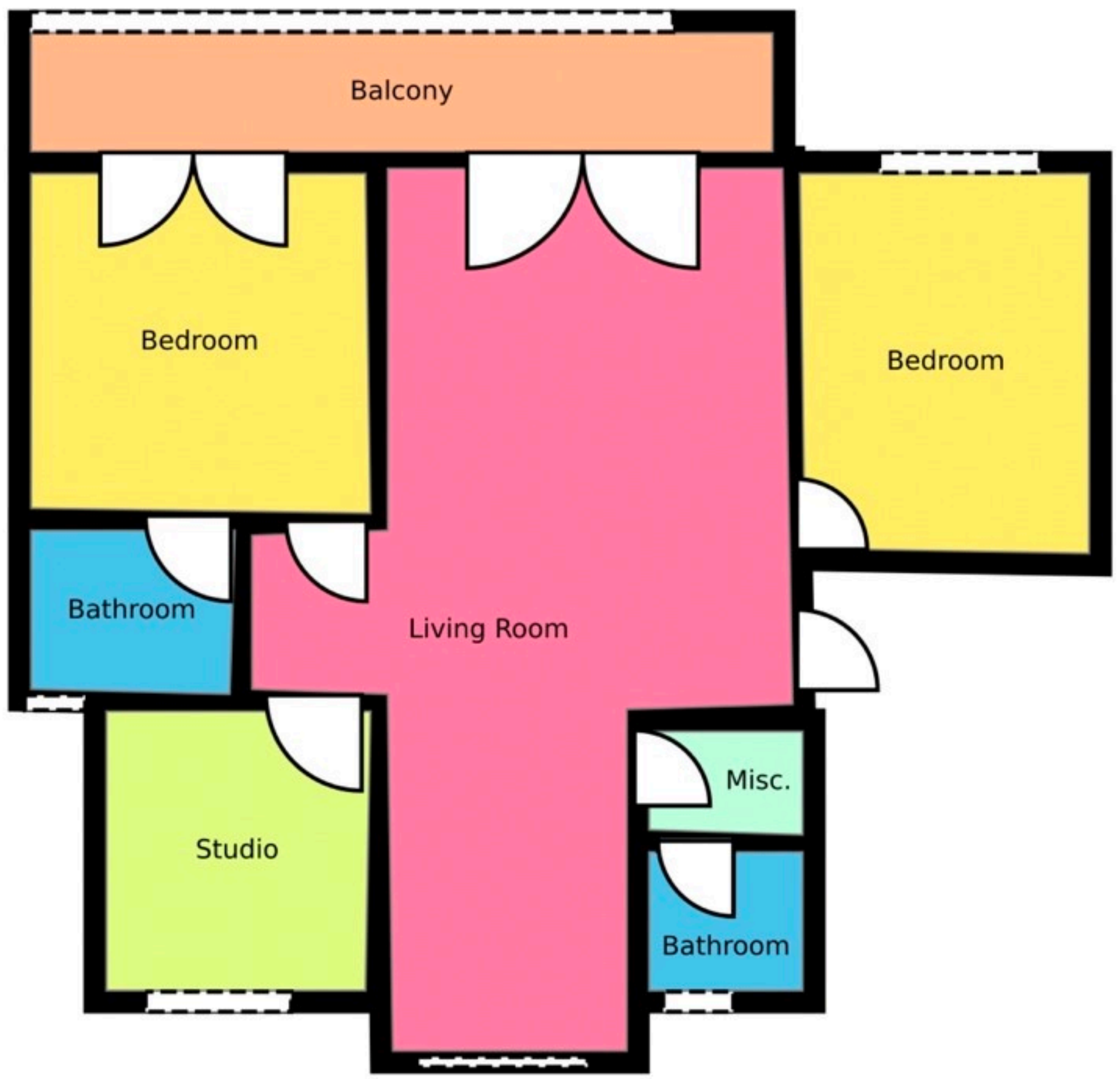
mask3d demo

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

# Floorplan Reconstruction from 3D Scans

RoomFormer [1]

Input: **3D Point Cloud**

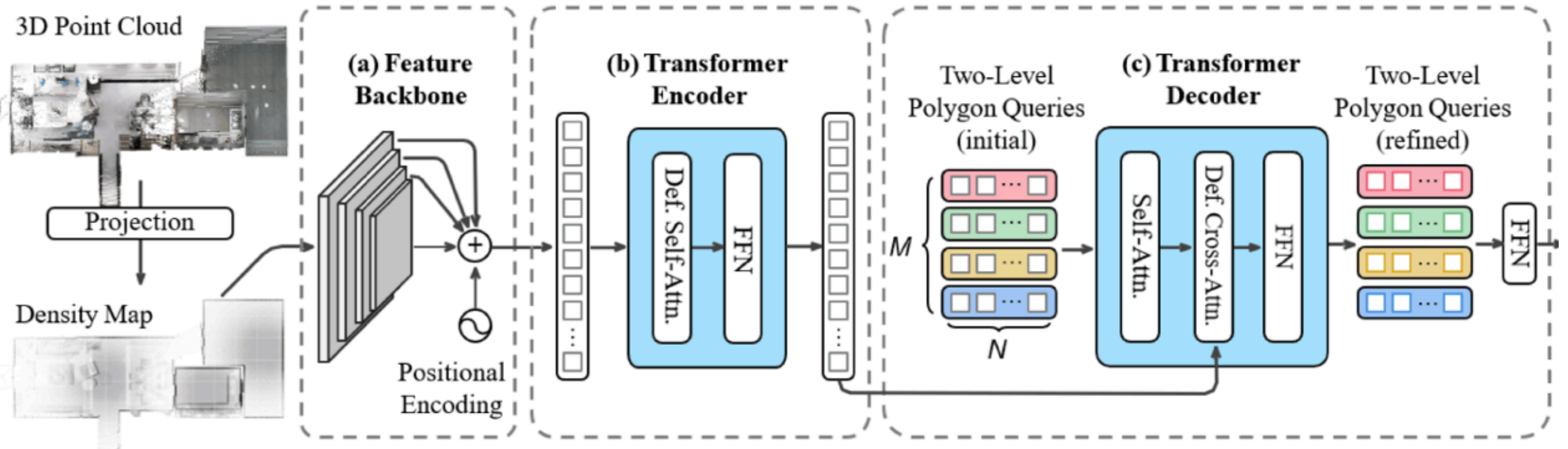


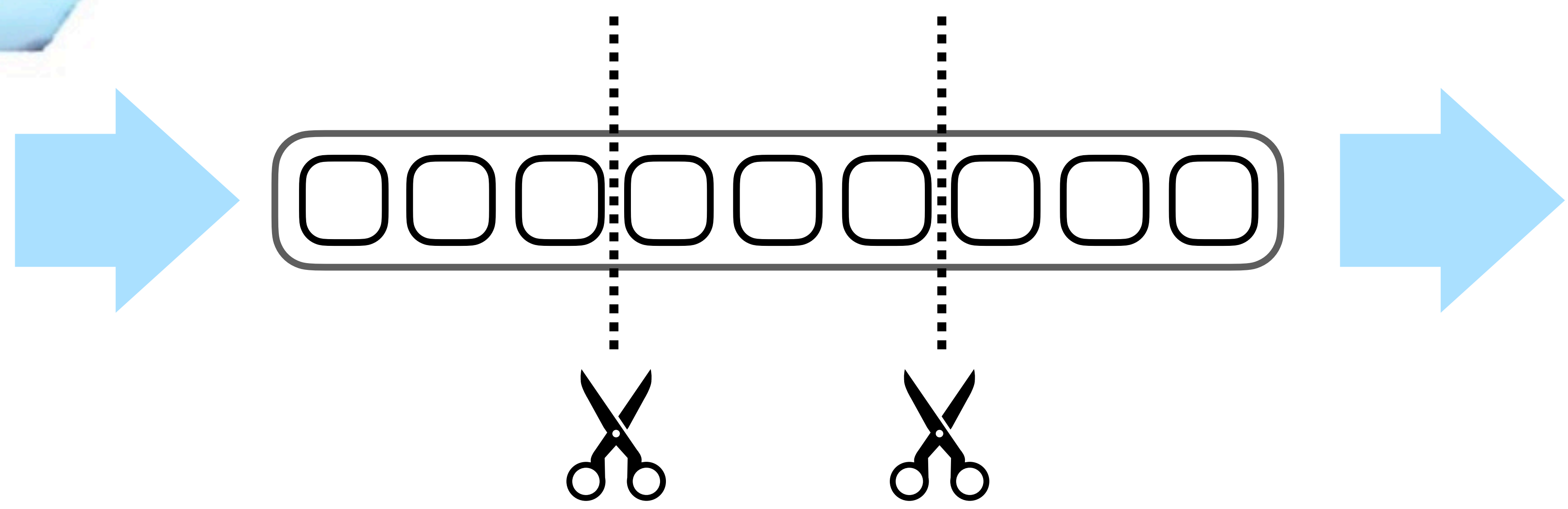

Output: **Vectorized 2D Floorplan**

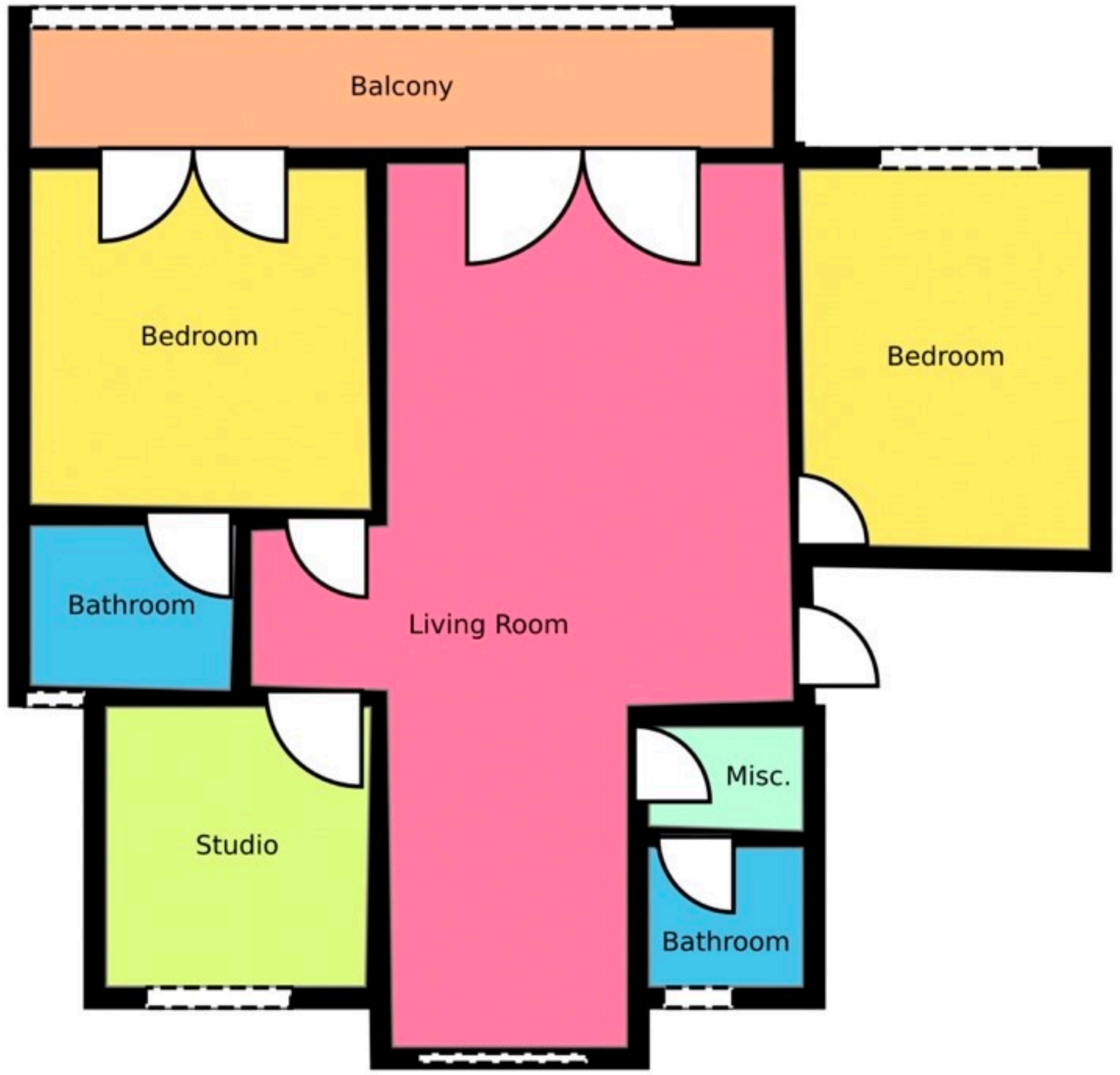
[1] Yue et al. "Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries" CVPR'23

# Floorplan Reconstruction from 3D Scans

## RoomFormer [1]





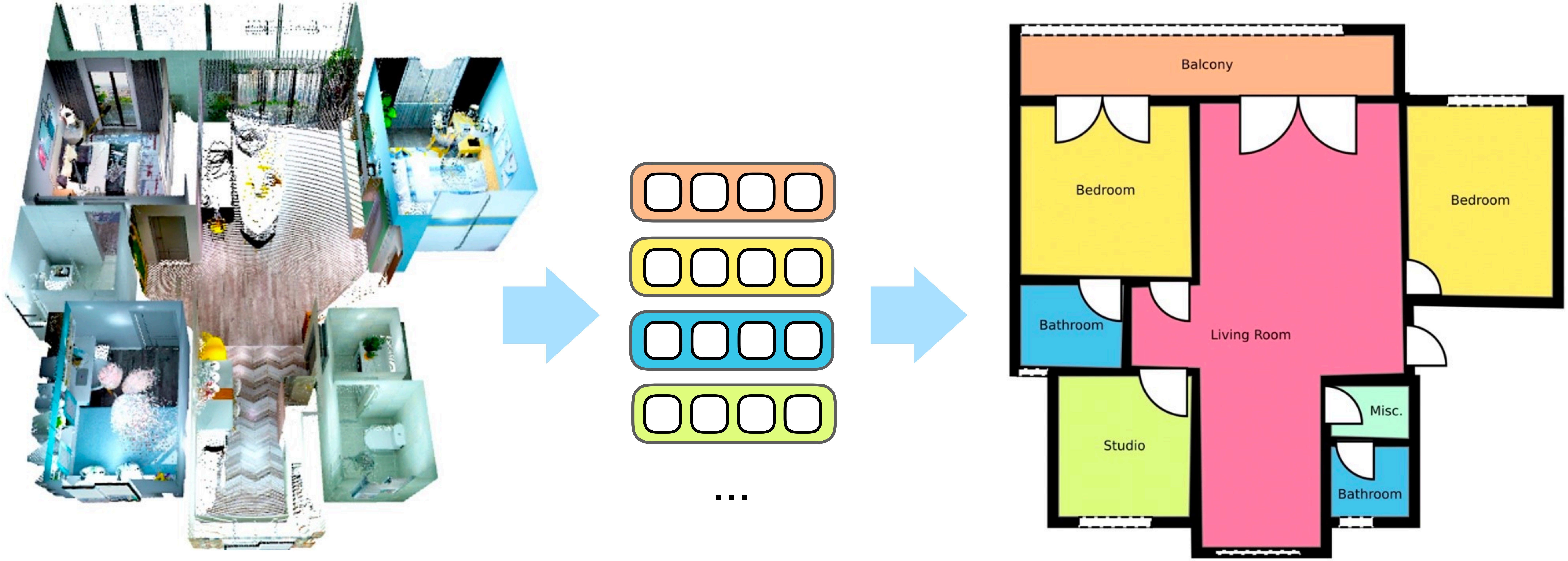

Input: **3D Point Cloud**

Output: **Vectorized 2D Floorplan**

[1] Yue et al. "Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries" CVPR'23

# Floorplan Reconstruction from 3D Scans

RoomFormer [1]




Input: **3D Point Cloud**

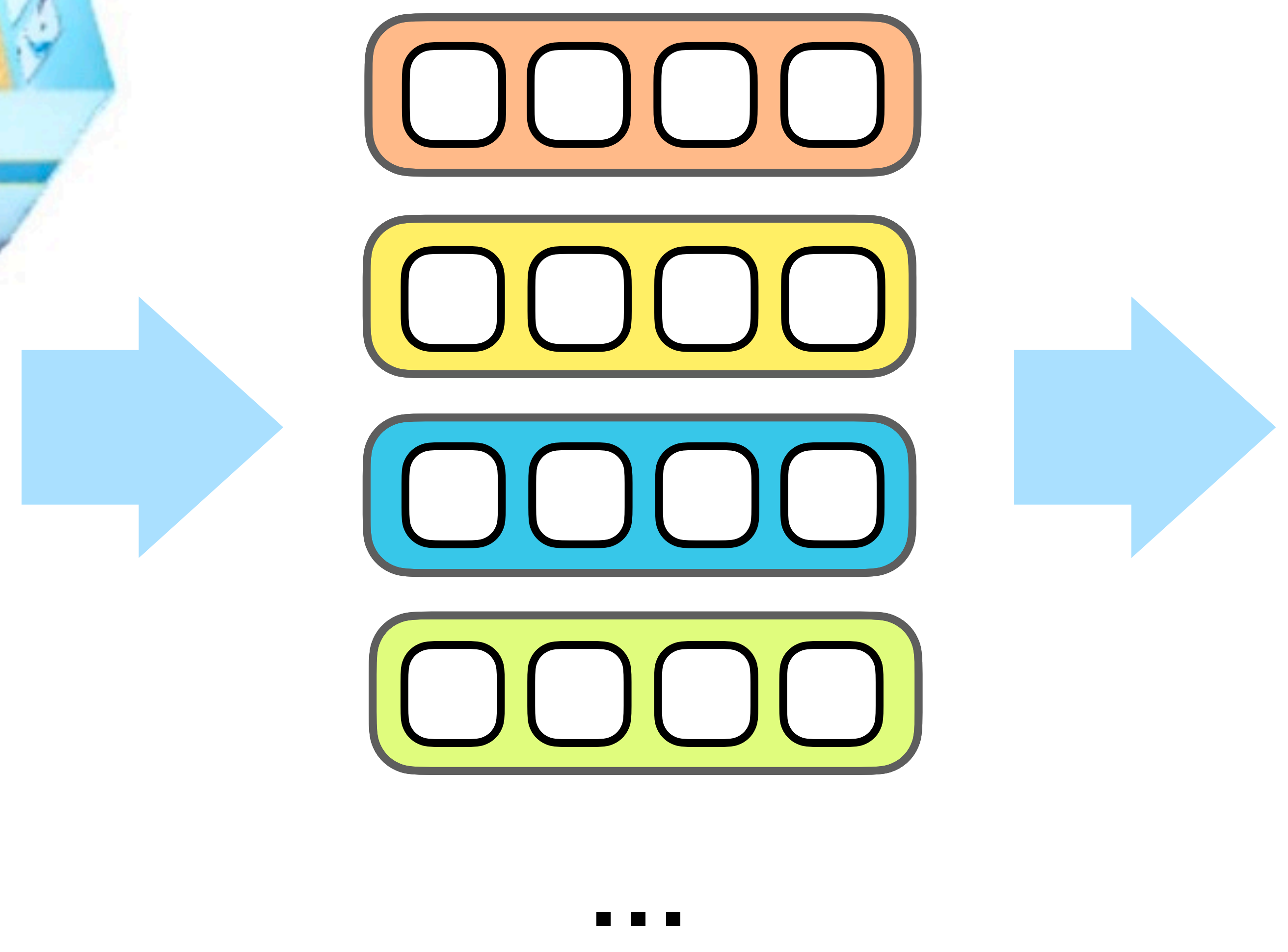
Output: **Vectorized 2D Floorplan**

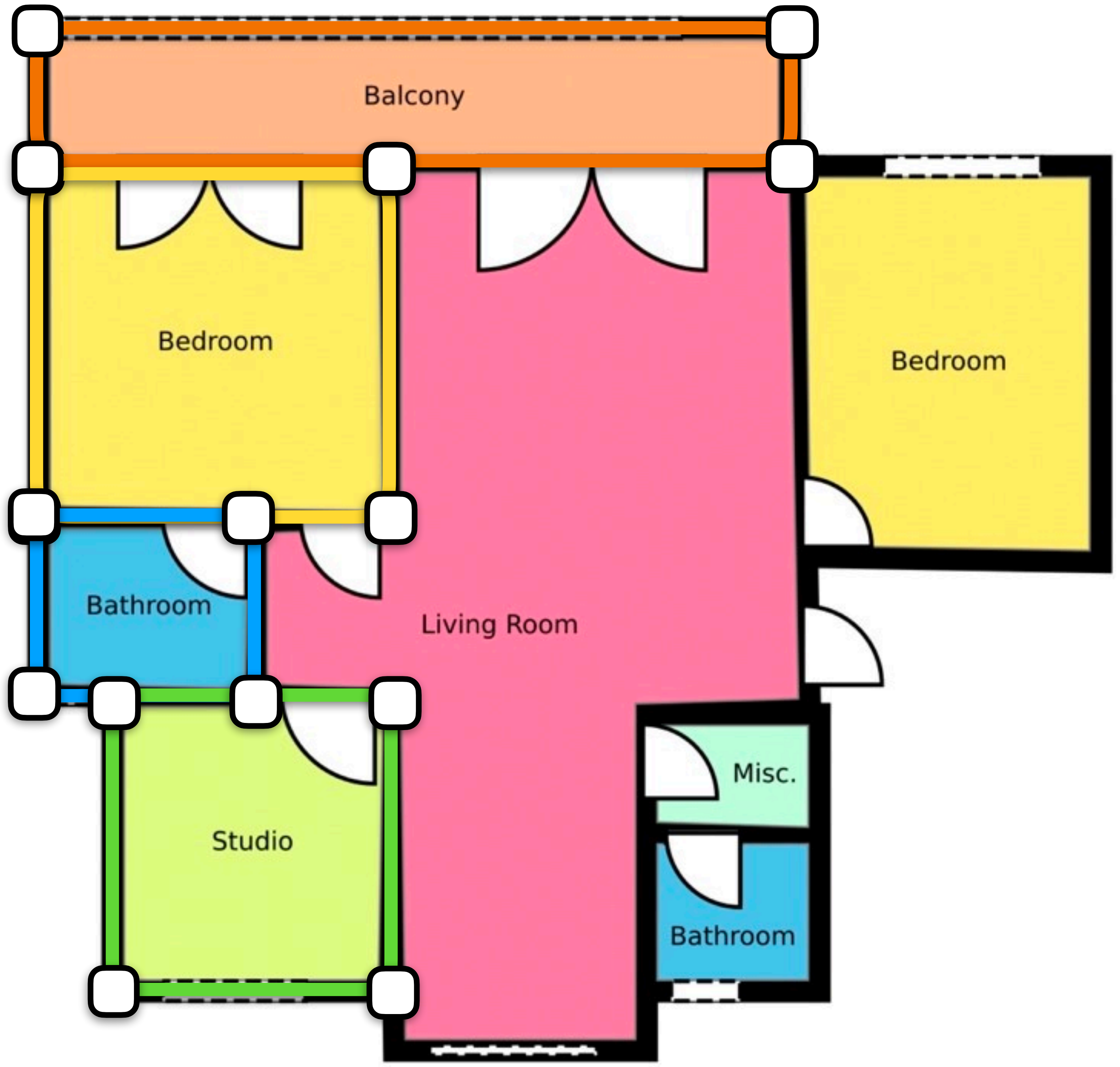
[1] Yue et al. "Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries" CVPR'23

# Floorplan Reconstruction from 3D Scans

RoomFormer [1] representation: **Floorplan as set of polygons**

Input: **3D Point Cloud**



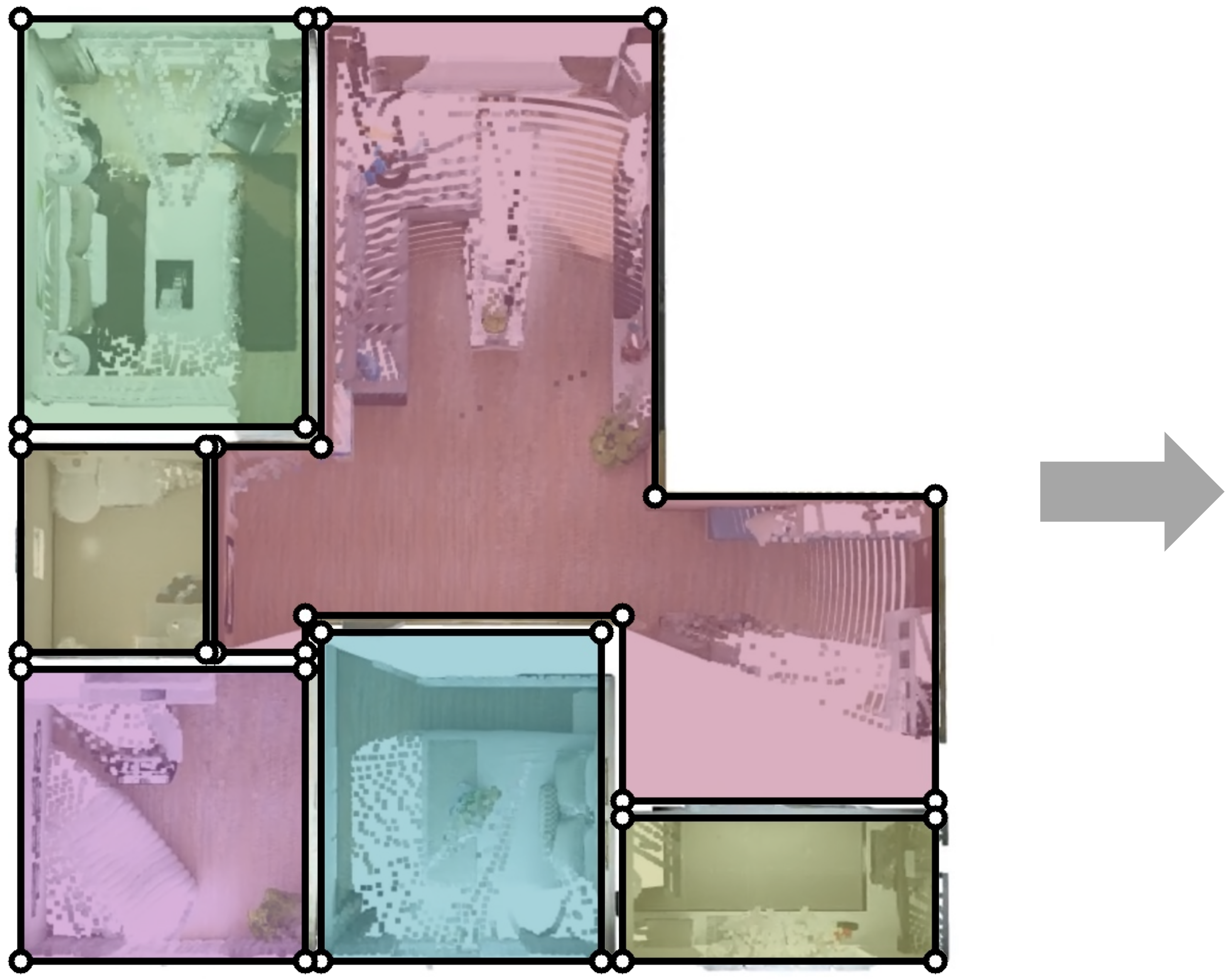


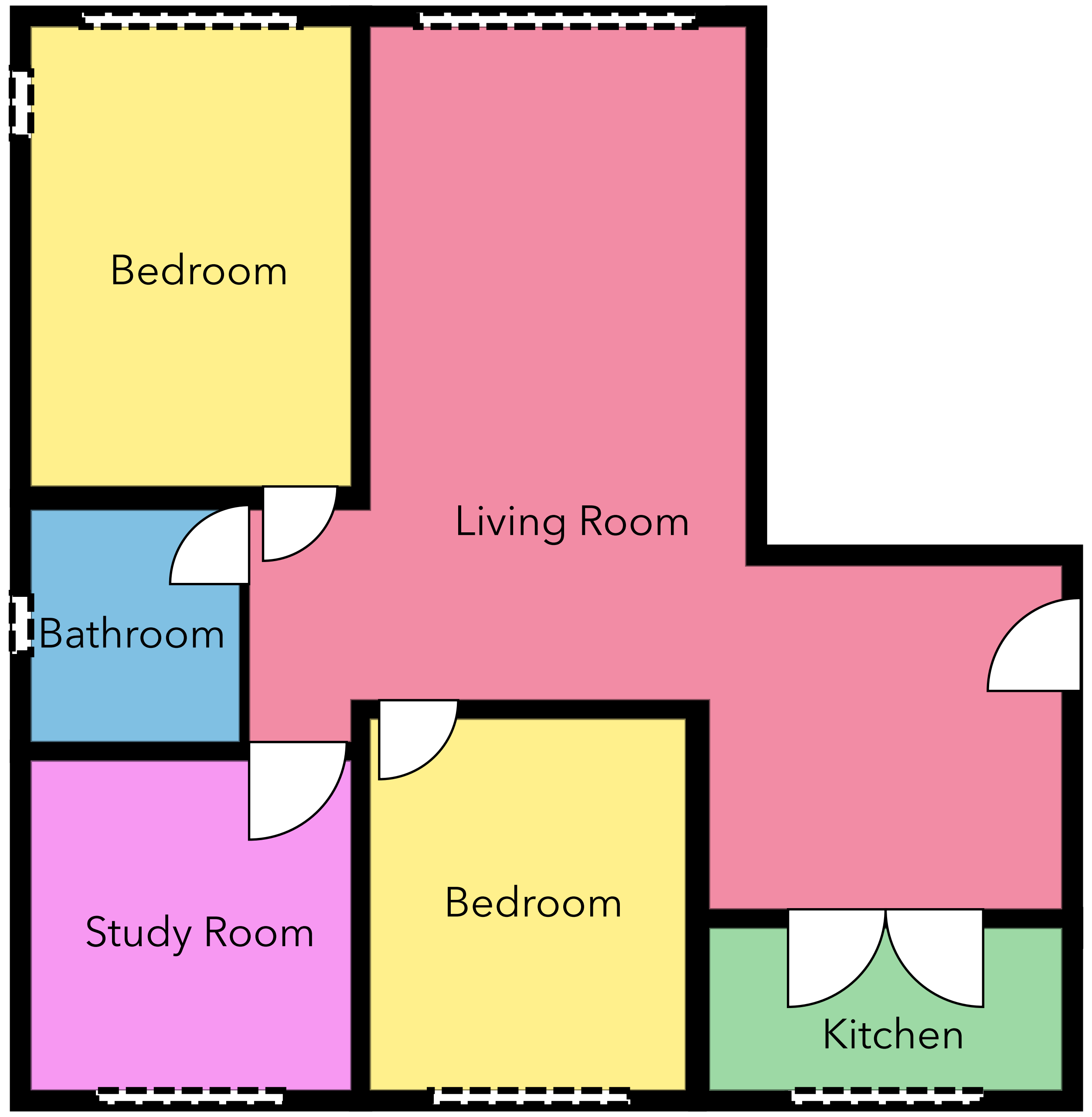

Output: **Vectorized 2D Floorplan**

[1] Yue et al. "Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries" CVPR'23

# Floorplan Reconstruction from 3D Scans

RoomFormer [1]







Input: **3D Scan**

Output: **2D Floorplan**

Additionally: **Semantic elements**
(Room types, doors, windows)

[1] Yue et al. "Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries" CVPR'23

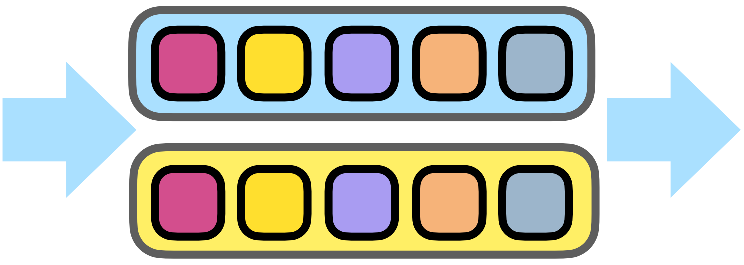
# 3D Segmentation of Humans

Human-Body Part Segmentation


Input: **3D Point Cloud**

Output: **Multi-Human Body-Parts**

Input: **3D Point Cloud**

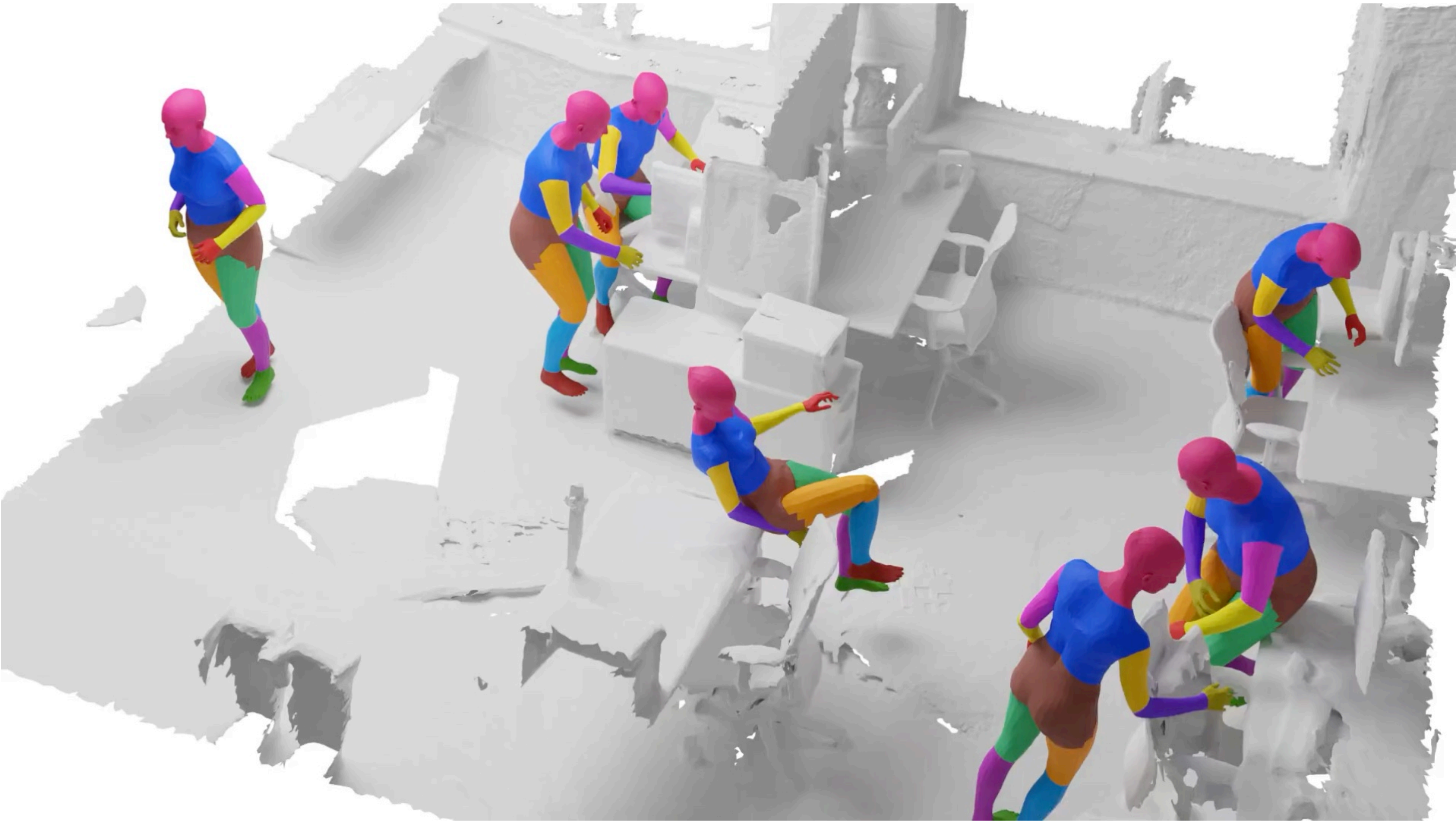



[1] Takmaz et al. "Human3D: 3D Segmentation of Humans in Point Clouds with Synthetic Data" ICCV'23

# 3D Segmentation of Humans

## Synthetic Training Data

**Synthesized Human Instances**



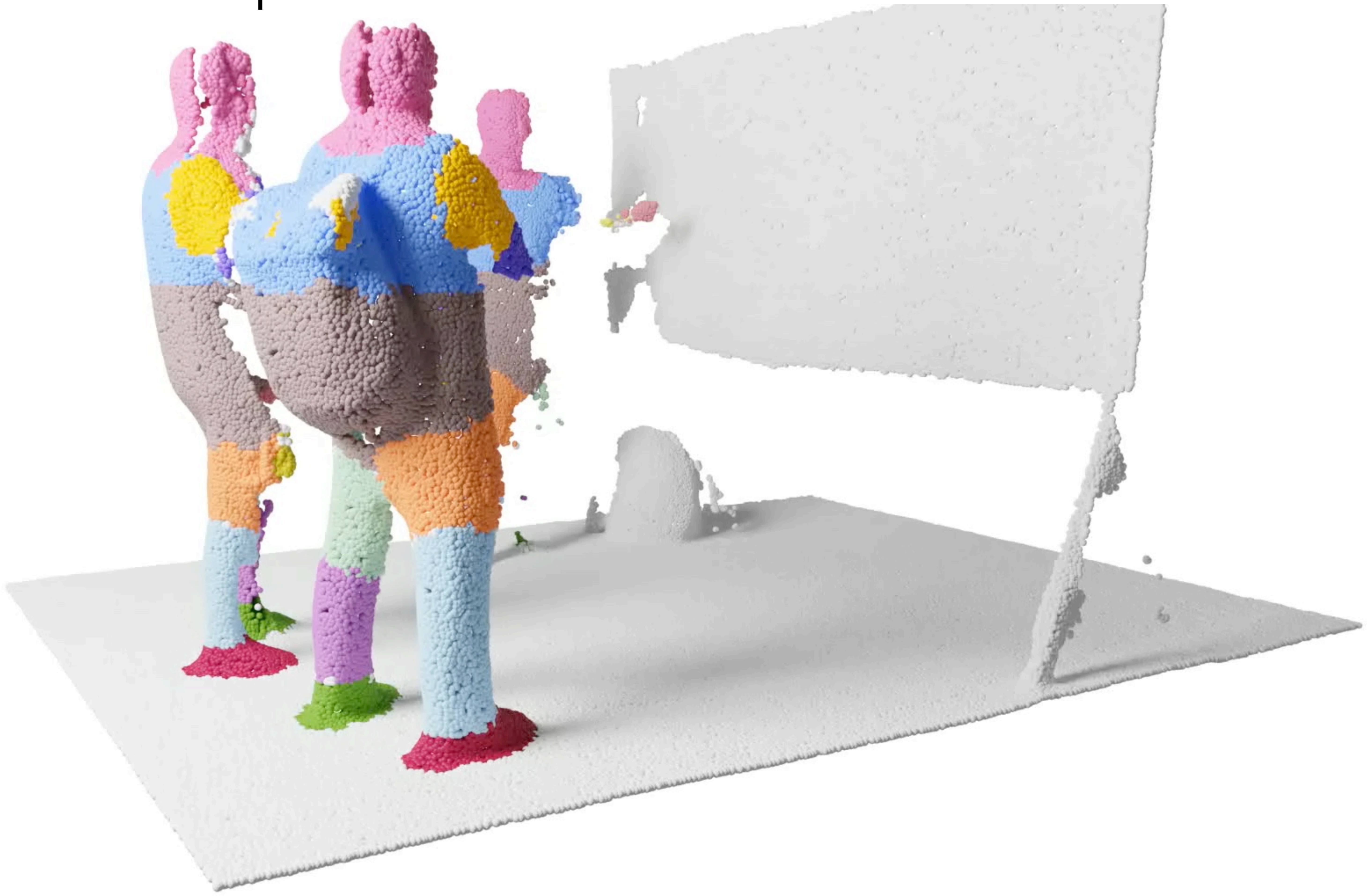
**Synthesized Human Body Parts**

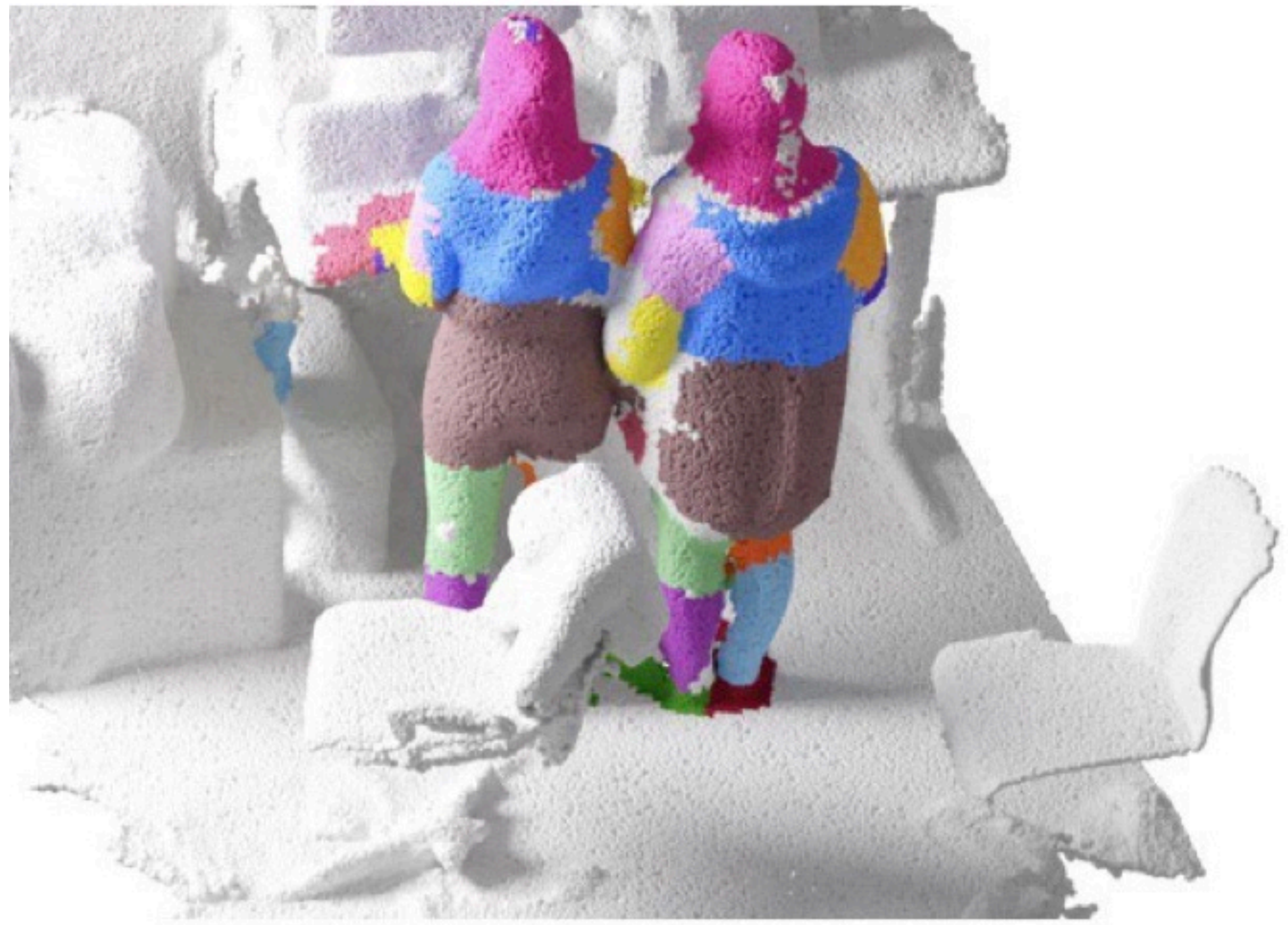
[1] Takmaz et al. "Human3D: 3D Segmentation of Humans in Point Clouds with Synthetic Data" ICCV'23

*How well does it really work ?*

# 3D Segmentation of Humans

## Real-World Examples





# 3D Scene Understanding *In-the-Wild*

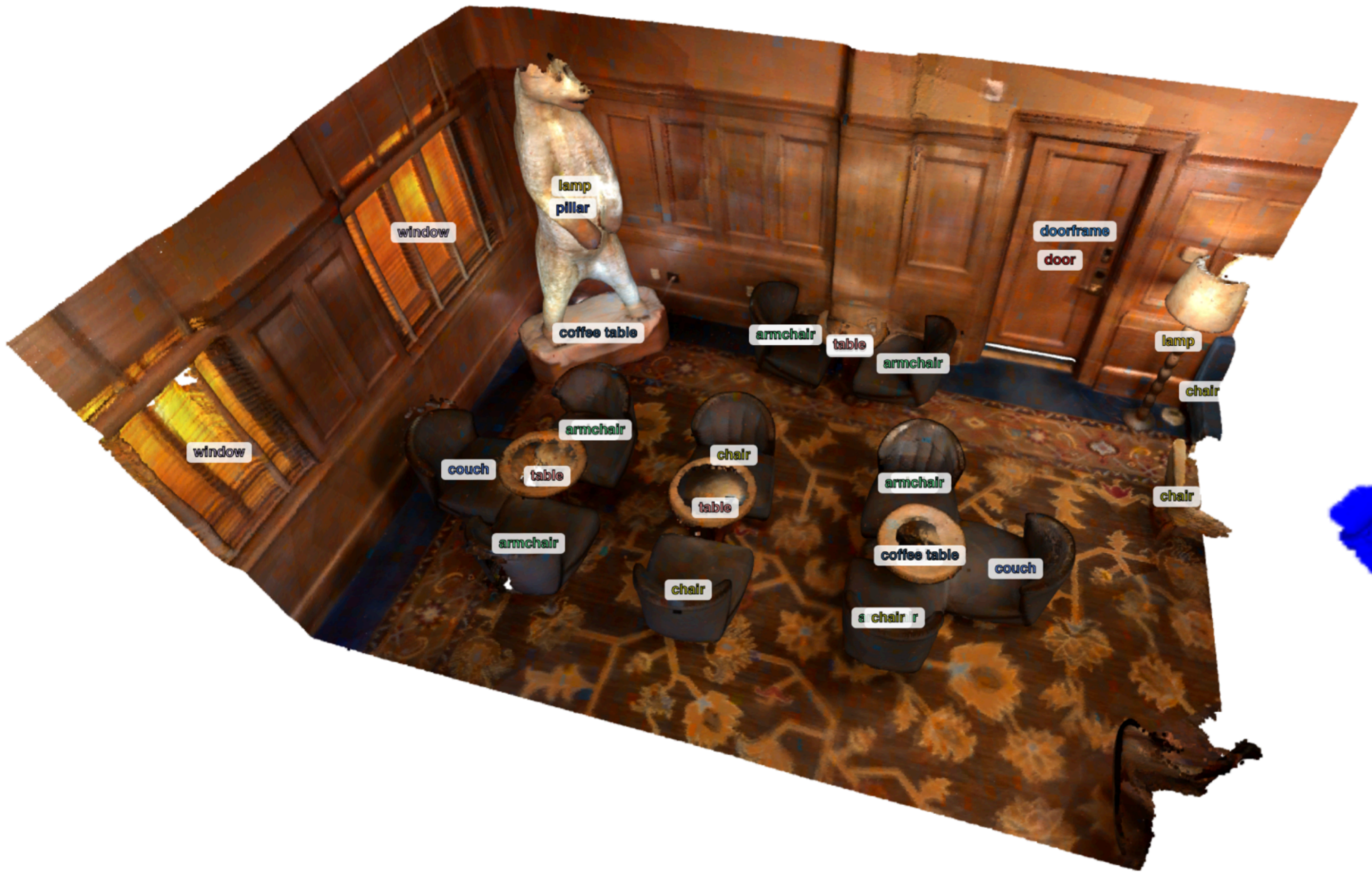
Current models work quite well for a large variety of tasks …


Input: 3D Point Cloud


Output: 3D Semantics

# 3D Scene Understanding *In-the-Wild*

Current models work quite well for a large variety of tasks ...

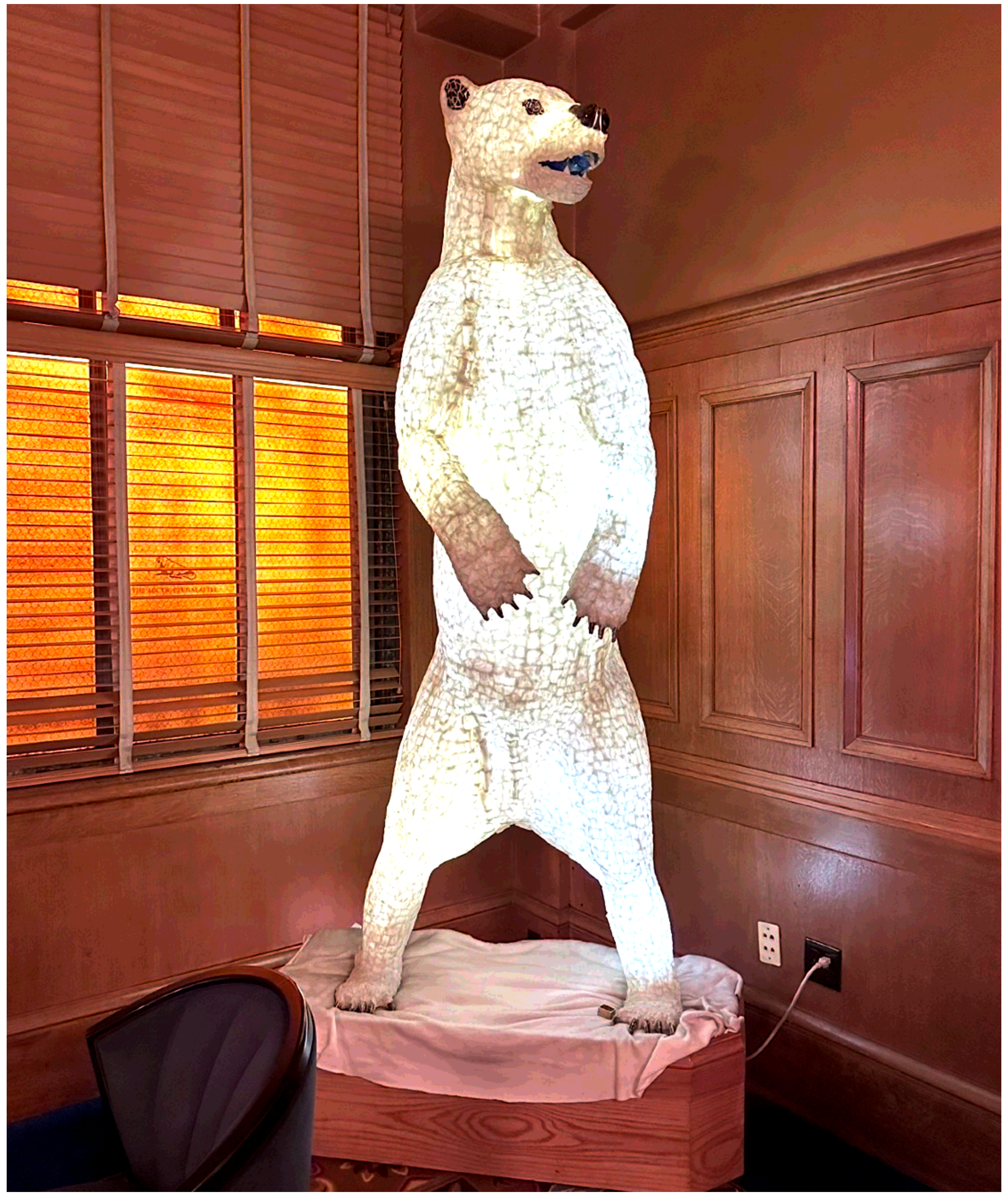


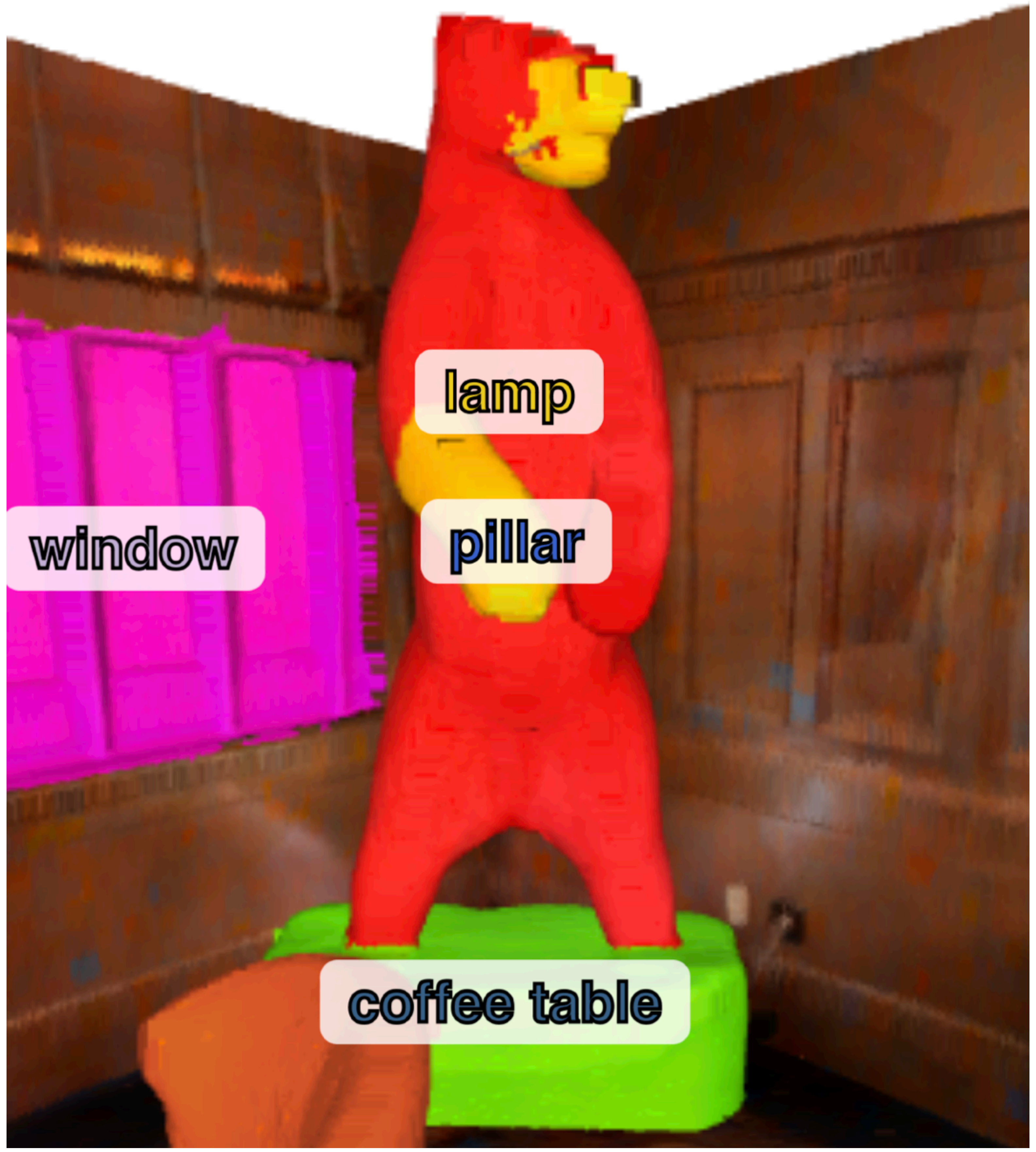

Input: 3D Point Cloud


Output: 3D Semantics

# 3D Scene Understanding *In-the-Wild*

... but **limited to a <u>predefined</u> closed set of classes!**



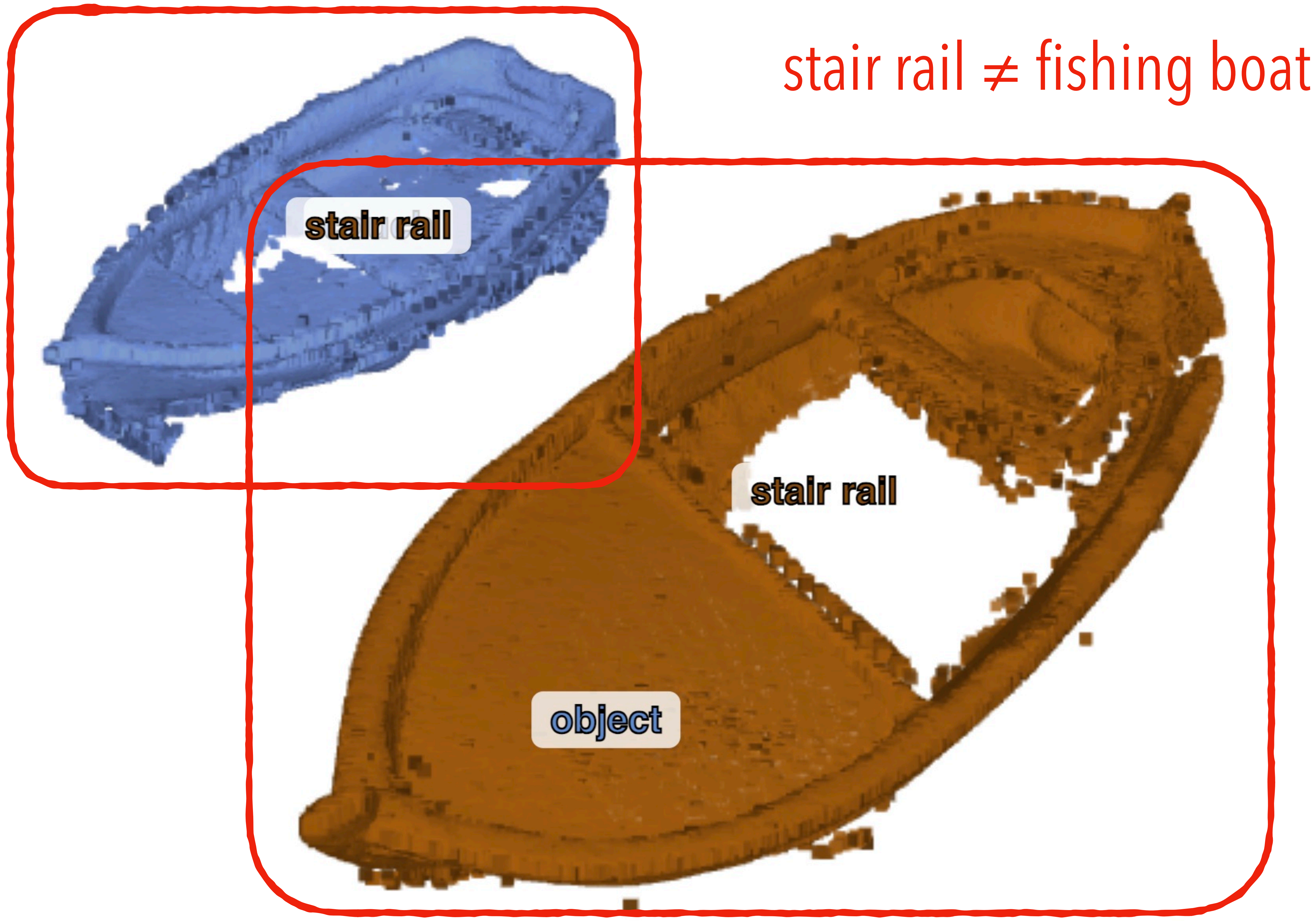



Output: 3D Instance Masks

# 3D Scene Understanding *In-the-Wild*

... but **limited to a <u>predefined</u> closed set of classes!**

Input: 3D Point Cloud




Output: 3D Semantics

# ***Open-World*** **3D Scene Understanding**

not limited to classes seen during training (closed-world)

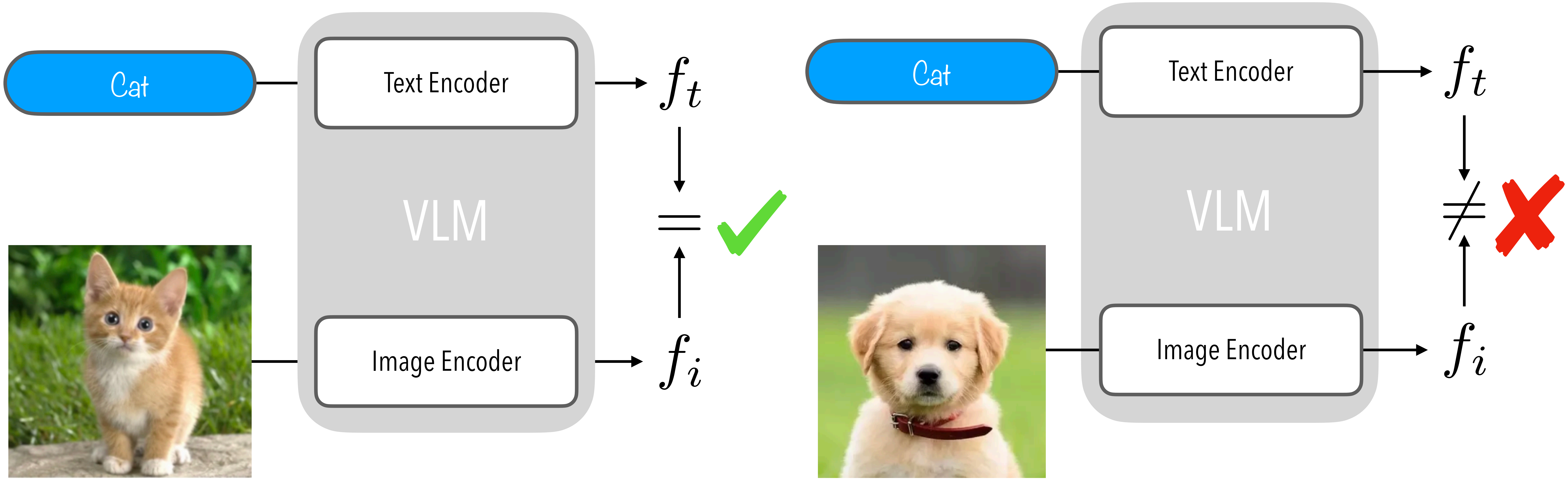
# Goal: Open-Vocabulary 3D Scene Understanding

Given arbitrary user-query, segment the corresponding scene elements

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

# How can we achieve Open-Vocabulary 3D Scene Understanding?

Large Visual Language Model (VLM) e.g., *CLIP [1] or SigLIP [2]*



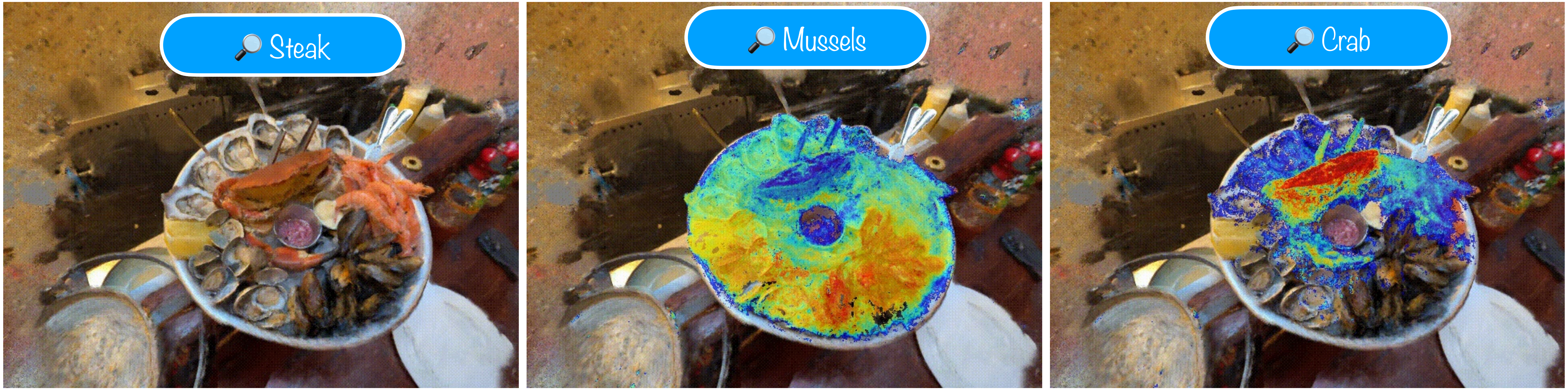

[1] Radford et al. "*Learning Transferable Visual Models From Natural Language Supervision*" ICML'21
[2] Zhai et al. "*Sigmoid Loss for Language Image Pre-Training*" ICCV'23

# How can we achieve Open-Vocabulary 3D Scene Understanding?

## Optimize NeRF representation with additional CLIP feature channel

Mechanism for zero-short image segmentation:

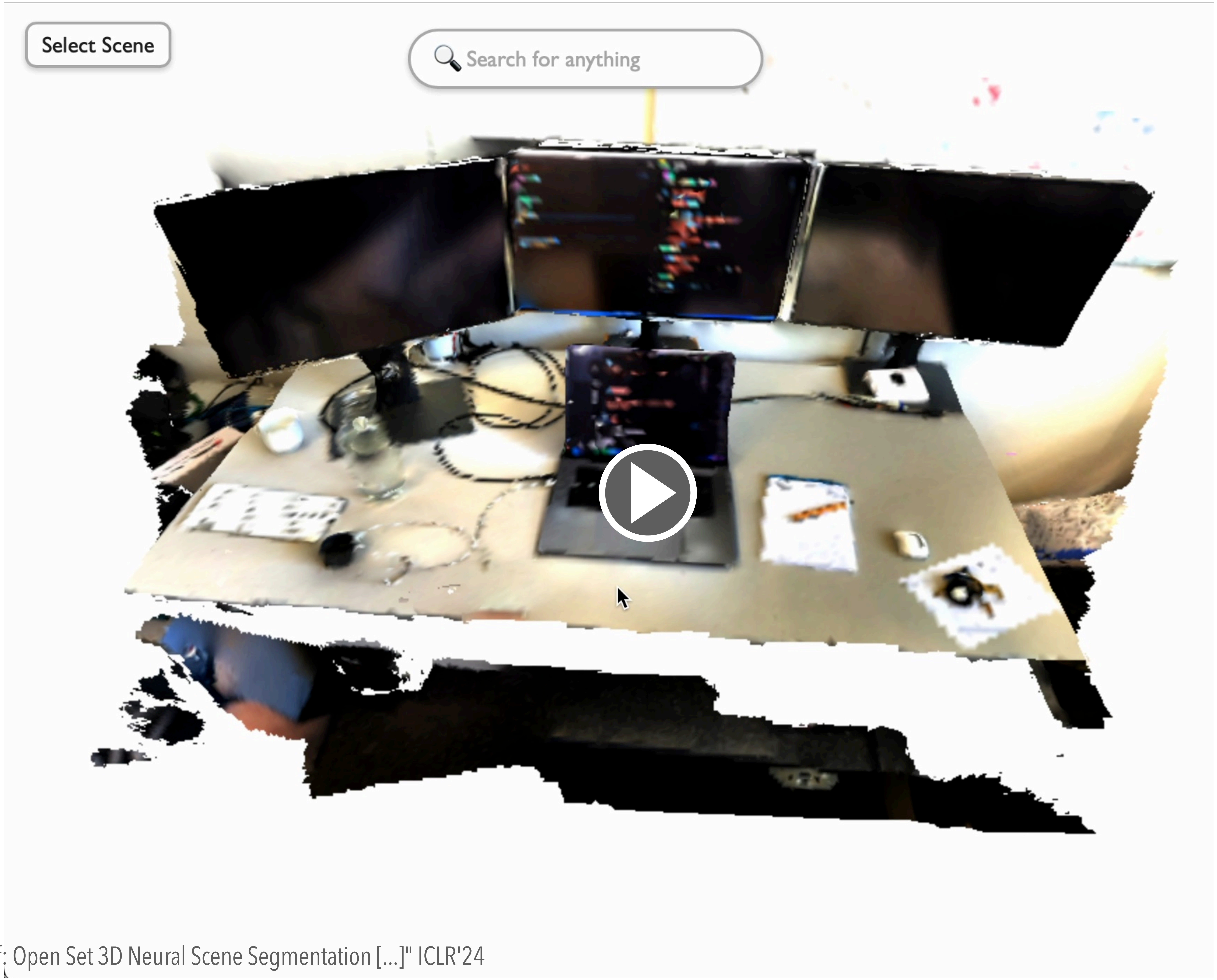
1. Compute CLIP [1] encoding of text query and per-pixel CLIP features via OpenSeg [2]
2. Get response from dot-product of normalized encodings



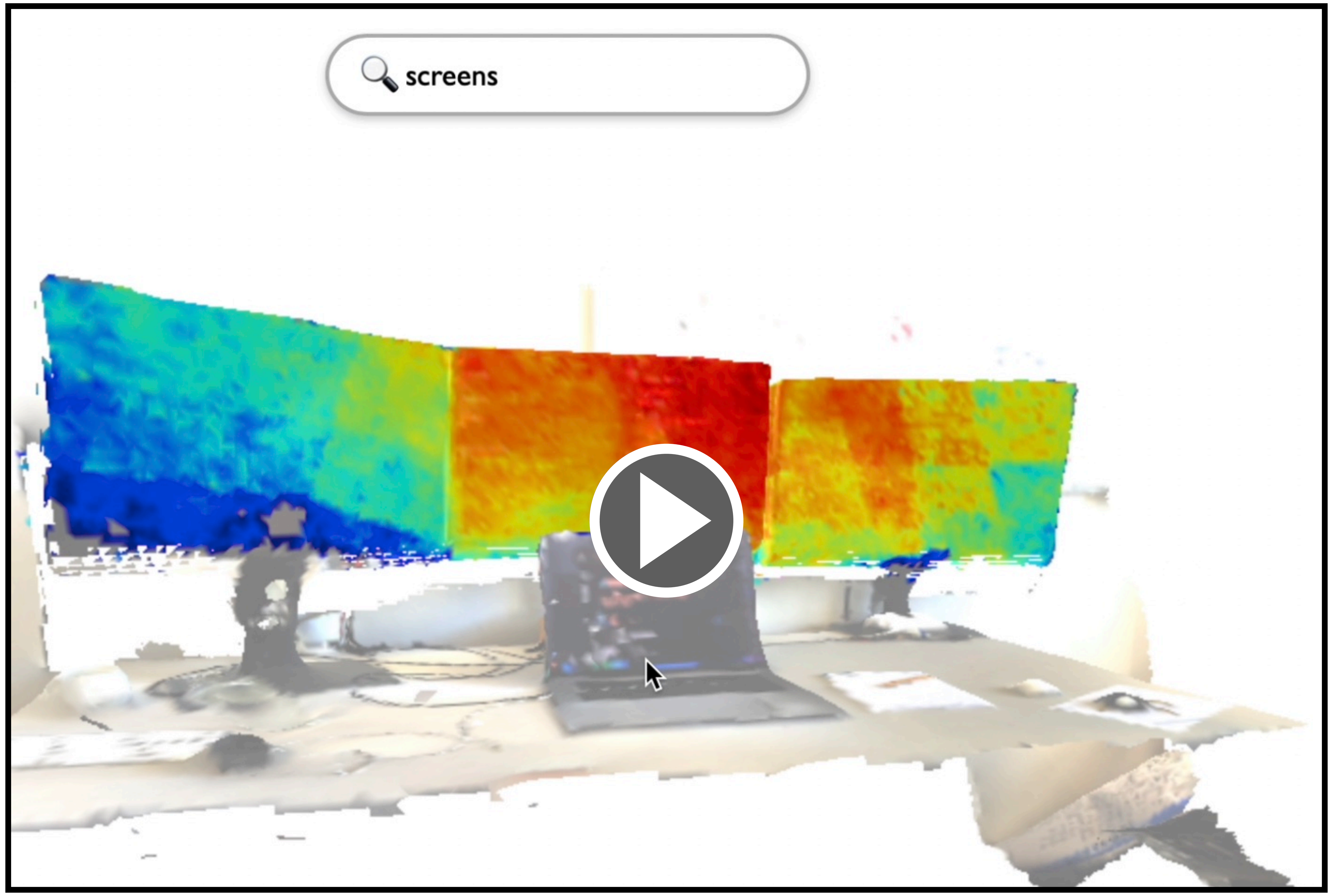



[1] Radford et al. "*Learning Transferable Visual Models From Natural Language Supervision*" ICML'21

[2] Zhai et al. "*Sigmoid Loss for Language Image Pre-Training*" ICCV'23




Engelmann et al. "OpenNerf: Open Set 3D Neural Scene Segmentation [...]" ICLR'24




*What about different **instances**?*

# Open-Vocabulary 3D Instance Segmentation

## OpenMask3D [1]

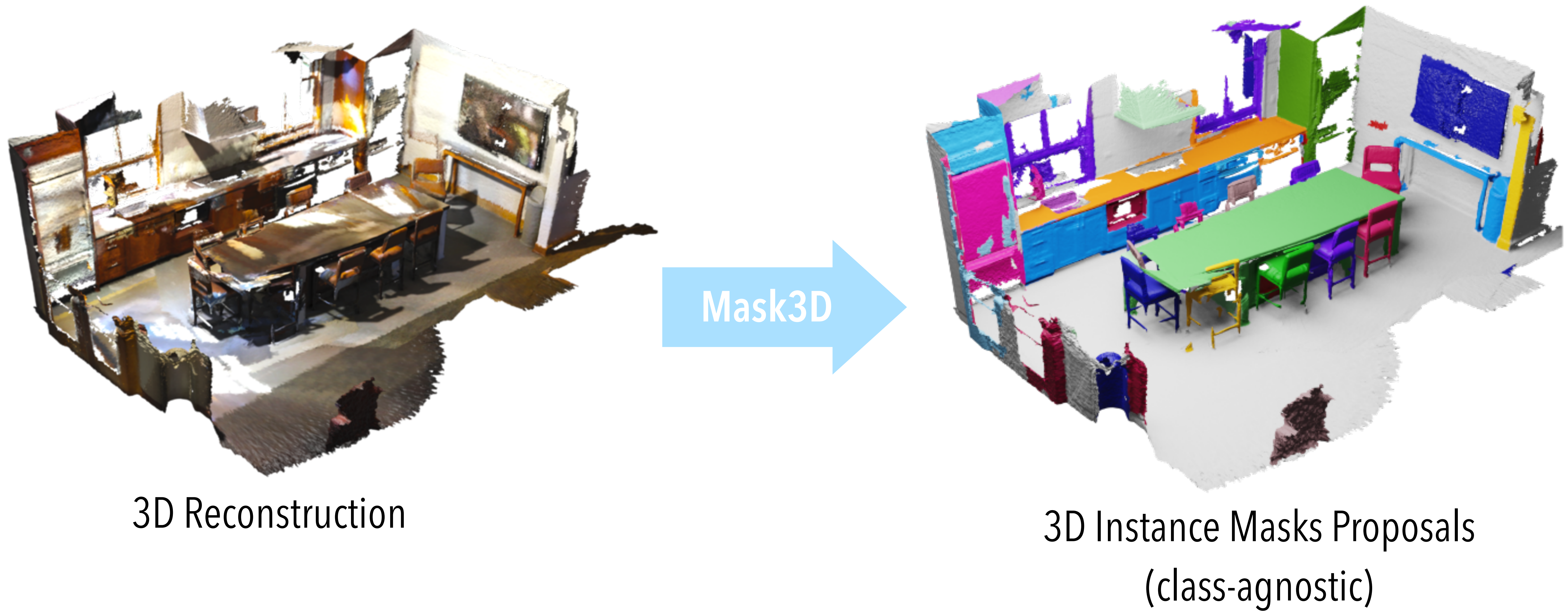
**Input:** 3D Scene Representation + Search Query


**Output:** 3D instance masks corresponding to search query


[1] Takmaz, Fedele et al. "OpenMask3D" NeurIPS'23

# OpenMask3D: Open-Vocabulary 3D Instance Segmentation

How to obtain the instance masks?



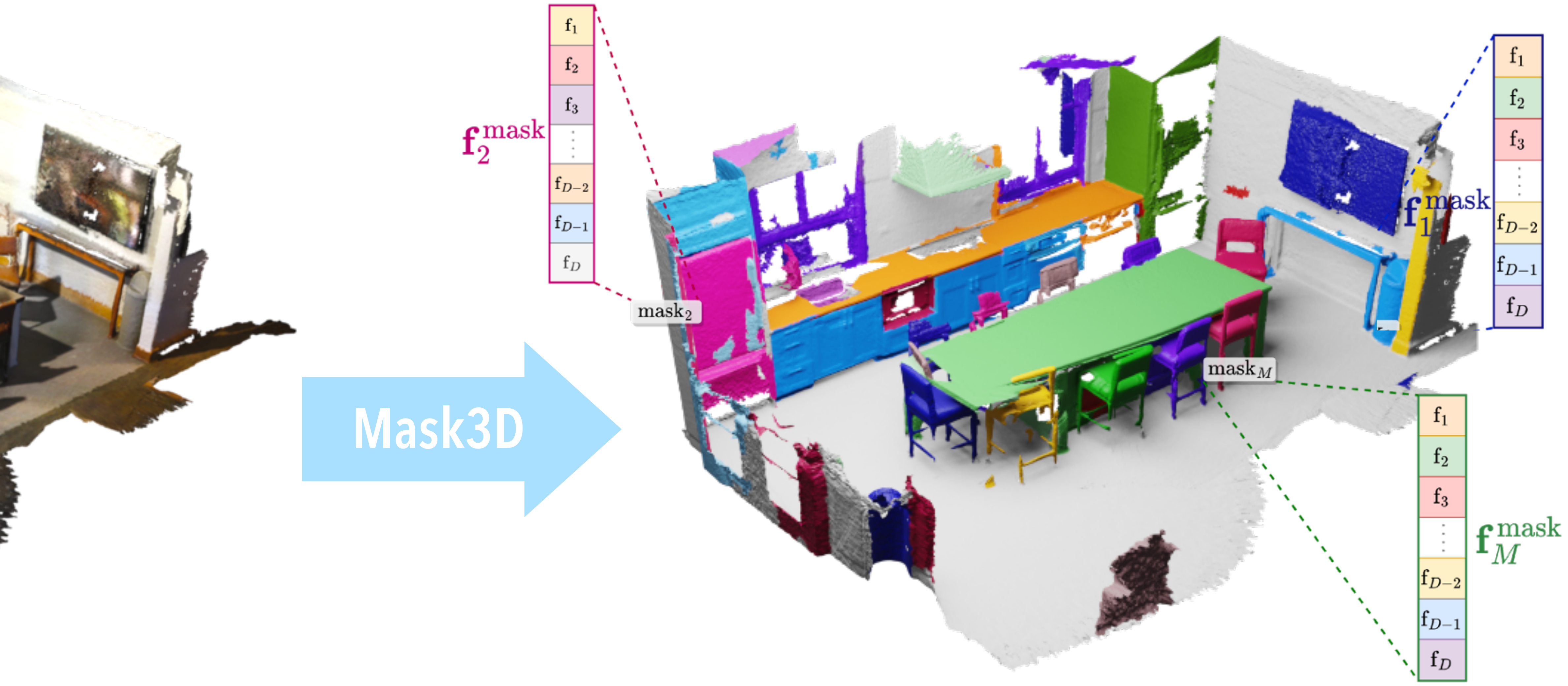


[1] Takmaz, Fedele et al. "OpenMask3D: Open-Vocabulary 3D Instance Segmentation" NeurIPS'23

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

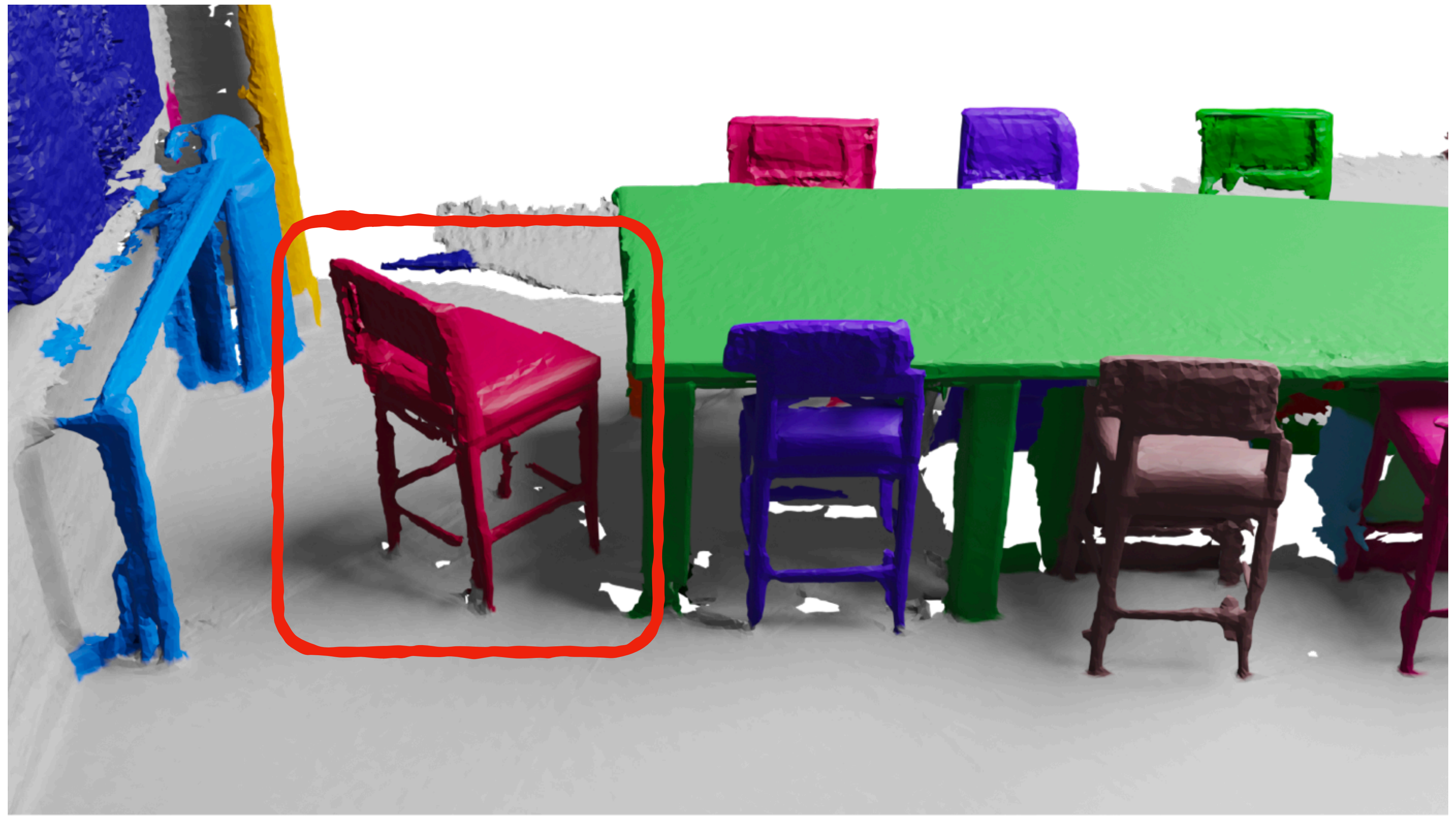
# OpenMask3D: Open-Vocabulary 3D Instance Segmentation

How to obtain the per-mask CLIP features?

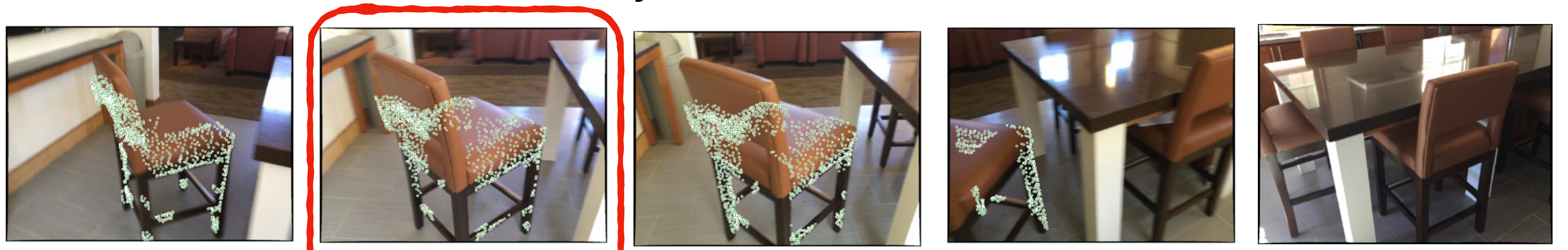



n

3D Instance Masks Proposals
(class-agnostic)



Project 3D mask to 2D views



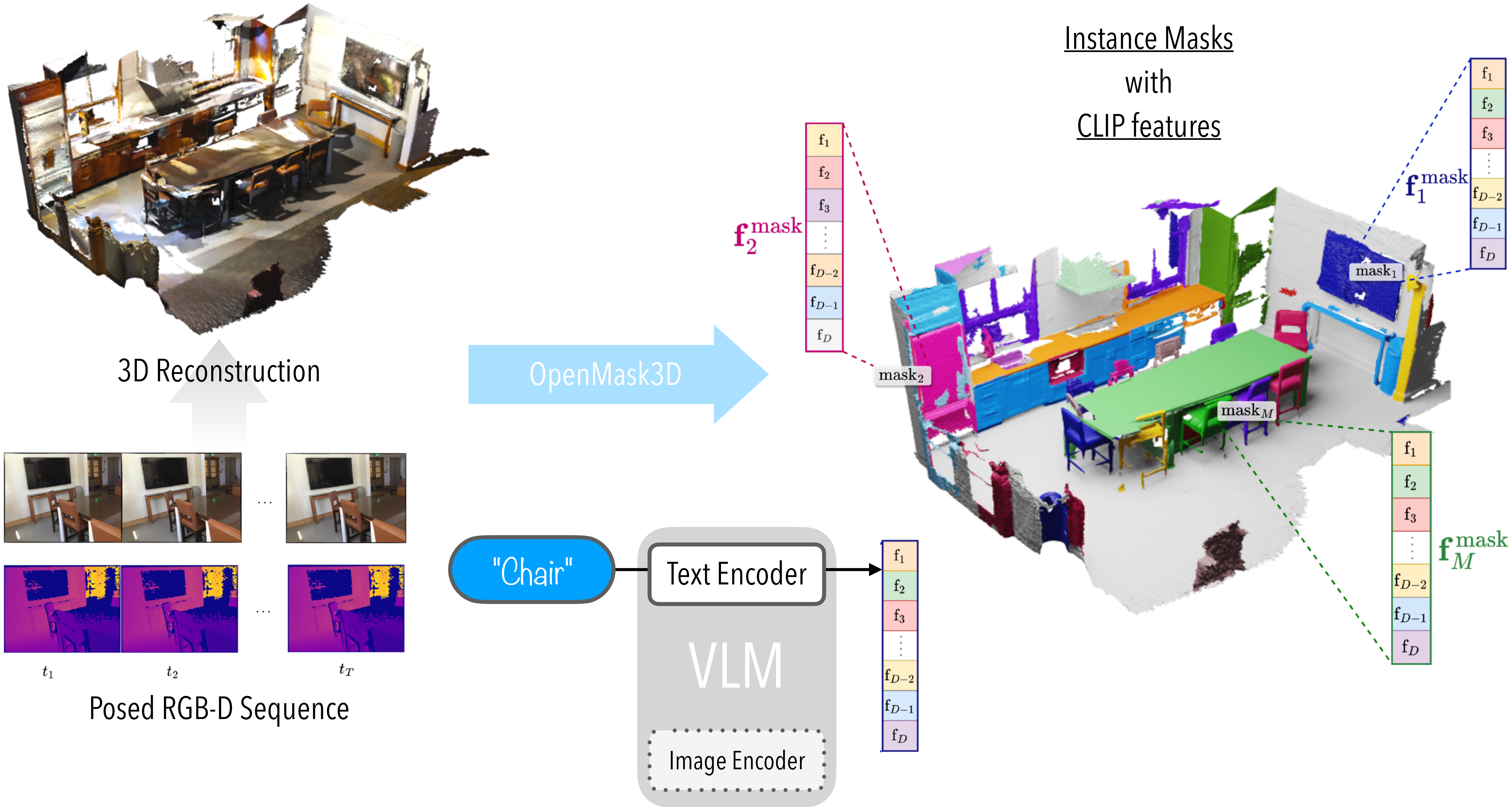
Visibility score: 100% 90% 94% 30% 0%

1. Compute tight bounding box via SAM.
2. Compute multi-scale CLIP features.
3. Average over multiple scales & views (top k views).


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

# OpenMask3D: Open-Vocabulary 3D Instance Segmentation

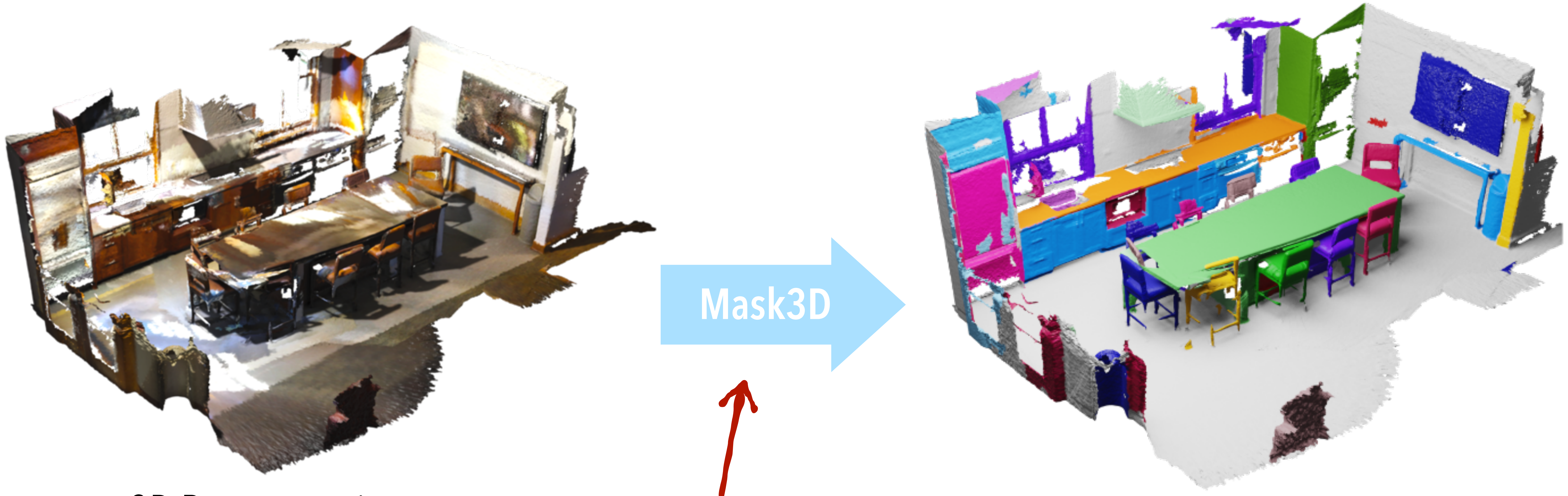
3D Scene Representation



*Who sees the limitation?*

# OpenMask3D: Open-Vocabulary 3D Instance Segmentation

How to obtain the instance masks?




3D Reconstruction

3D Instance Masks Proposals
(class-agnostic)

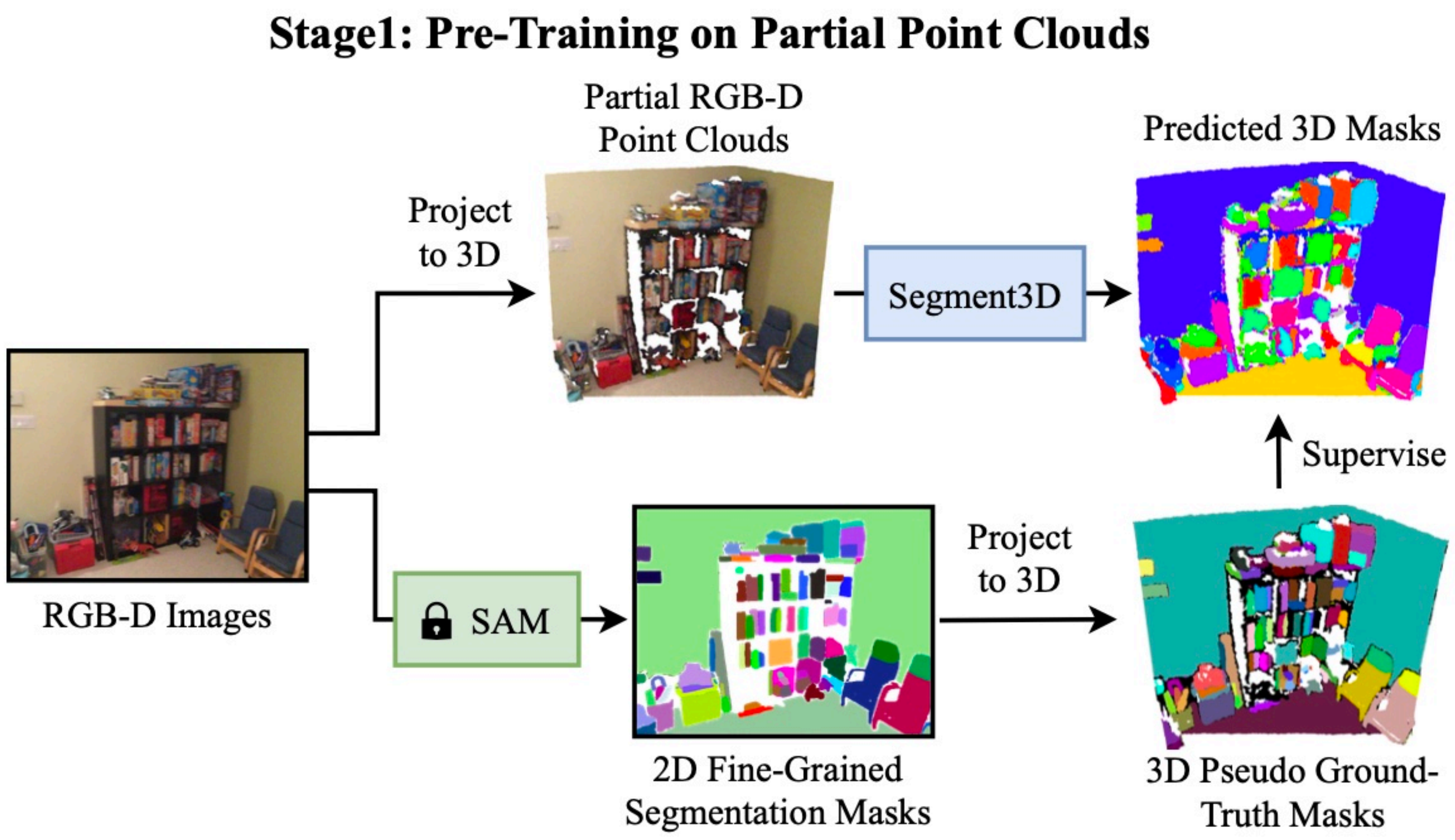
trained on "closed-world" dataset

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

# Segment Anything Model (SAM)

# Open-World 3D Segmentation

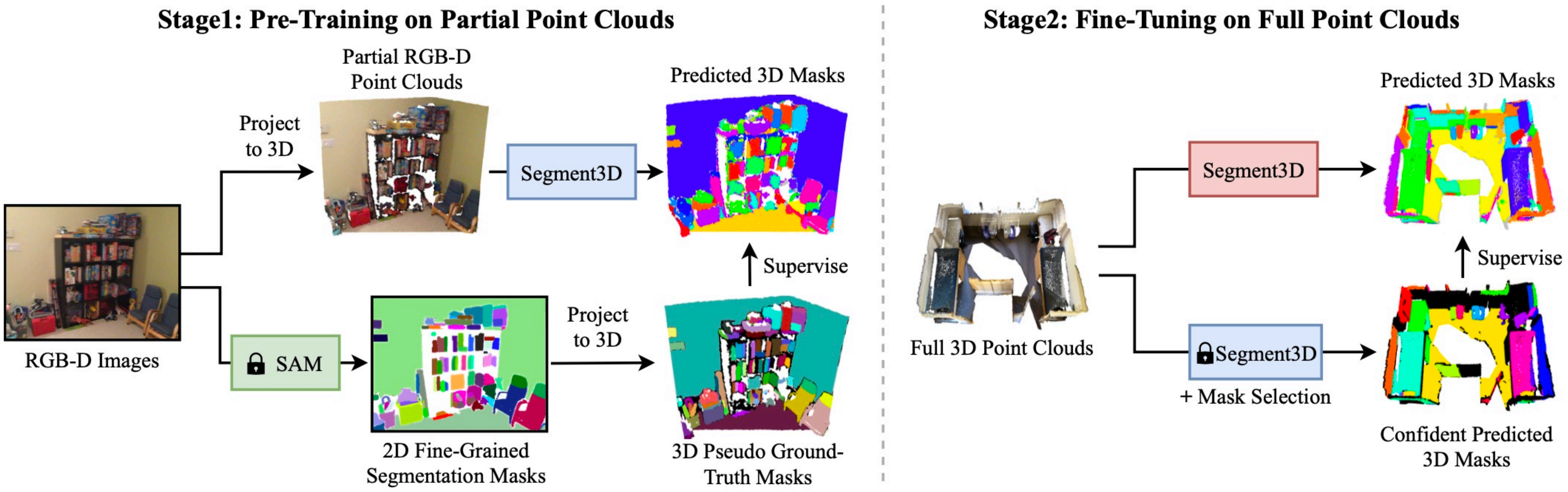
**Problem**: Manually labeled datasets are naturally limited to a closed set of classes (for example ScanNet)
**Question**: Can we use segmentation foundation model for open-set 3D segmentation? (SAM)
**Challenge**: Domain gap between 2D image space and 3D geometry space.




[1] Huang et al. "Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels" ECCV'24

# Open-World 3D Segmentation

**Problem**: Manually labeled datasets are naturally limited to a closed set of classes (for example ScanNet)
**Question**: Can we use segmentation foundation model for open-set 3D segmentation? (SAM)
**Challenge**: Domain gap between 2D image space and 3D geometry space.




[1] Huang et al. "Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels" ECCV'24

# Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels

Mask3D
trained on manual labels.

**Segment3D**
trained on automatic labels.

# Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation

for Open-Vocabulary 3D Segmentation

# Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation

Demo: segment3d.github.io

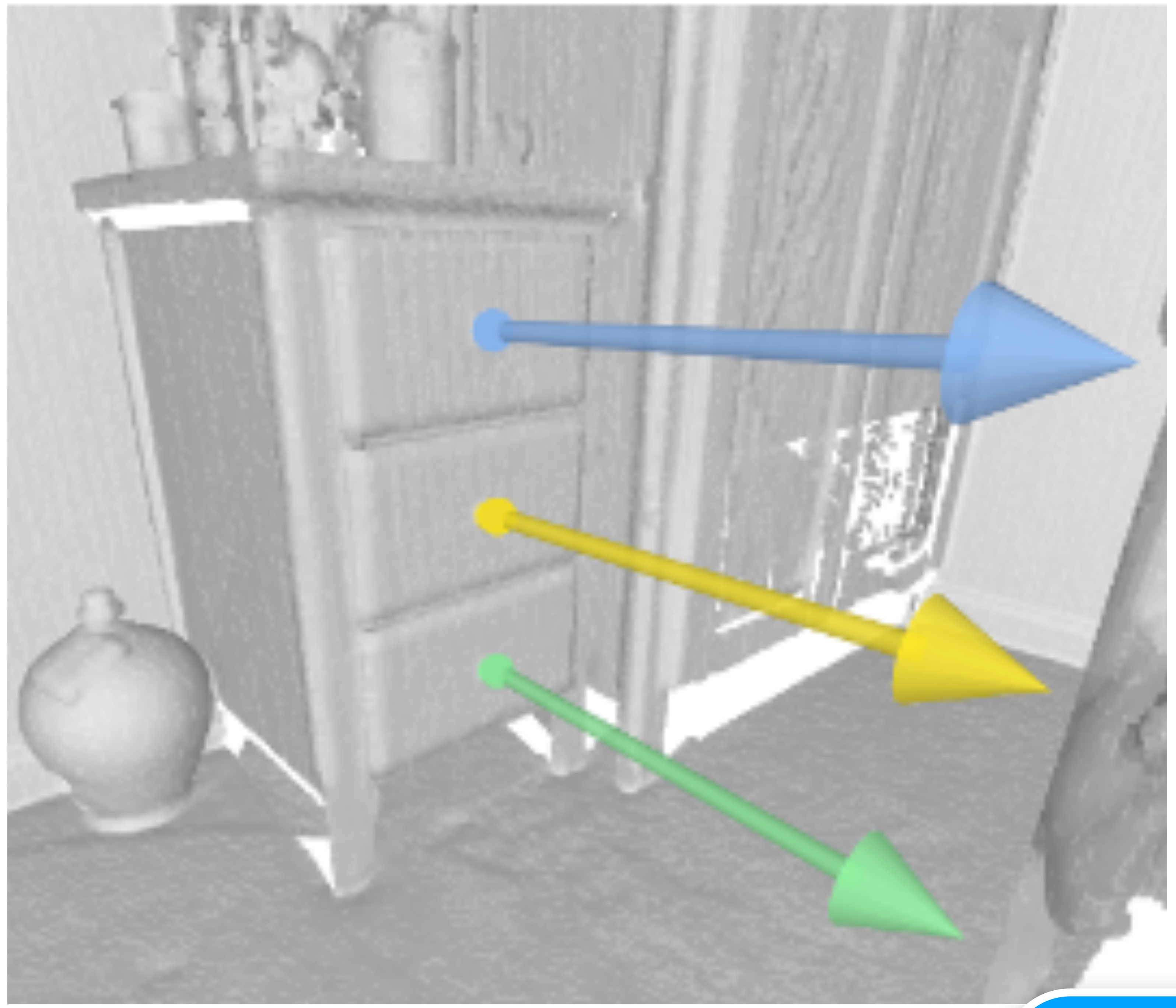
[1] Huang et al. "Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels" ECCV'24

*Is this all we need?*


BostonDynamics
ETH zürich
Computer Vision
and Geometry Lab

# Towards *Functional* 3D Scene Understanding

From Objects ...



... to Interactions & Functionalities

Open the drawer

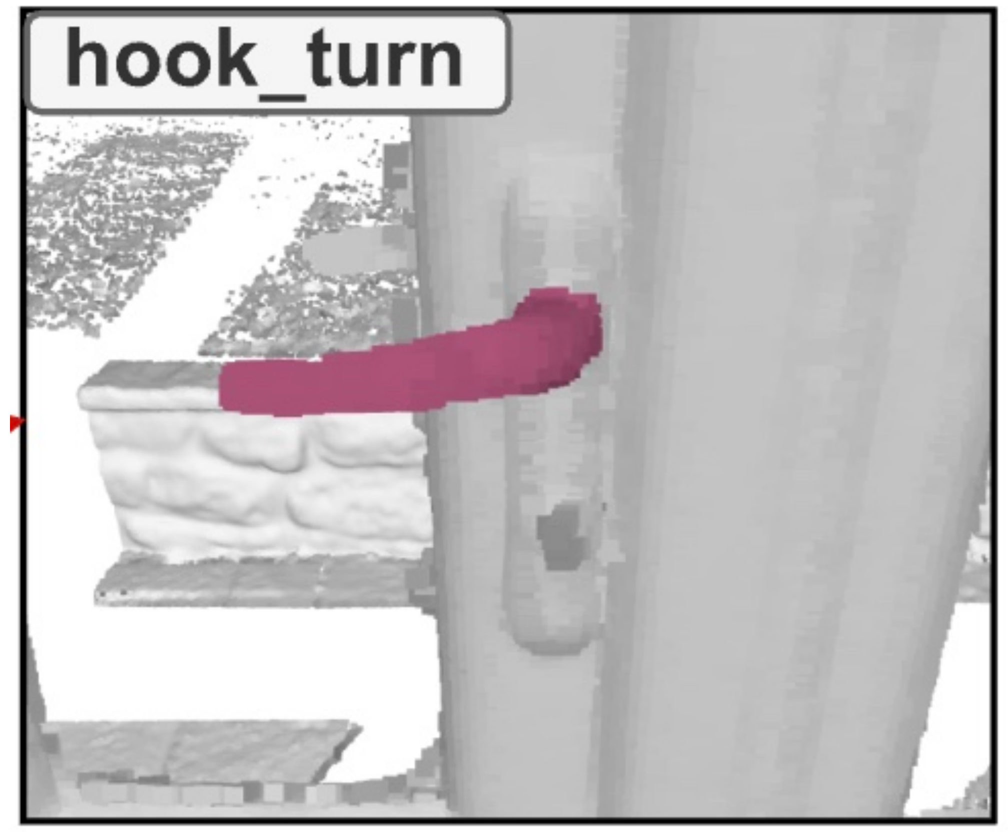
***Where?*** ***How?*** ***What?***

[1] Delitzas et al. "SceneFun3D: Fine-grained Functionality and Affordance Understanding in 3D Scenes" CVPR'24 (Oral)


Task 1: Functionality segmentation

# **SceneFun3D:** Fine-grained Functionality and Affordance Understanding in 3D Scenes

## Functionality Annotations

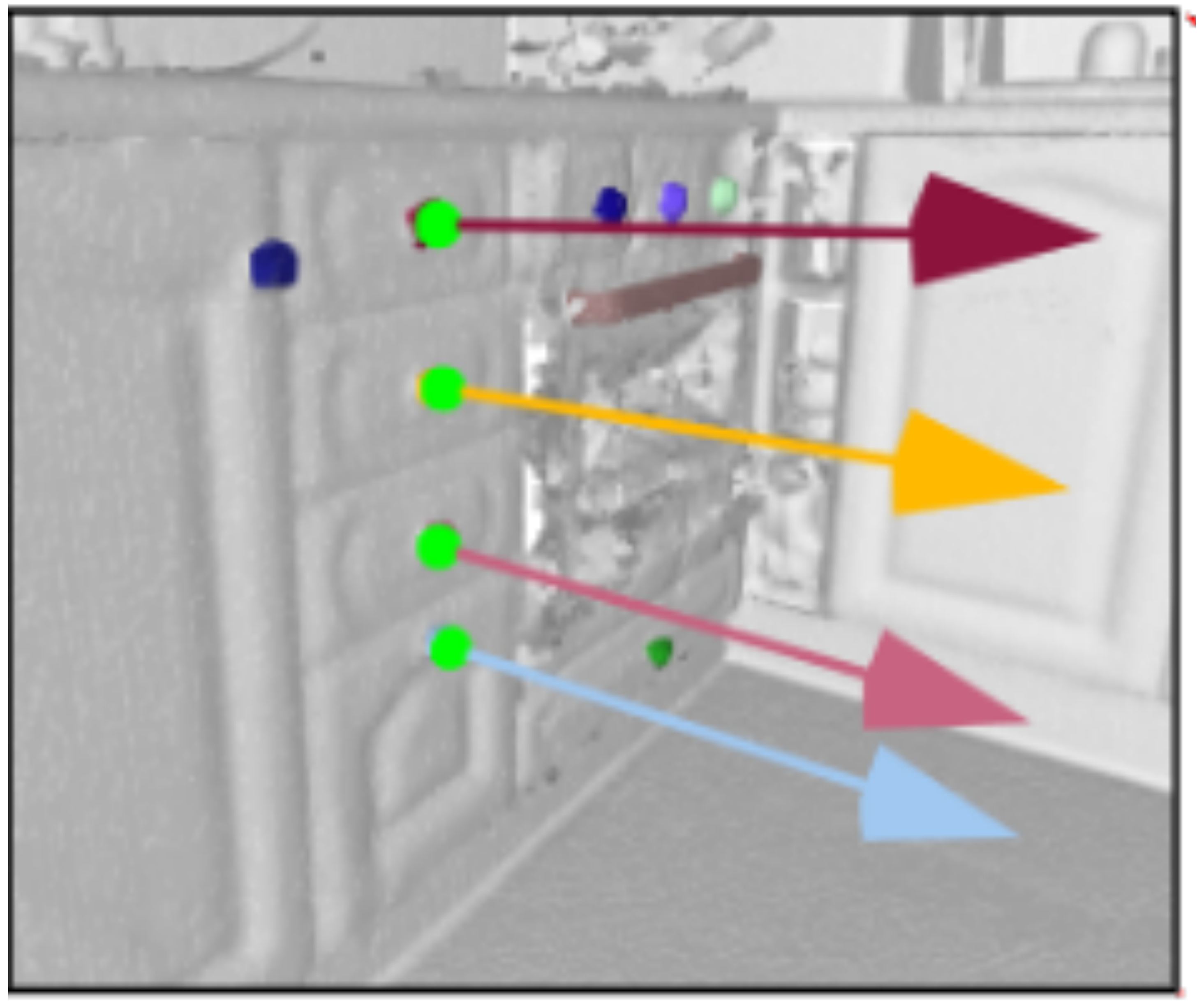



hook_turn hook_pull
key_press plug_in
pinch_pull tip_push
foot_push unplug

## Natural Language Task Descriptions


Open the oven door

## Motion Annotations



[1] Delitzas et al. "SceneFun3D: Fine-grained Functionality and Affordance Understanding in 3D Scenes" CVPR'24 (Oral)

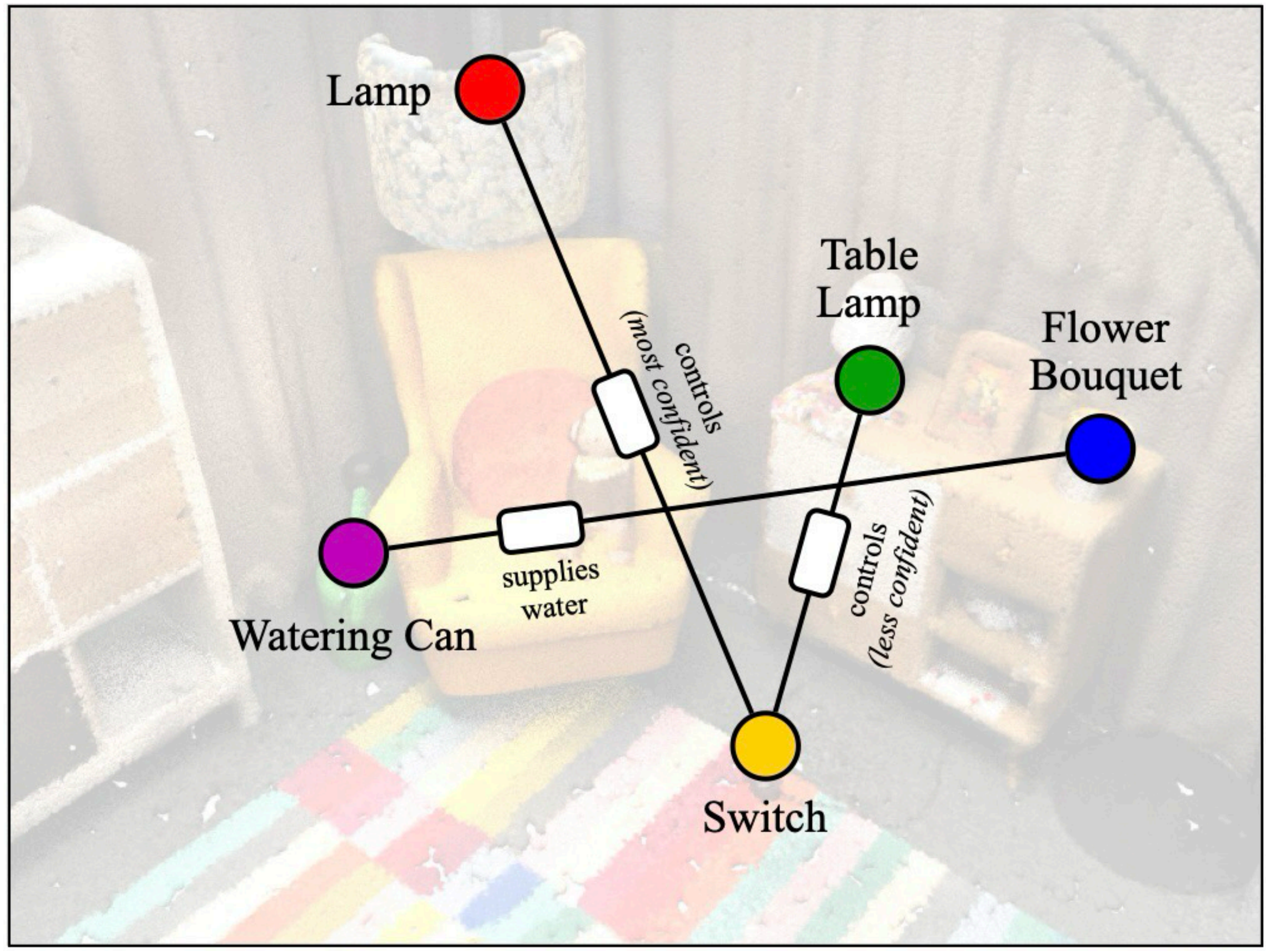
# Towards *Functional* 3D Scene Understanding



[1] Zhang et al. "OpenFunGraph: Open-Vocabulary Functional 3D Scene Graphs for Real-World Indoor Spaces" CVPR'25 (Highlight)

# Towards *Functional* 3D Scene Understanding

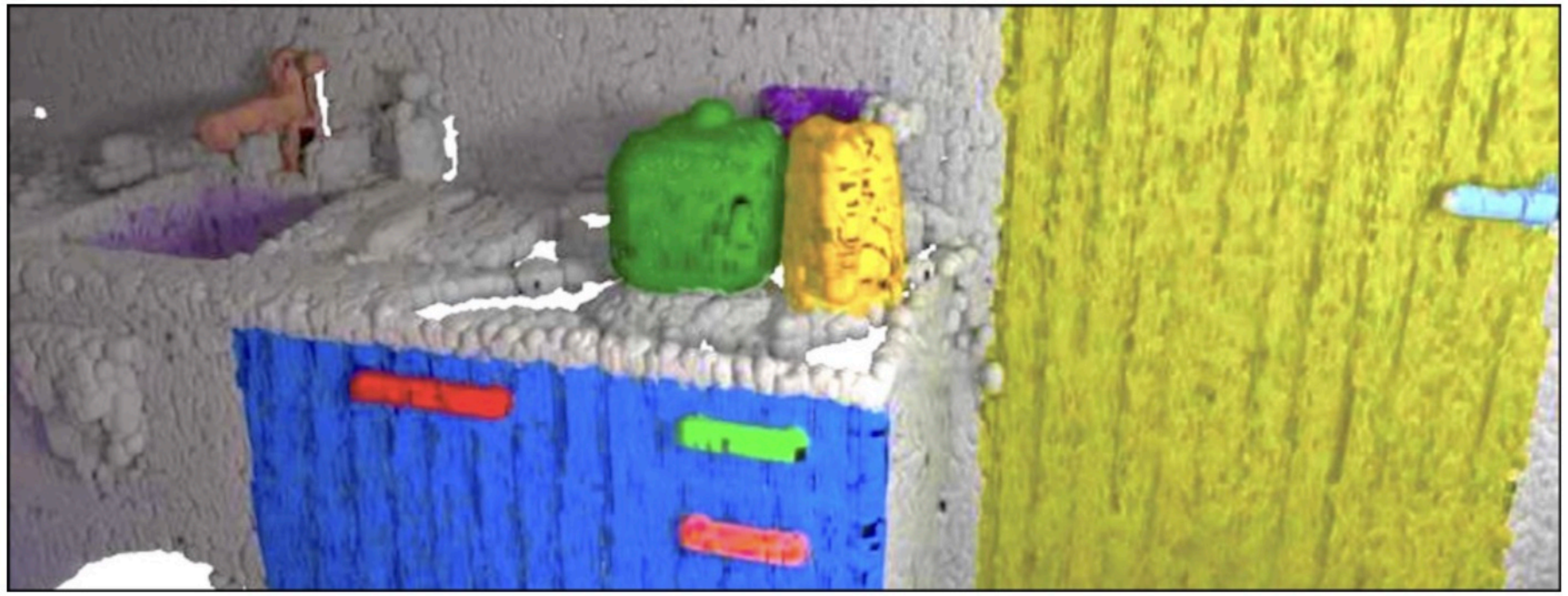
Input: **RGB-D + 3D Reconstruction**



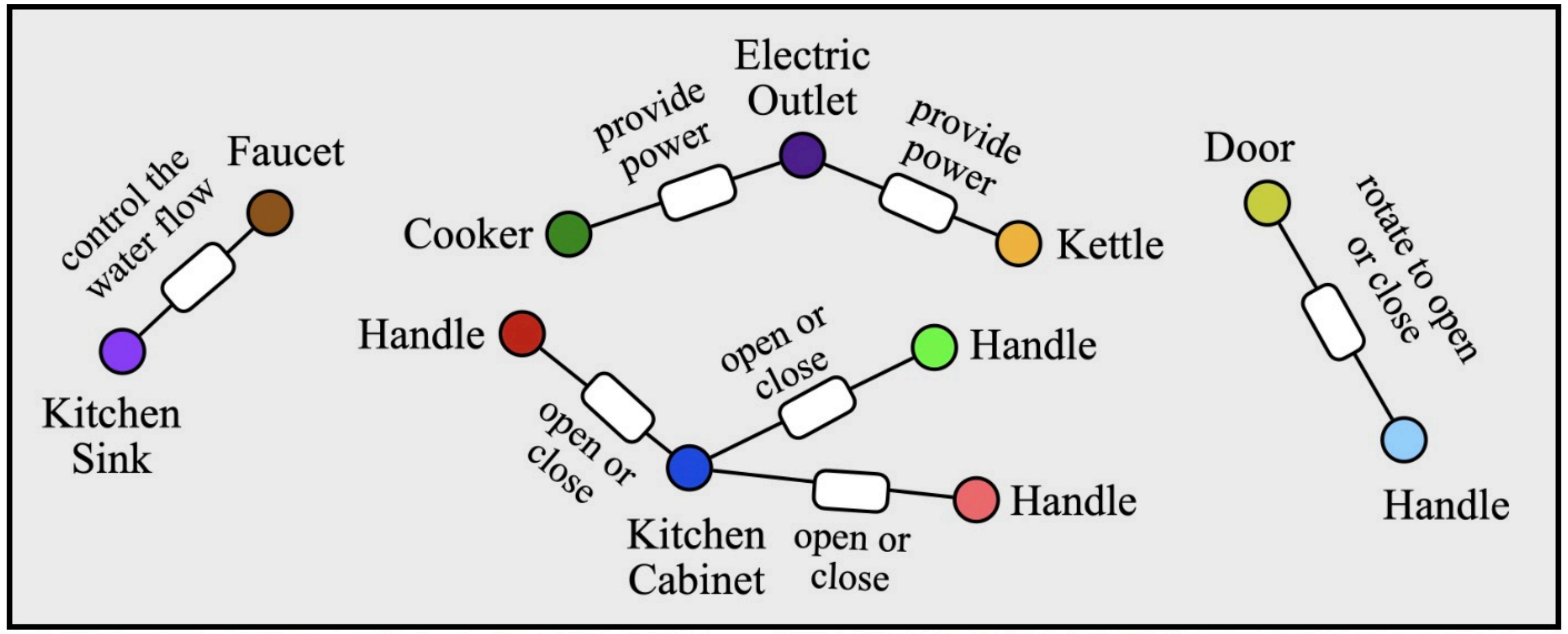

Output: **Functional 3D Scene Graph**

[1] Zhang et al. "OpenFunGraph: Open-Vocabulary Functional 3D Scene Graphs for Real-World Indoor Spaces" CVPR'25 (Highlight)

# Open-Vocabulary Functional 3D Scene Graphs



LiDAR 3D Scans

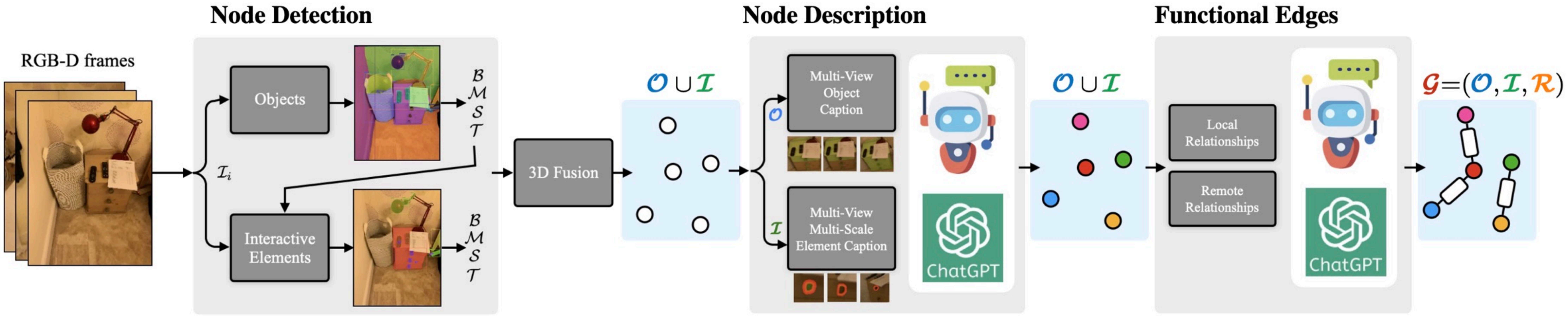



3D Scene Graphs Annotations

[1] Zhang et al. "OpenFunGraph: Open-Vocabulary Functional 3D Scene Graphs for Real-World Indoor Spaces" CVPR'25 (Highlight)

# Open-Vocabulary Functional 3D Scene Graphs

Key idea: Leverage Knowledge from Foundation Models to infer Functionalities



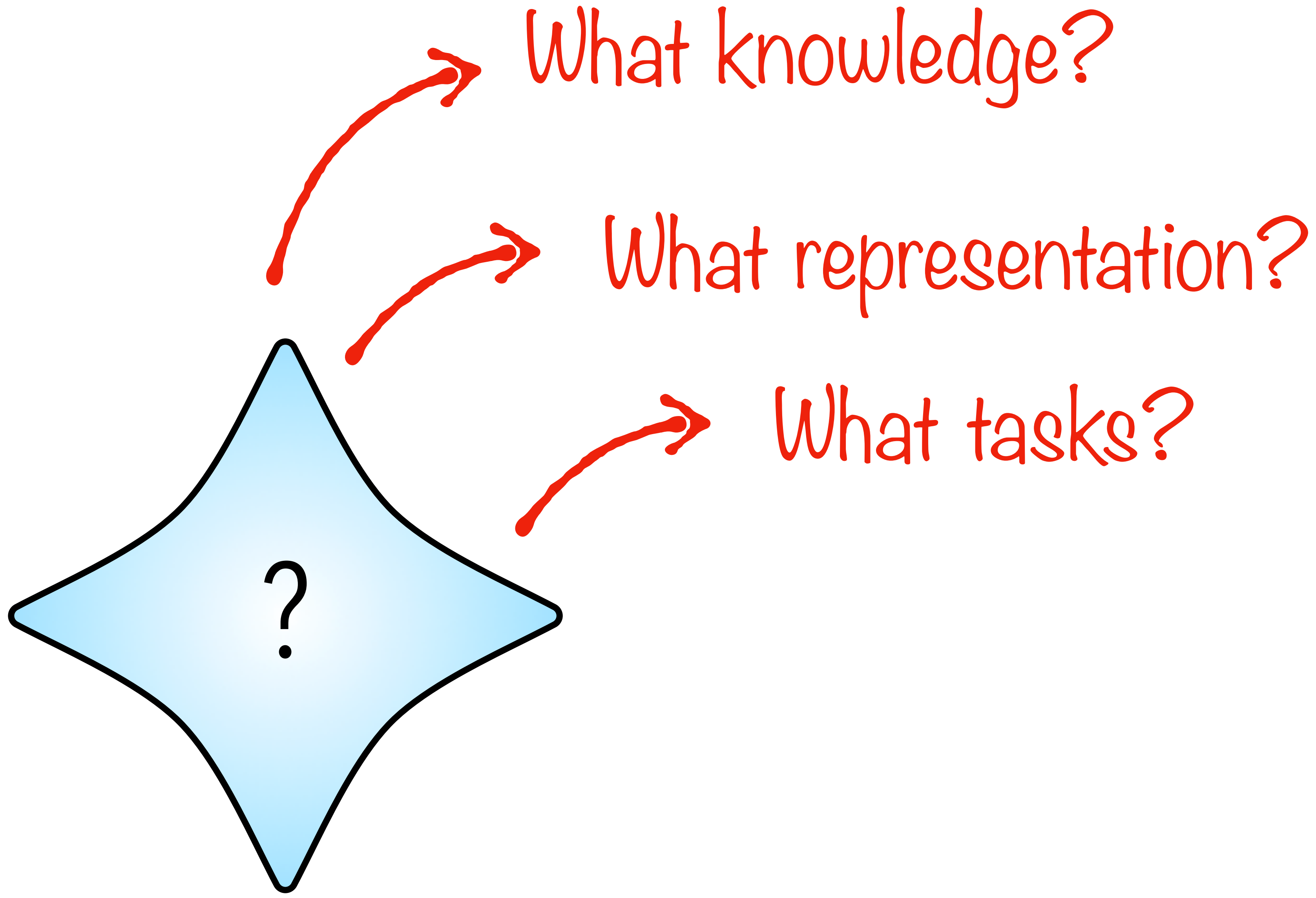

[1] Zhang et al. "OpenFunGraph: Open-Vocabulary Functional 3D Scene Graphs for Real-World Indoor Spaces" CVPR'25 (Highlight)

# *3D Scene Representations*

# 3D Scene Representations

Input: **3D Scene**

**Scene Understanding**




Output: **Extracted Knowledge**

Slide credit: Animation is from Trellis

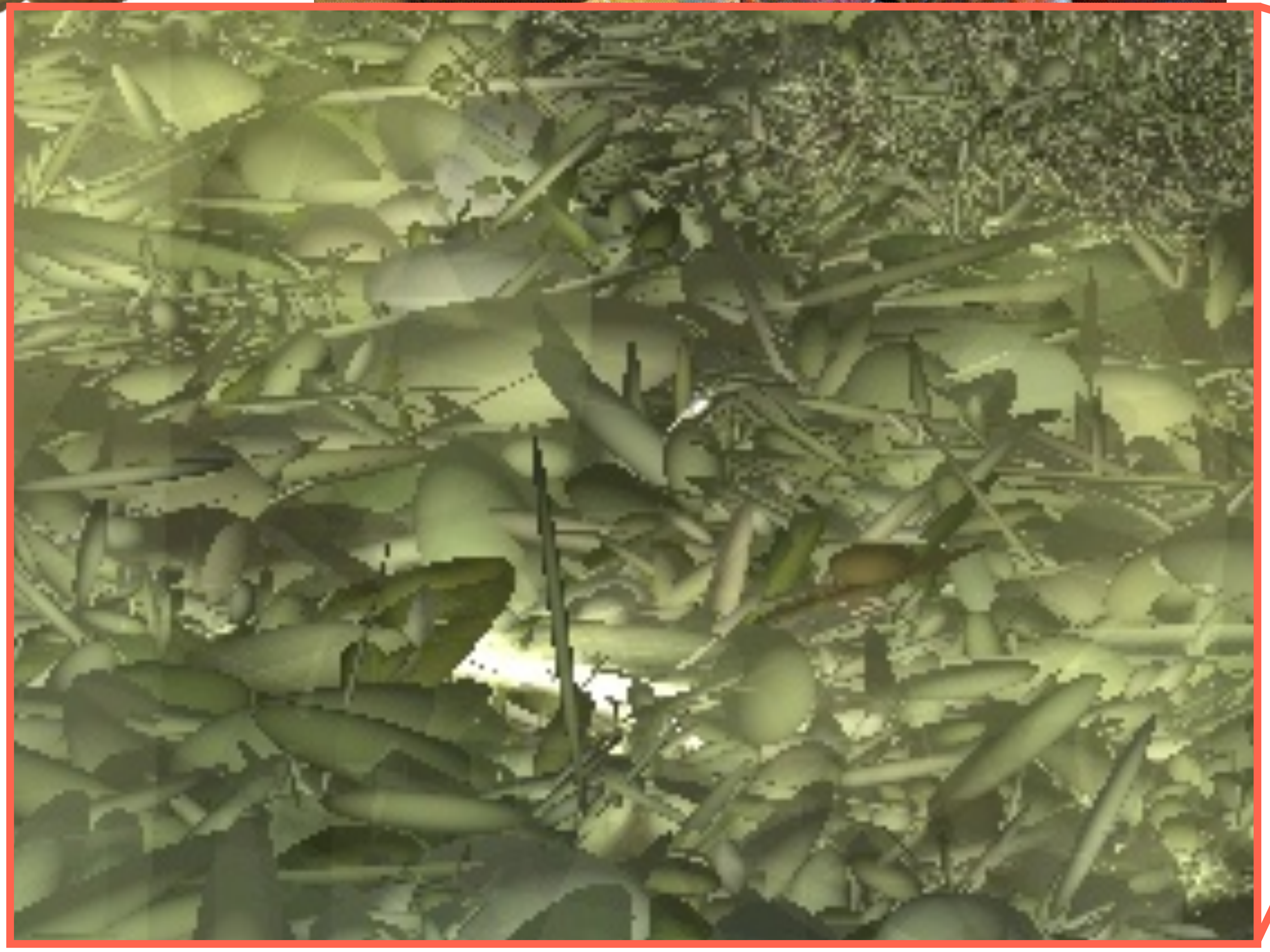
# What makes a good 3D scene representation?

Point Clouds

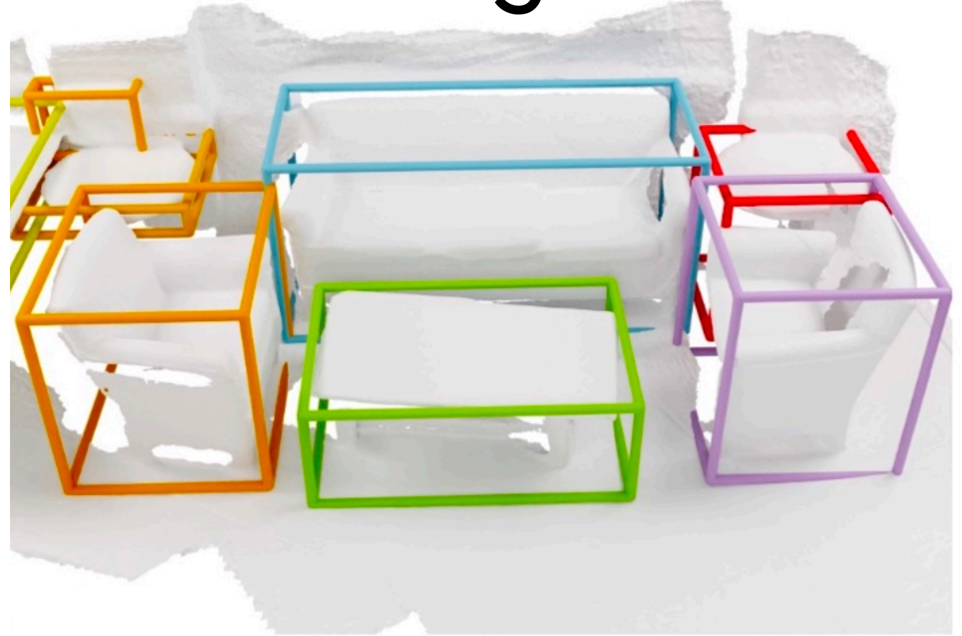
Polygon Meshes

NeRFs

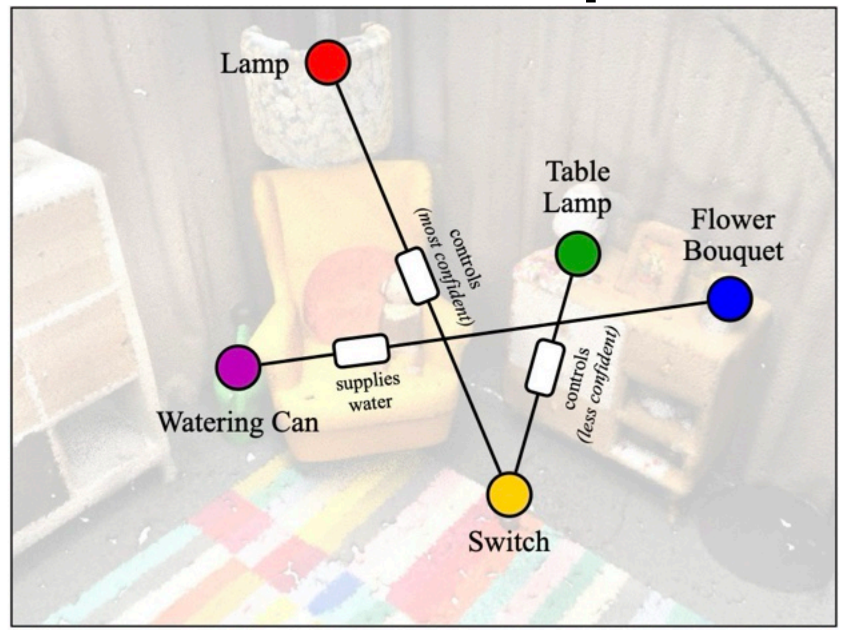
Gaussian Splats



Bounding Boxes



Scene Graphs



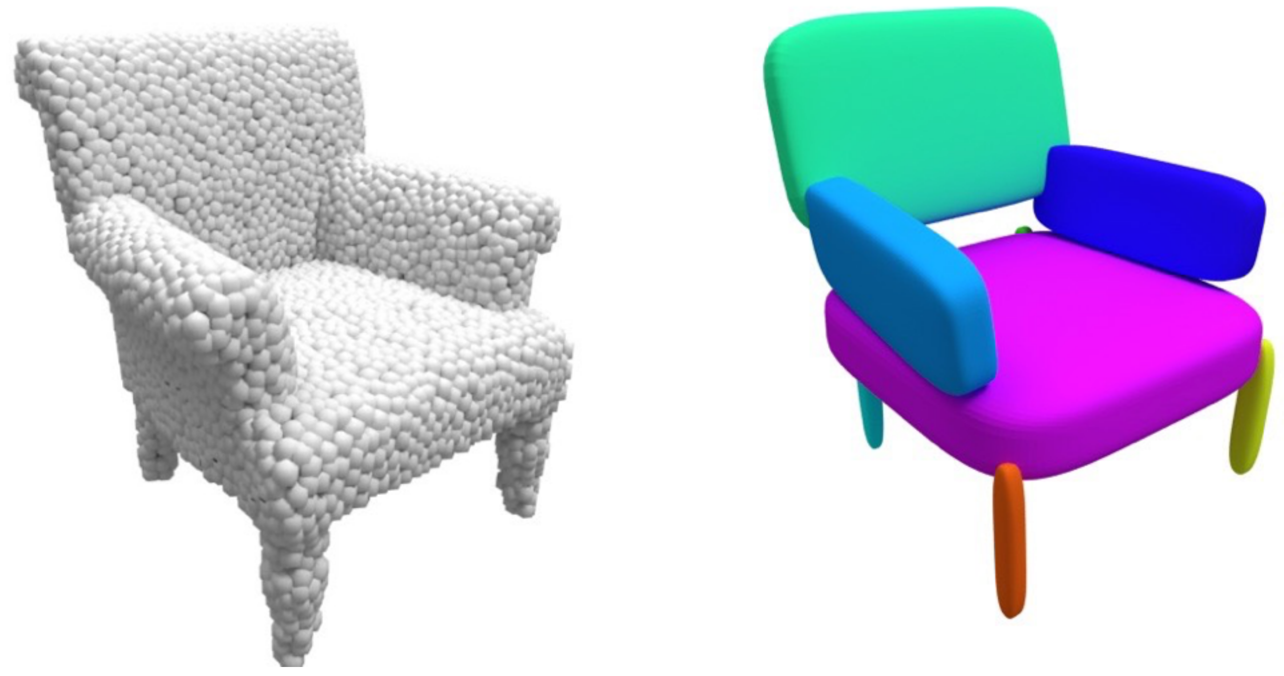
# 3D Scene Representations with Superquadrics

3D Point Cloud
1'000'000 points

Geometric Primitives
300 Superquadrics

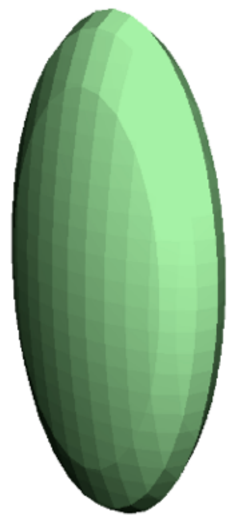
[1] Fedele et al. "SuperDec: 3D Scene Decomposition with Superquadric Primitives" arxiv'25

# 3D Primitive Types



Superquadrics

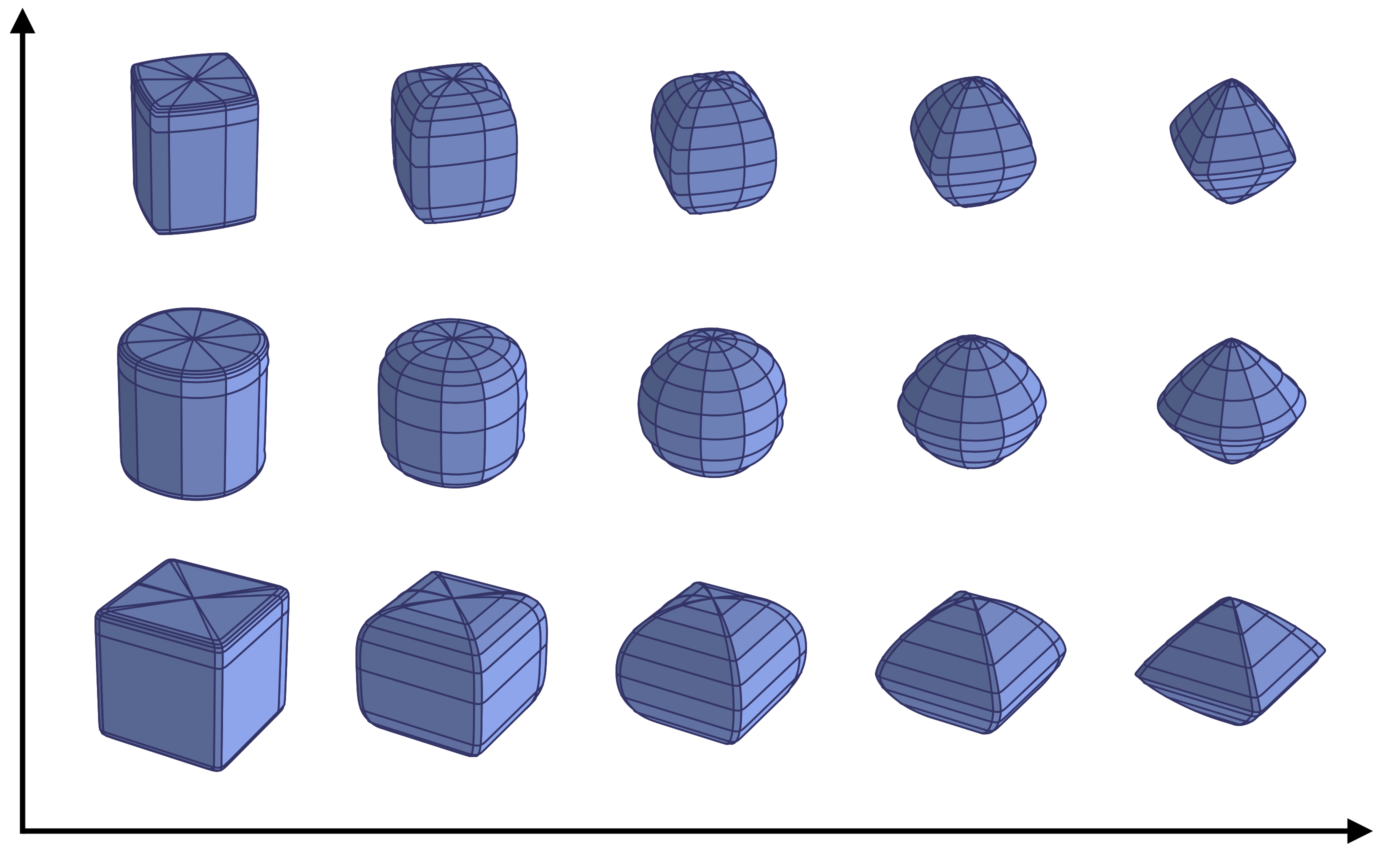
## Ellipsoids / Gaussians



$$f(x,y,z) = \left(\frac{|x|}{a_x}\right)^2 + \left(\frac{|y|}{a_y}\right)^2 + \left(\frac{|z|}{a_z}\right)^2 = 1$$

3 parameters

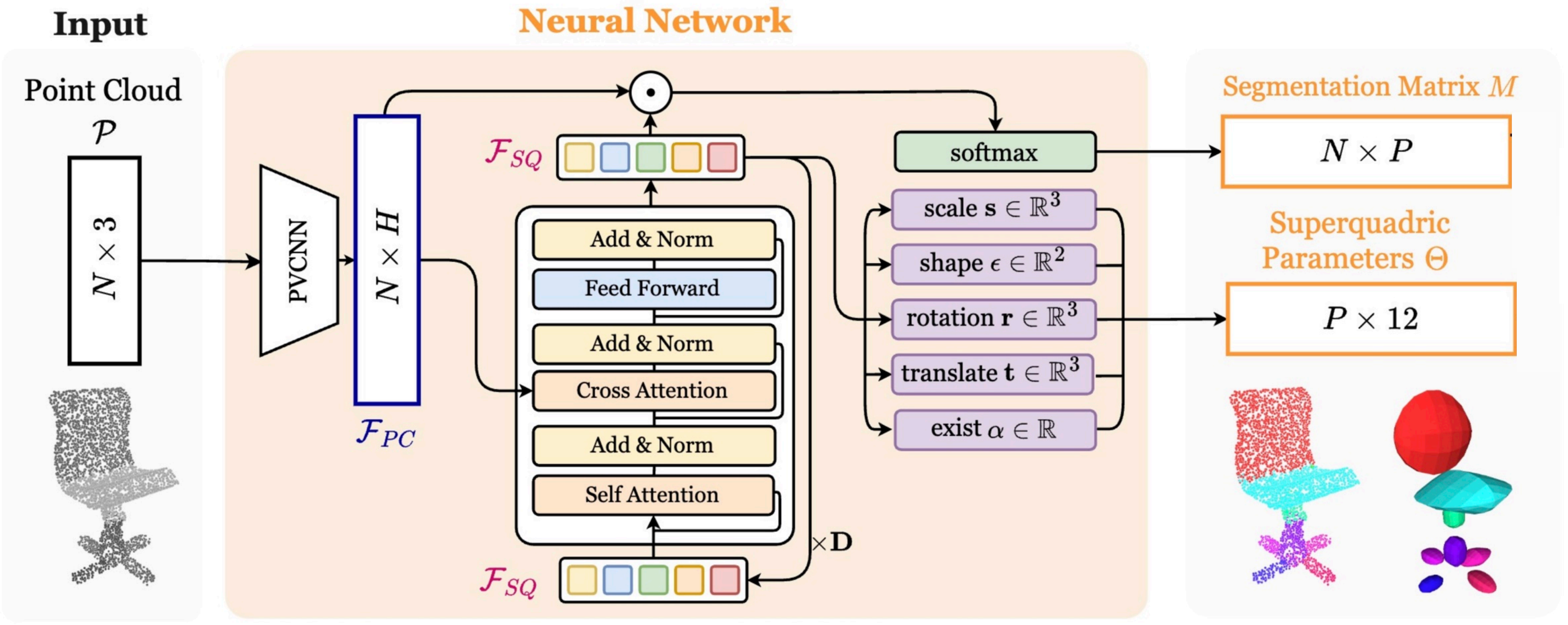
## Superquadrics



$$f(x,y,z) = \left(\left(\frac{|x|}{a_x}\right)^{\frac{2}{\epsilon_2}} + \left(\frac{|y|}{a_y}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{|z|}{a_z}\right)^{\frac{2}{\epsilon_1}} = 1$$

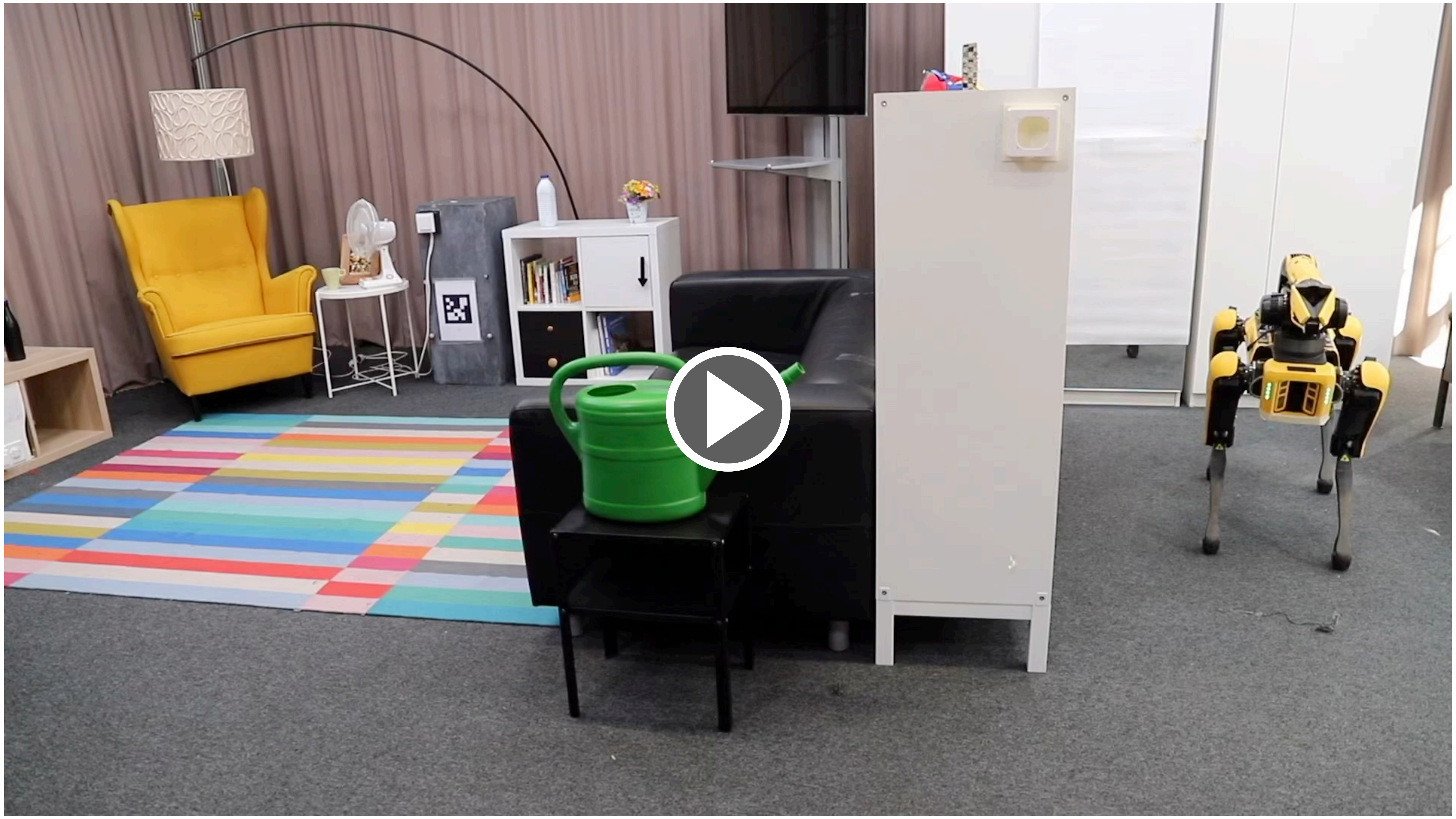
5 parameters

# 3D Scene Decomposition with Superquadrics



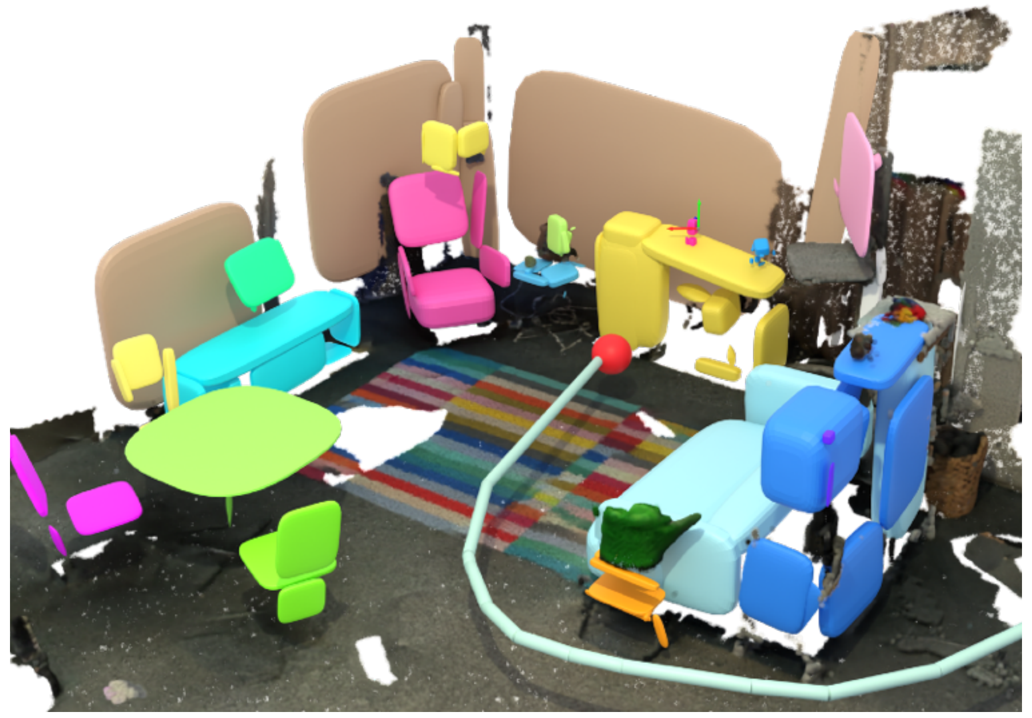

[1] Fedele et al. "SuperDec: 3D Scene Decomposition with Superquadric Primitives" arxiv'25

# 3D Scene Decomposition with Superquadrics



Point Cloud



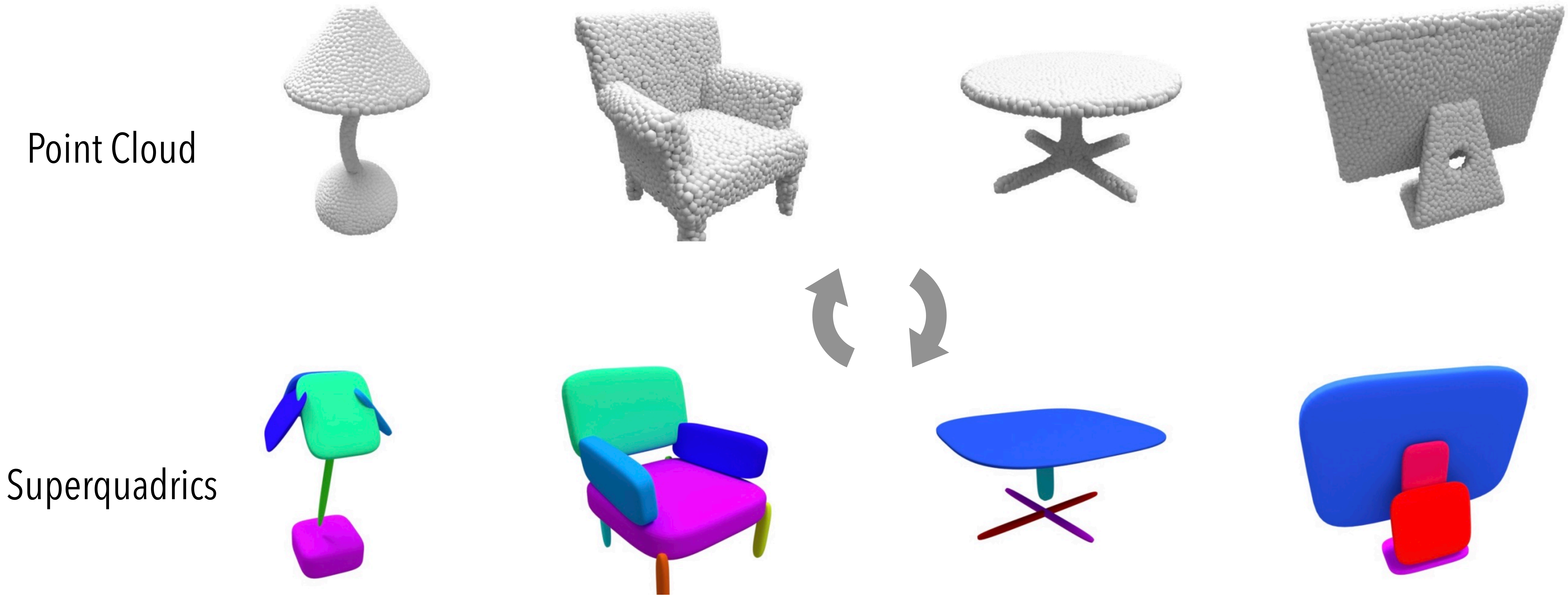



Path Planning

Grasping Pose

[1] Fedele et al. "SuperDec: 3D Scene Decomposition with Superquadric Primitives" arxiv'25

# 3D Scene Decomposition with Superquadrics



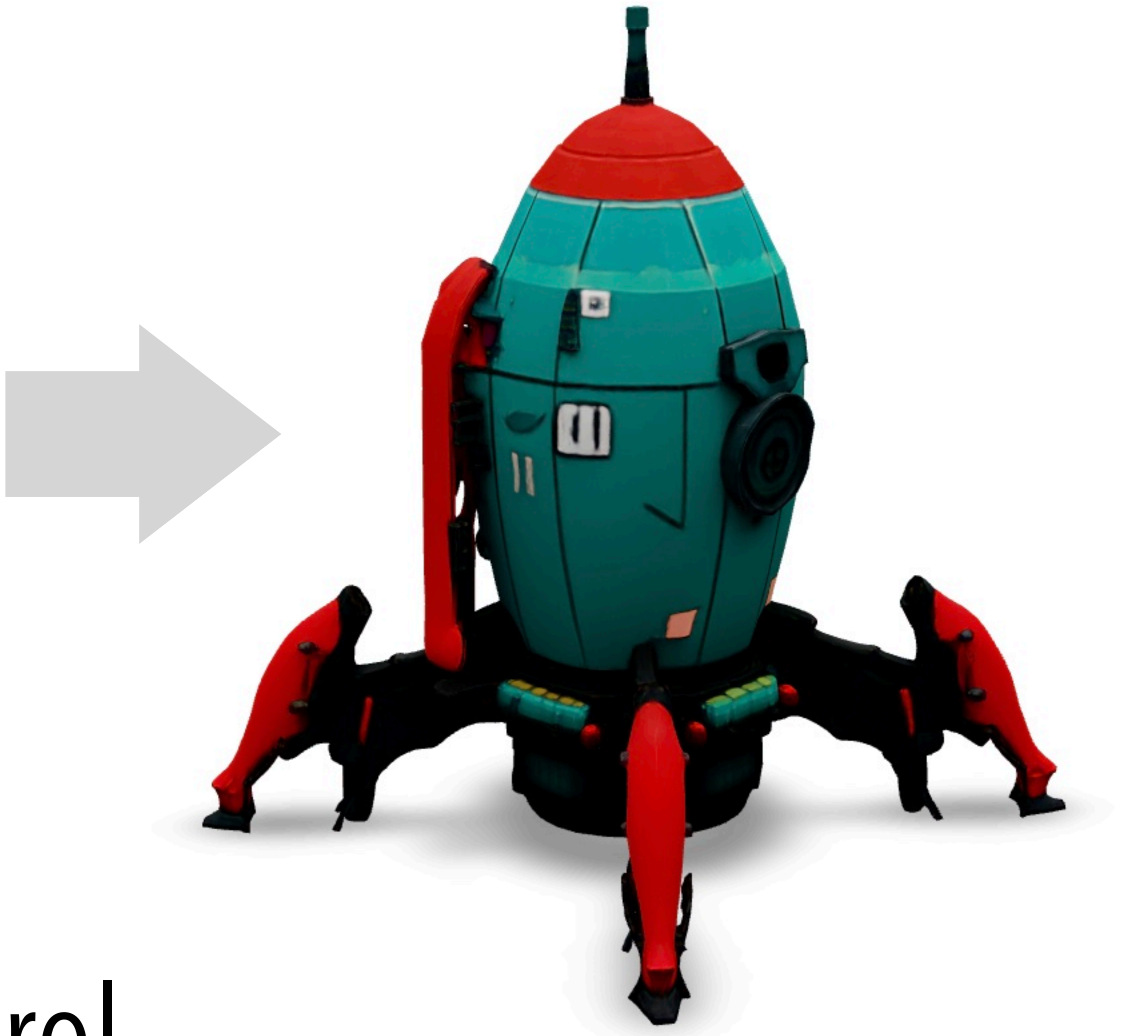

[1] Fedele et al. "SuperDec: 3D Scene Decomposition with Superquadric Primitives" arxiv'25

# *Controllable 3D Generation*

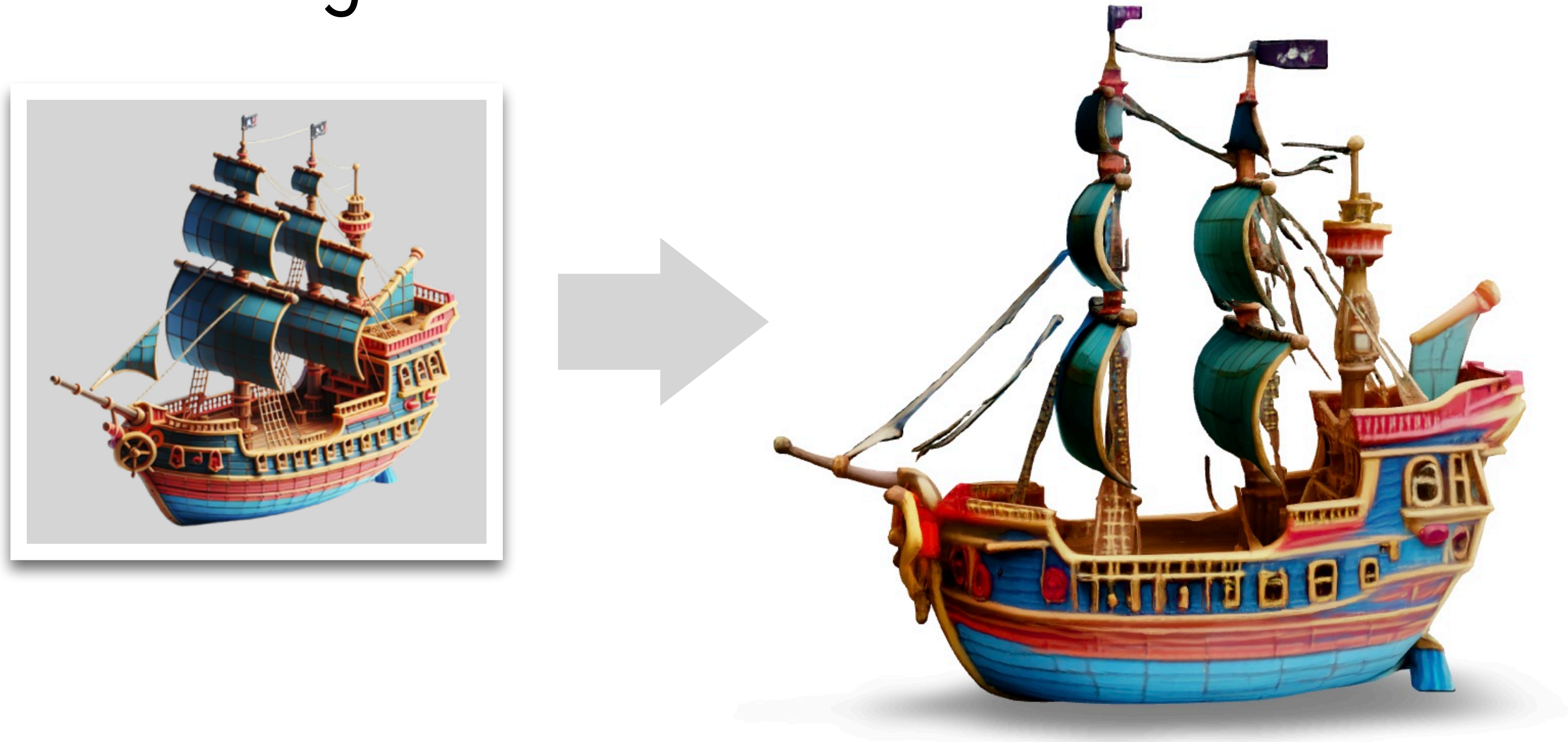
# Controllable 3D Generation

## with Text Control

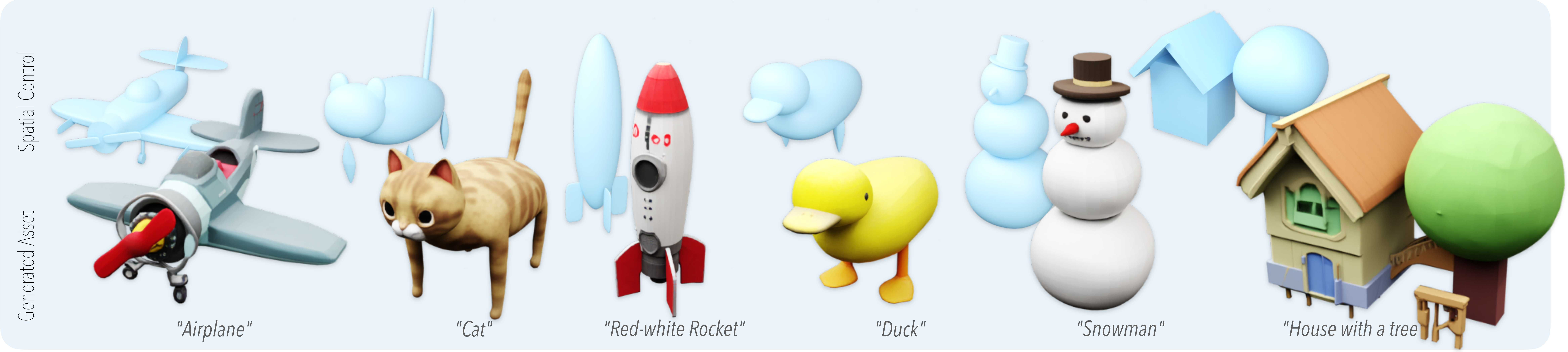
A stylized, cartoonish rocket with a red dome top and black antenna, teal cylindrical middle section with red bands and black connectors.



## with Image Control



## with Spatial Control



# Controllable 3D Generation and Editing

Spatial guidance enables fine-grained control over the object geometry

Alex Delitzas

Elisabetta Fedele

Ayça Takmaz

Yuanwen Yue

Jonas Schult

Rui Huang

# Foundation Models Meet 3D Vision

*Toward Open-World 3D Scene Understanding*
*and Controllable 3D Generation*

Want to work on these topics?
Reach out!

**Francis Engelmann** PostDoc Stanford

Guest Lecture CS231A | June 4th, 2025