Towards Embodied 3D Foundation Models

Zubair Irshad
Research Scientist
Toyota Research Institute

09/8/2024

zubairirshad.com

Bio

- Currently based in Silicon Valley, CA working as a Research Scientist at TRI
- PhD in ME from Georgia Tech
- Training AI and deep models since 6+ years
- Fulbright Scholar
- Various industry experiences
- Publications/Patents/Open-Source Contributions

Agenda

- Recent 3D Representations – 10 mins
- What are foundation models? – 5 mins
- How to build towards 3D foundation models– 25 mins
- Wrap up / Q&A – 10-15 mins

Part 1: Recent 3D Representations

What are Neural Fields or NeRFs?

Approach to transform 2D pictures into 3D Scenes

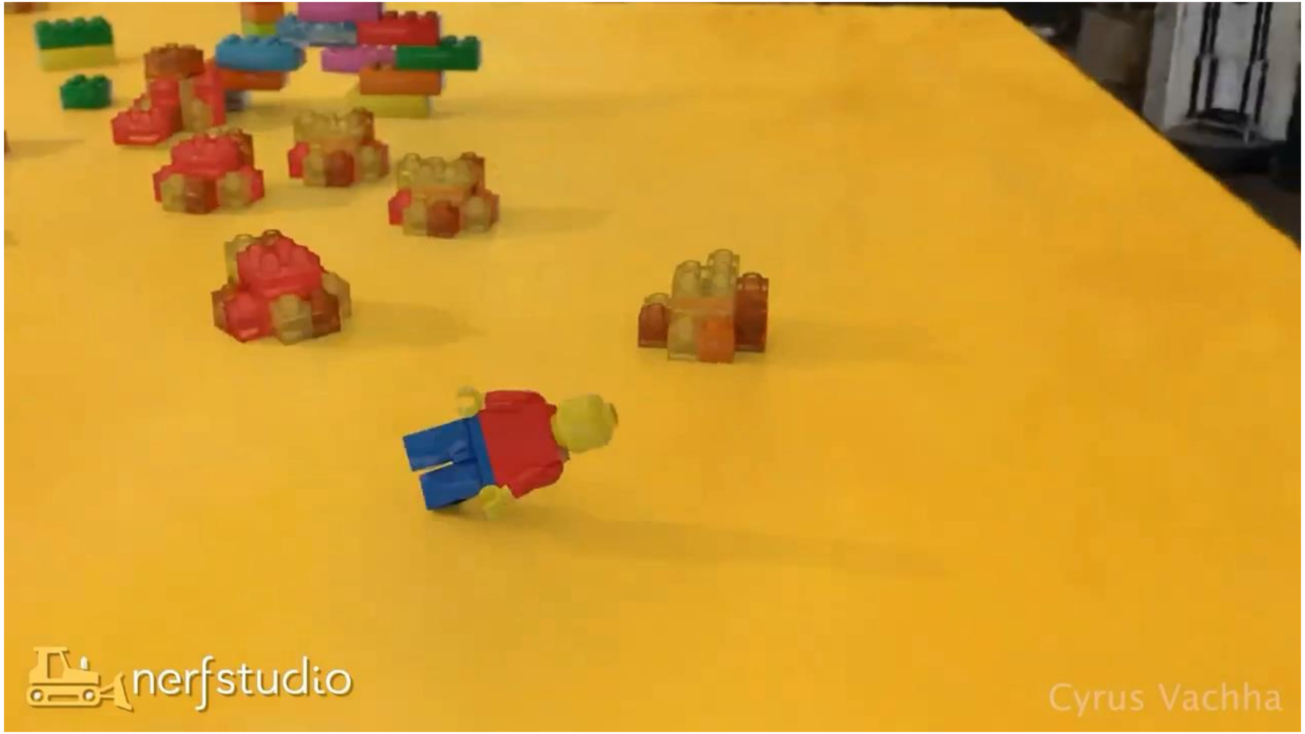
Tancick et al, BlockNeRF, CVPR 2022

Applications

Scene Understanding for Outdoor Scenes

Irshad et al, NeO 360, ICCV 2023

Visual Effects

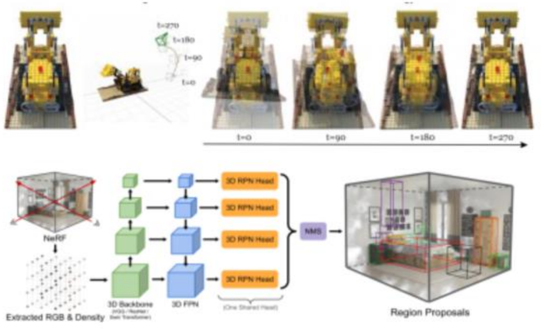


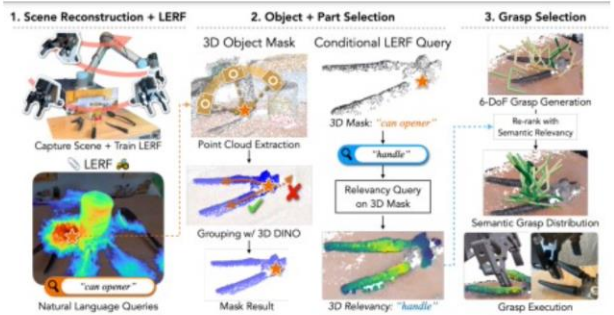
NeRFStudio VFX

Language guided 3D Querying

J. Kerr et al. LeRF, ICCV 2023

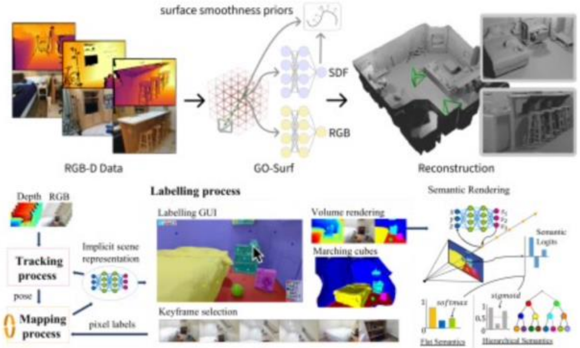
Robotics



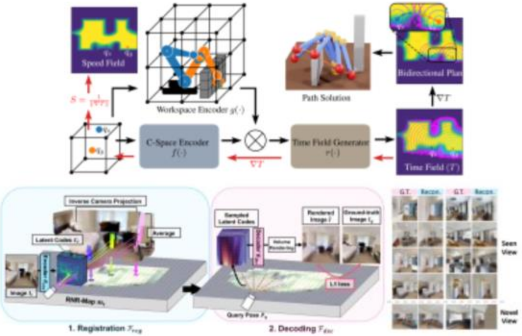


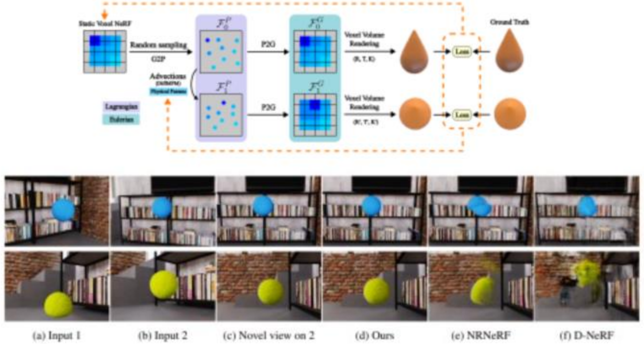
Object Pose Estimation

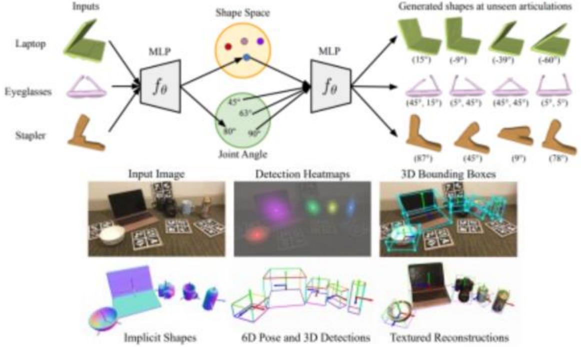
Manipulation/RL



SLAM







Navigation

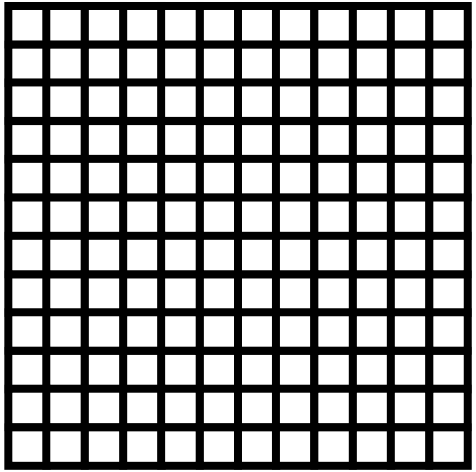
Physics

Object Reconstruction

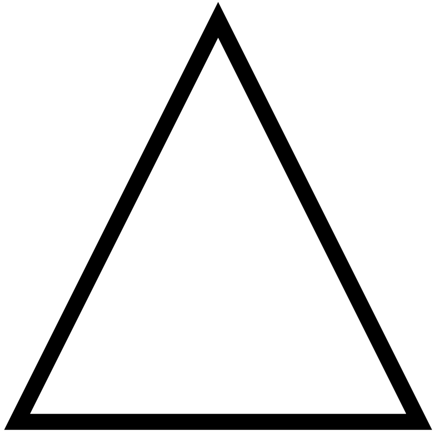
Let’s go back to 2010 on how we were understanding 3D back then

Truncated Signed Distance Function (TSDF)

Truncated Signed Distance Function (TSDF)



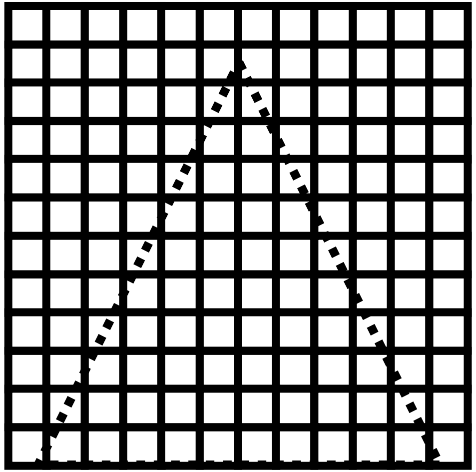
Initialized Grid



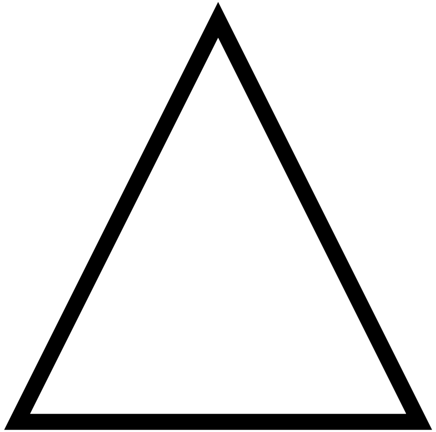
Target Geometry

Slide modified from Mathew Tancik’s talk

Truncated Signed Distance Function (TSDF)



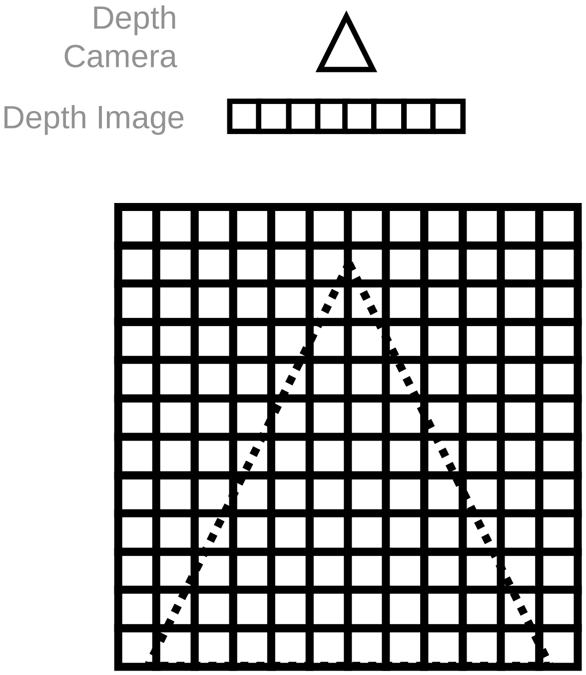
Initialized Grid



Target Geometry

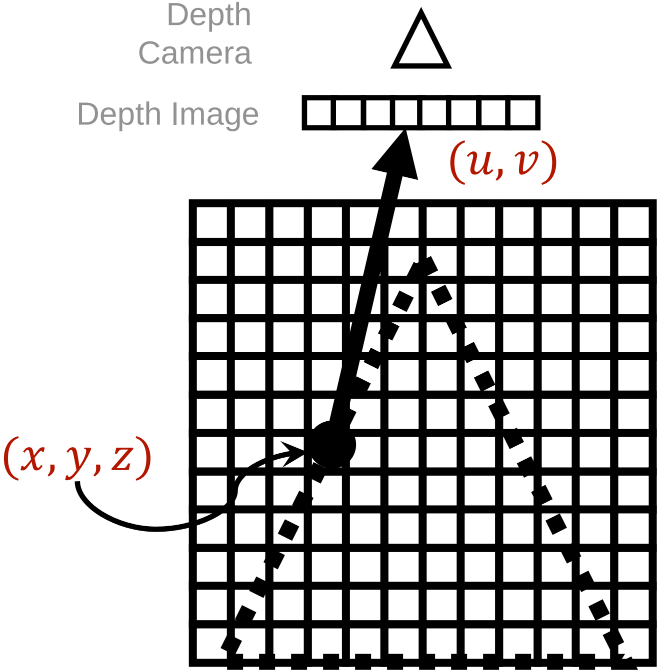
Slide modified from Mathew Tancik’s talk

Truncated Signed Distance Function (TSDF)



TSDF Update

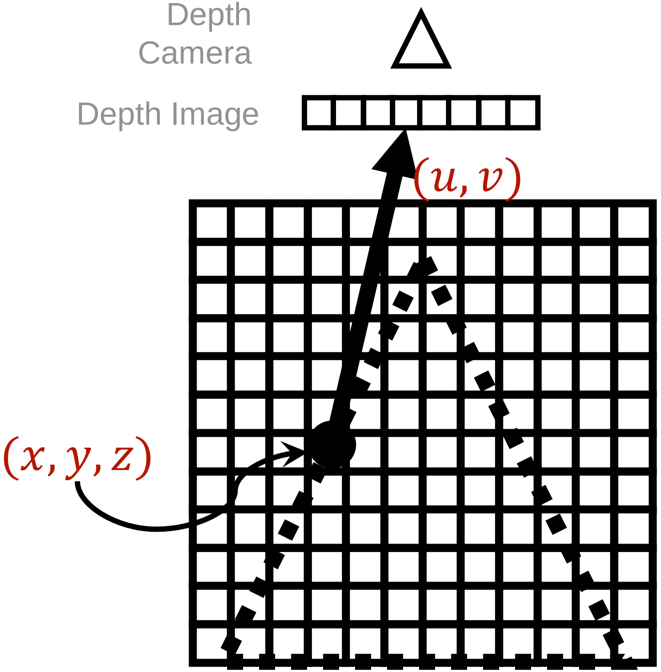
Truncated Signed Distance Function (TSDF)



TSDF Update

- For each 3D voxel location in the *camera coordinate* (x, y, z):

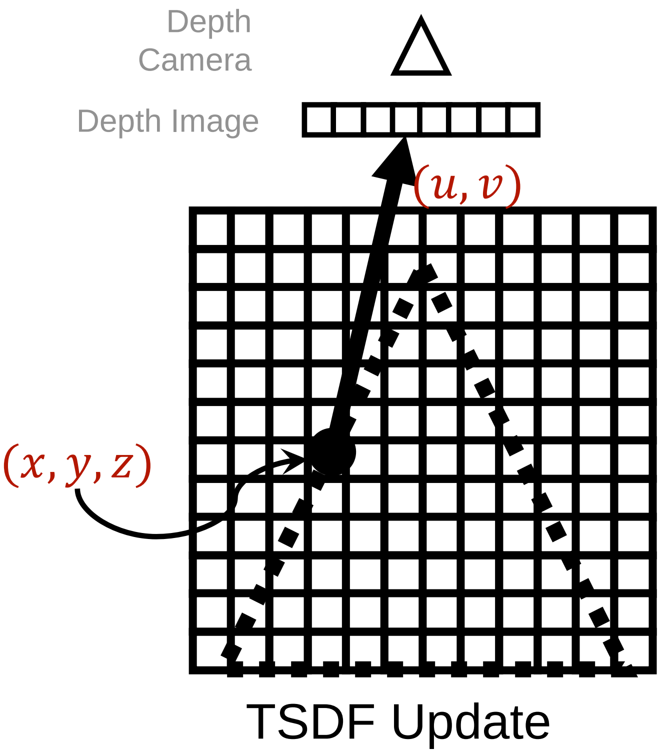
Truncated Signed Distance Function (TSDF)



TSDF Update

- For each 3D voxel location (x, y, z) in the *camera coordinate*:
  - Project (x, y, z) to 2D pixel (u, v).

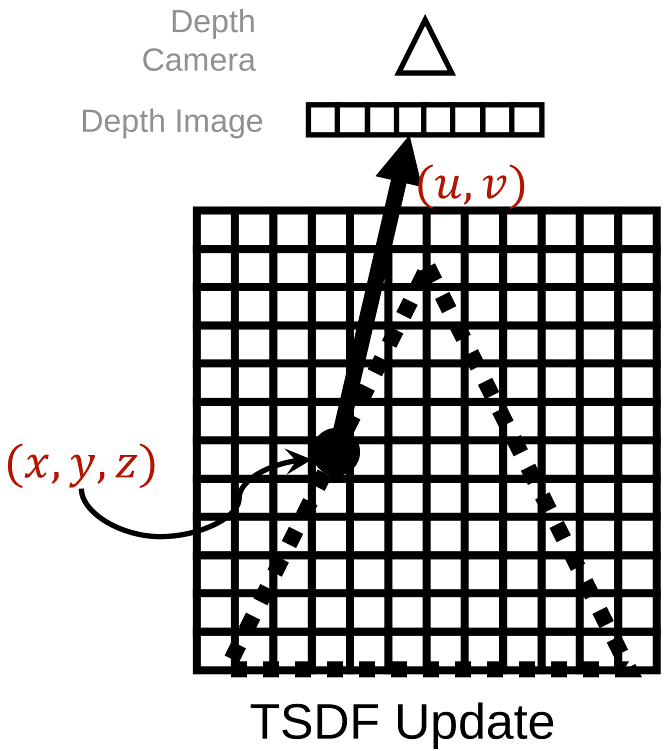
Truncated Signed Distance Function (TSDF)



- For each 3D voxel location (x, y, z) in the *camera coordinate*:
  - Project (x, y, z) to 2D pixel (u, v).
  - Read the depth value $d(u, v)$ at pixel (u, v).
  -
  -

Slide modified from Mathew Tancik's talk

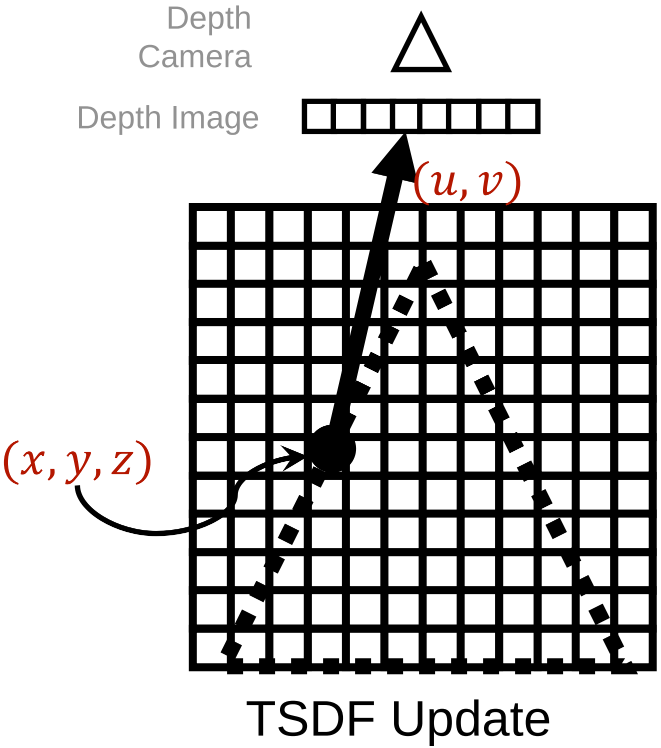
Truncated Signed Distance Function (TSDF)



- For each 3D voxel location (x, y, z) in the *camera coordinate*:
  - Project (x, y, z) to 2D pixel (u, v).
  - Read the depth value $d(u, v)$ at pixel (u, v).
  - Compute $d_{proj} = d(u, v) - z$.
  -

Slide modified from Mathew Tancik's talk

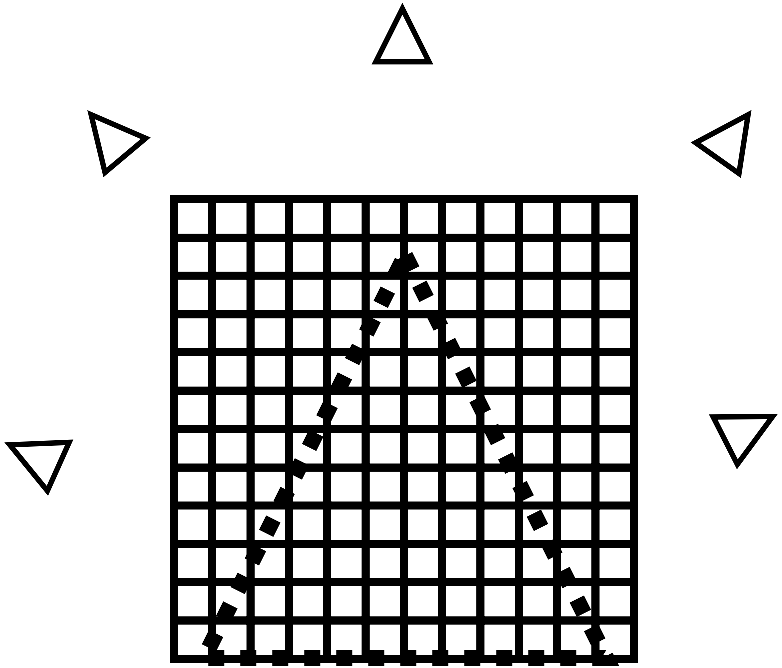
Truncated Signed Distance Function (TSDF)



- For each 3D voxel location (x, y, z) in the *camera coordinate*:
  - Project (x, y, z) to 2D pixel (u, v).
  - Read the depth value $d(u, v)$ at pixel (u, v).
  - Compute $d_{proj} = d(u, v) - z$.
  - Normalize, truncate, and update the value stored in the voxel if $|d_{proj}|$ is smaller.

Slide modified from Mathew Tancik's talk

Truncated Signed Distance Function (TSDF)

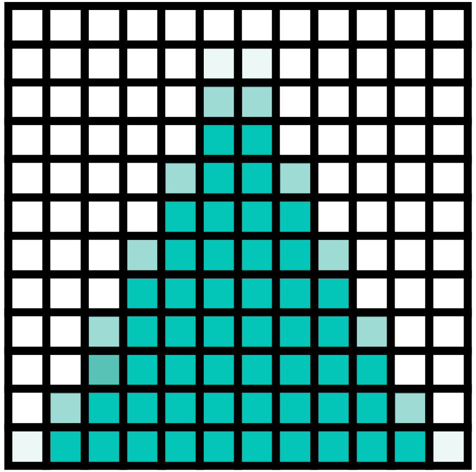


Multi-view Observations

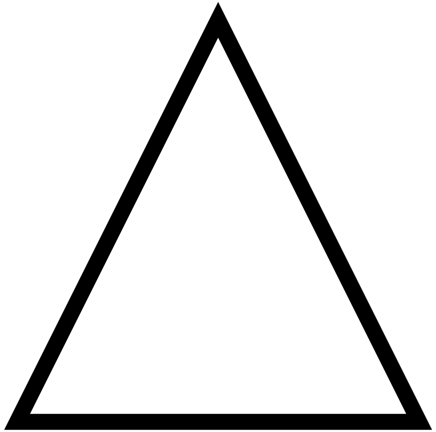
- For each 3D voxel location (x, y, z) in the *camera coordinate*:
  - Project (x, y, z) to 2D pixel (u, v).
  - Read the depth value $d(u, v)$ at pixel (u, v).
  - Compute $d_{proj} = d(u, v) - z$.
  - Normalize, truncate, and update the value stored in the voxel.

Slide modified from Mathew Tancik's talk

Truncated Signed Distance Function (TSDF)



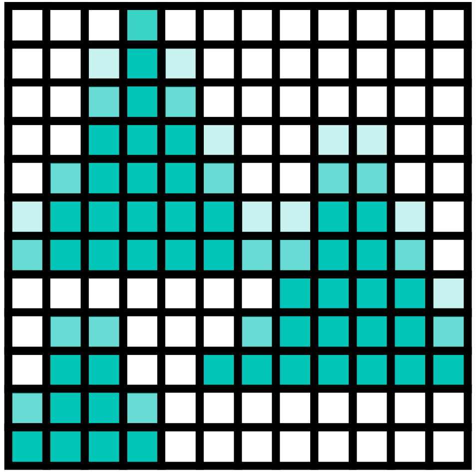
Reconstruction



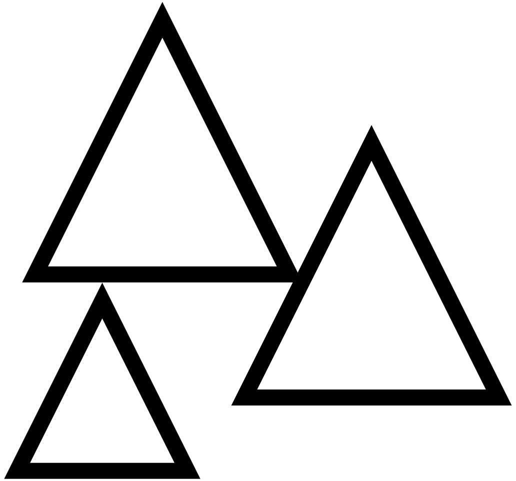
Target Geometry

Slide modified from Mathew Tancik's talk

Truncated Signed Distance Function (TSDF)



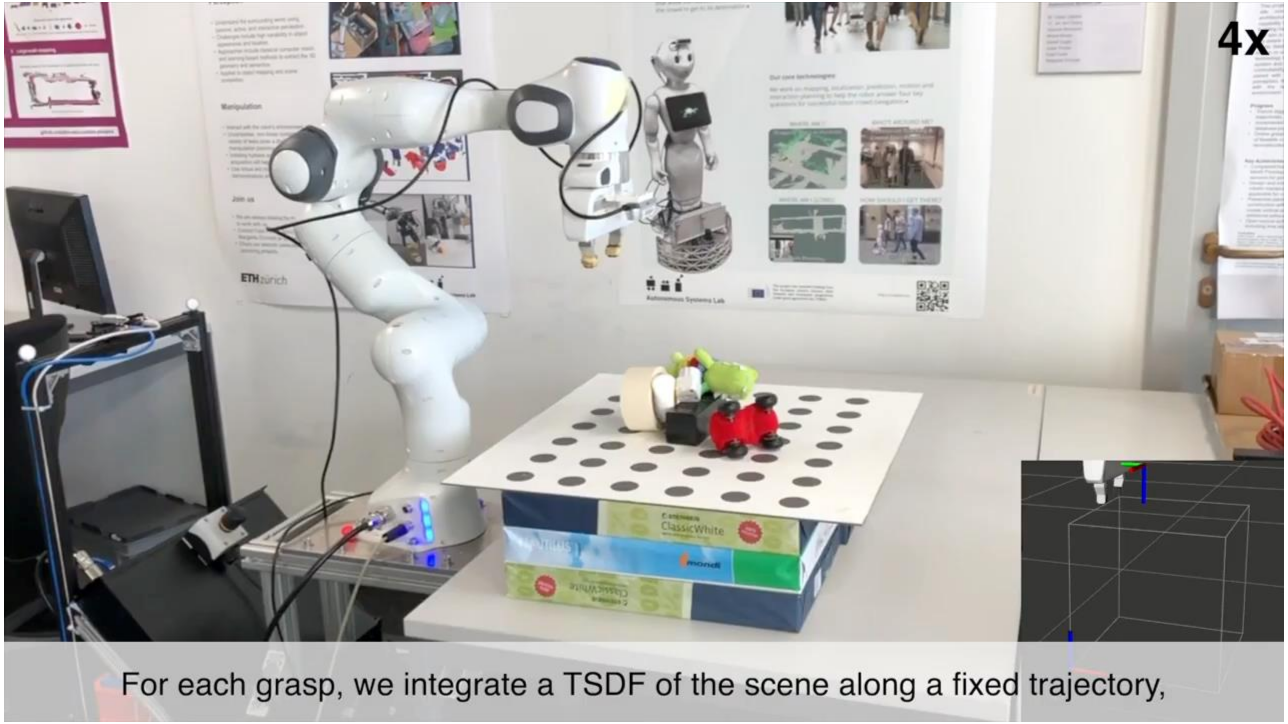
Reconstruction



Target Geometry

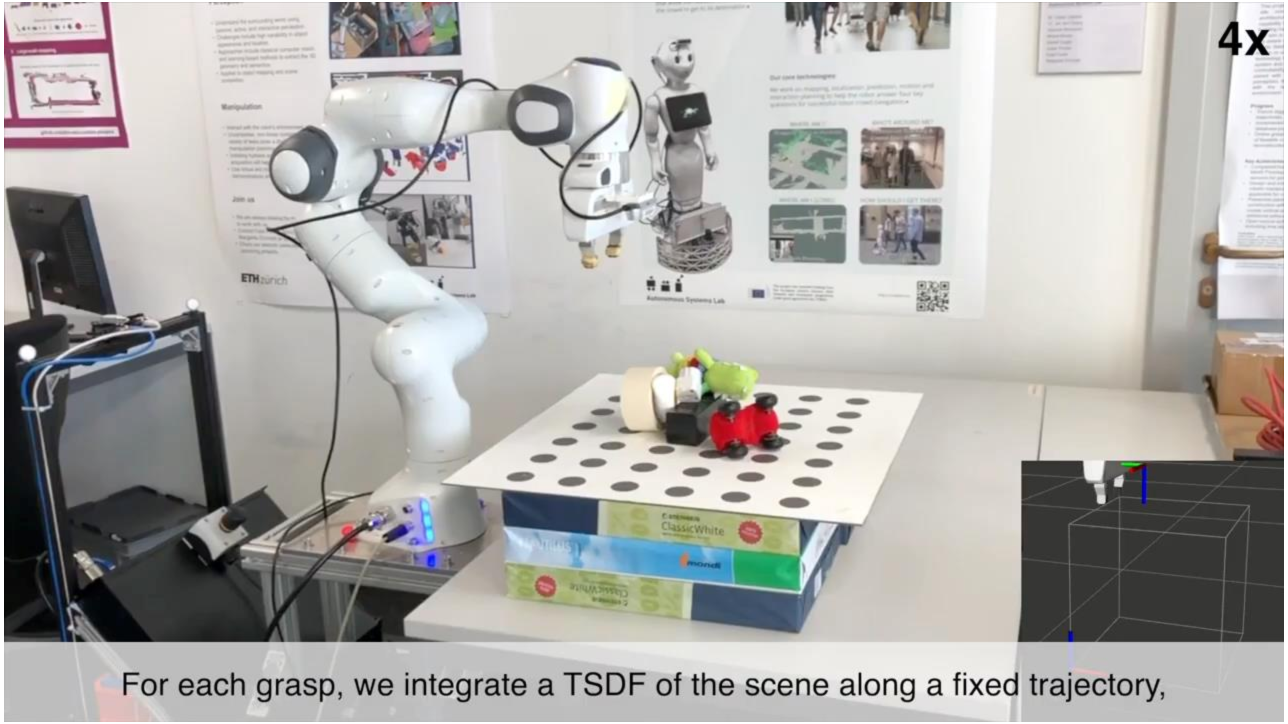
Slide modified from Mathew Tancik's talk

Truncated Signed Distance Function (TSDF)



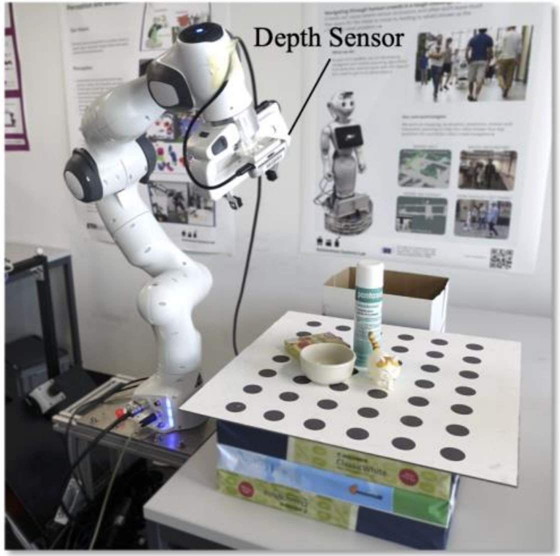
Slide modified from Mathew Tancik's talk

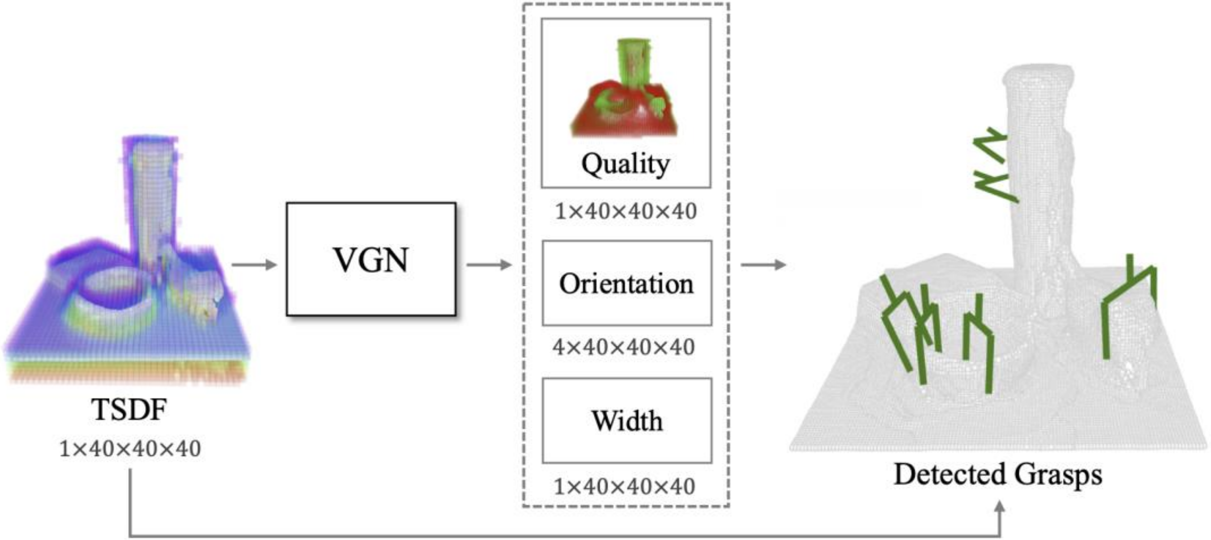
TSDF for Robotics Grasping



VGN, Breyer et al.

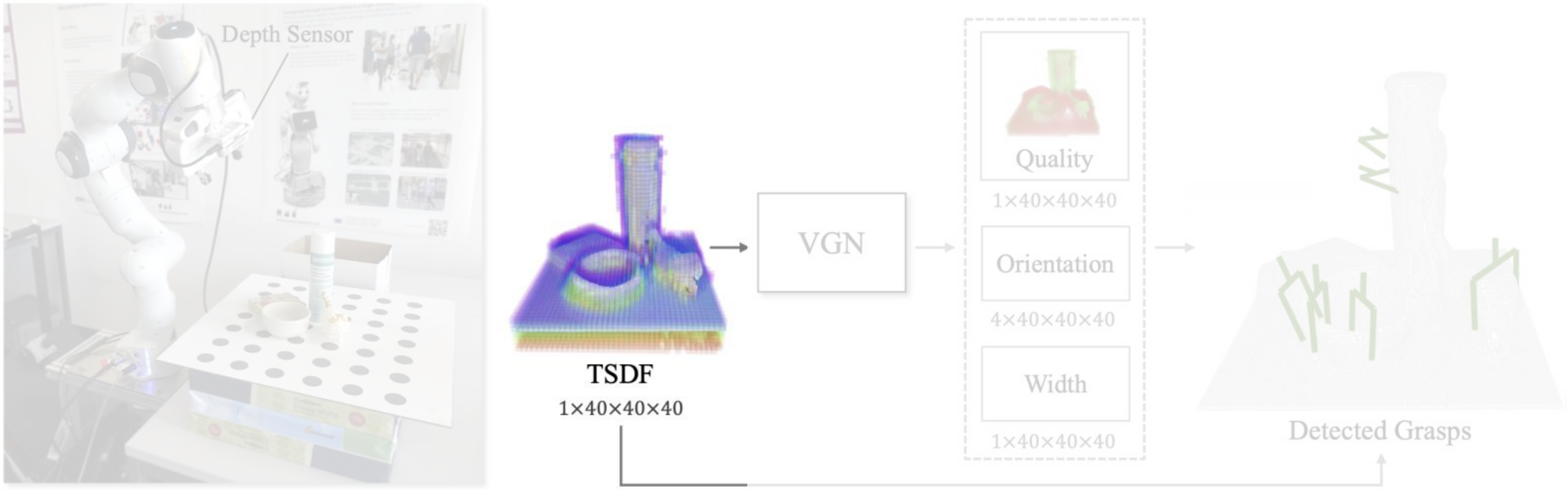
TSDF for Robotics Grasping





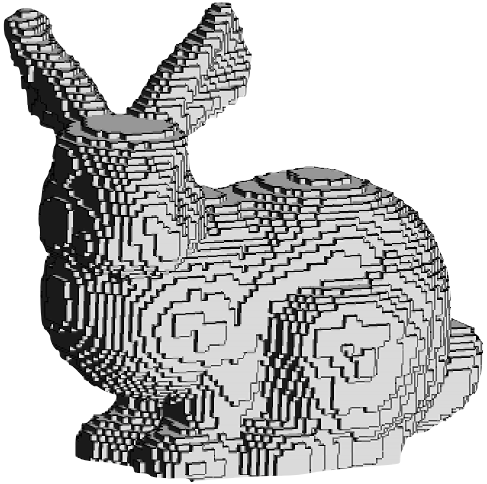
VGN, Breyer et al.

TSDF for Robotics Grasping



VGN, Breyer et al.

Search for a better 3D Representation

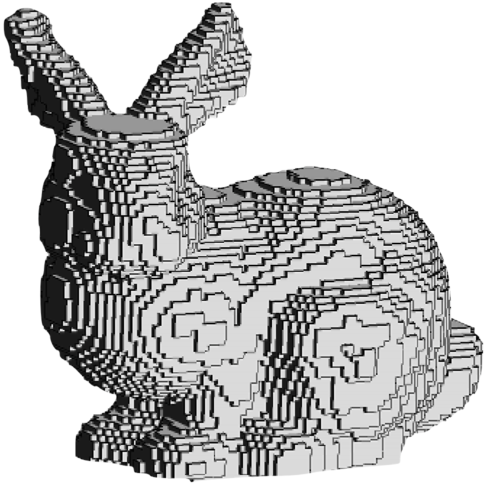


Voxel

Easy to optimize
Large memory footprint

Slide modified from Mathew Tancik's talk

Is there a better Solution?



Voxel

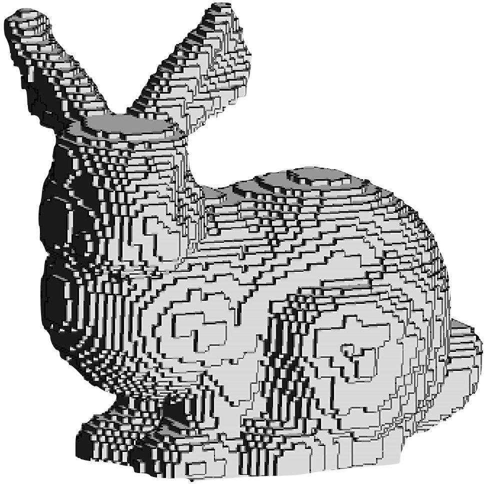
?

Easy to optimize
Large memory footprint

Easy to optimize
Small memory footprint

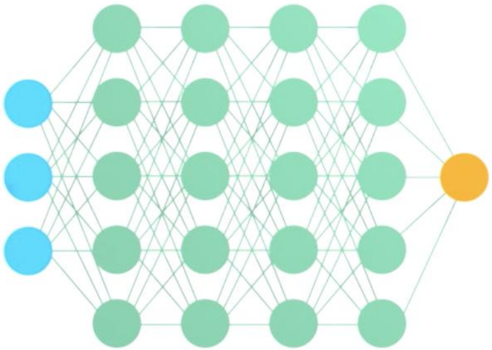
Slide modified from Mathew Tancik's talk

Implicit Representation



Voxel

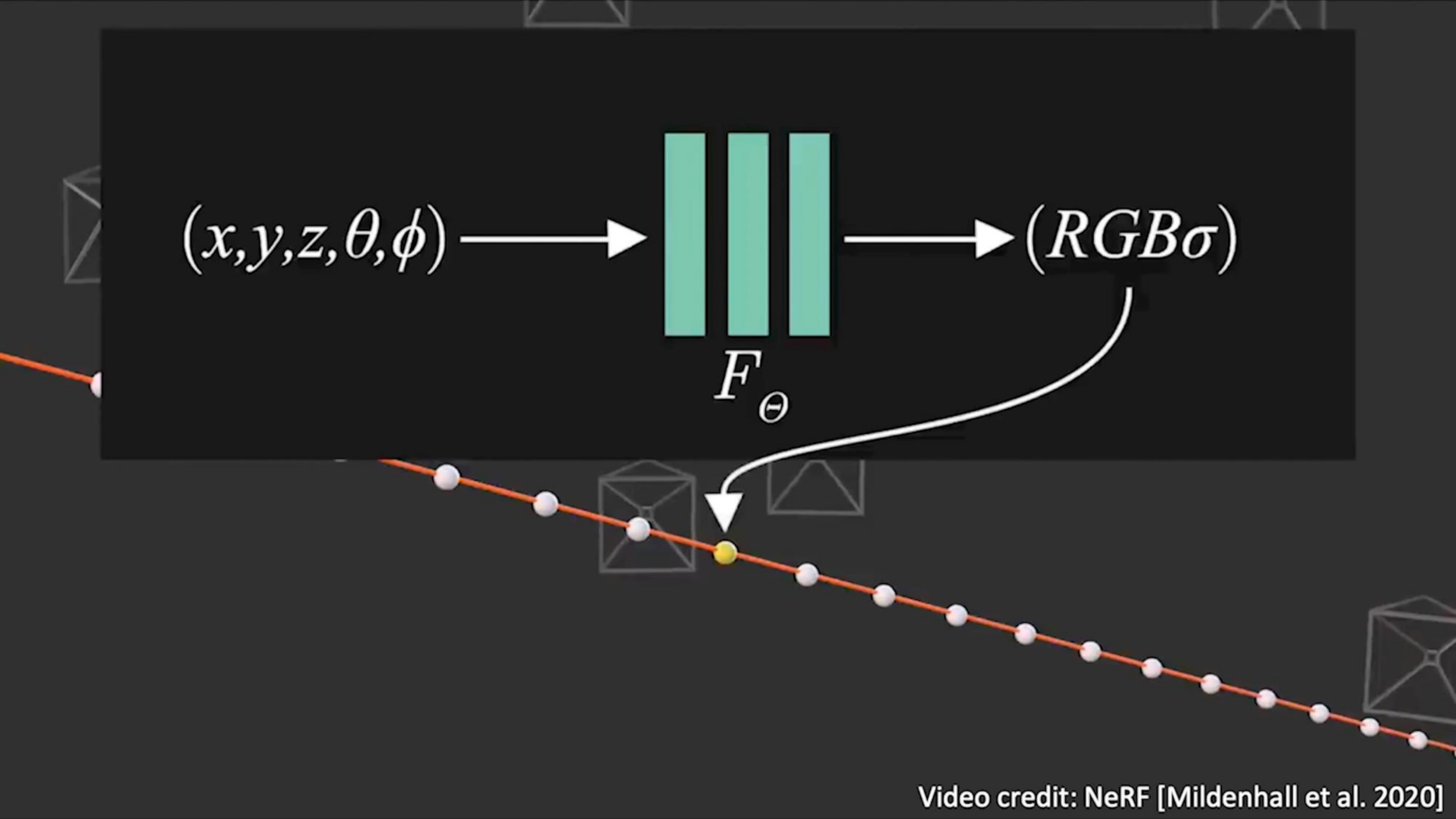
Easy to optimize
Large memory footprint

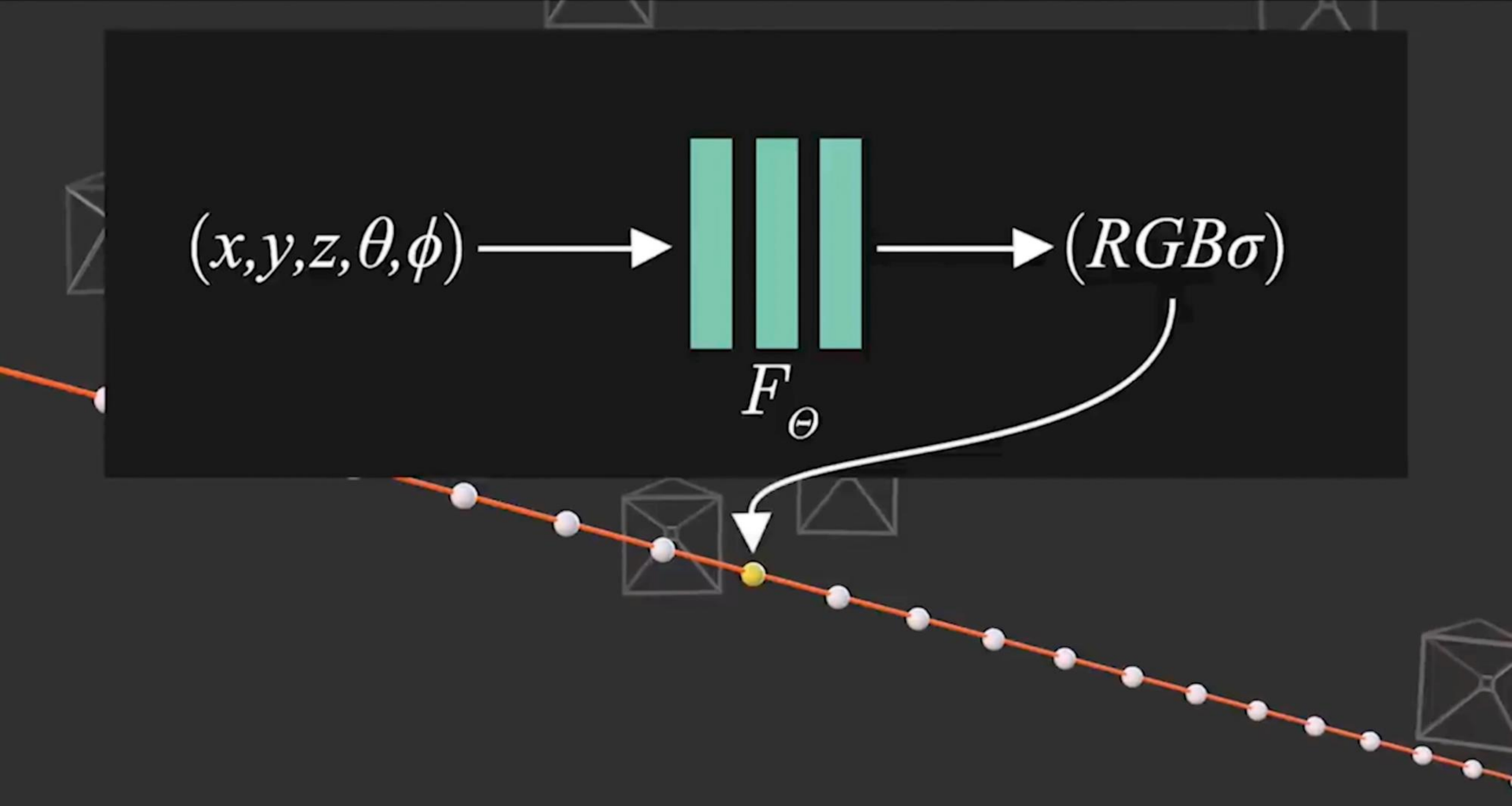


Multi Layer Perceptron

Easy to optimize
Small memory footprint

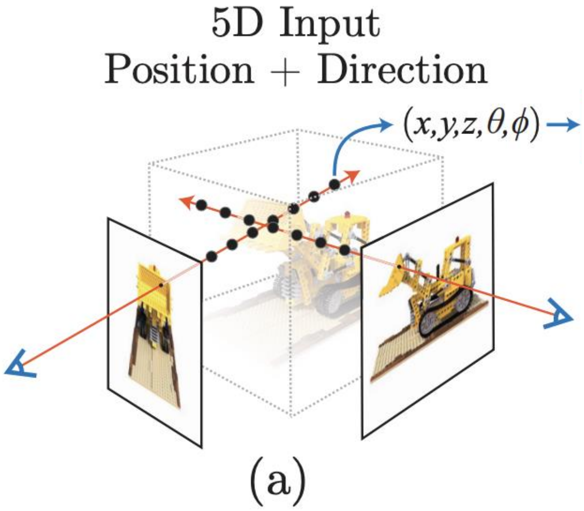
Slide modified from Mathew Tancik's talk


(x,y,z,θ,ϕ)
$(RGB\sigma)$
F_Θ
Video credit: NeRF [Mildenhall et al. 2020]

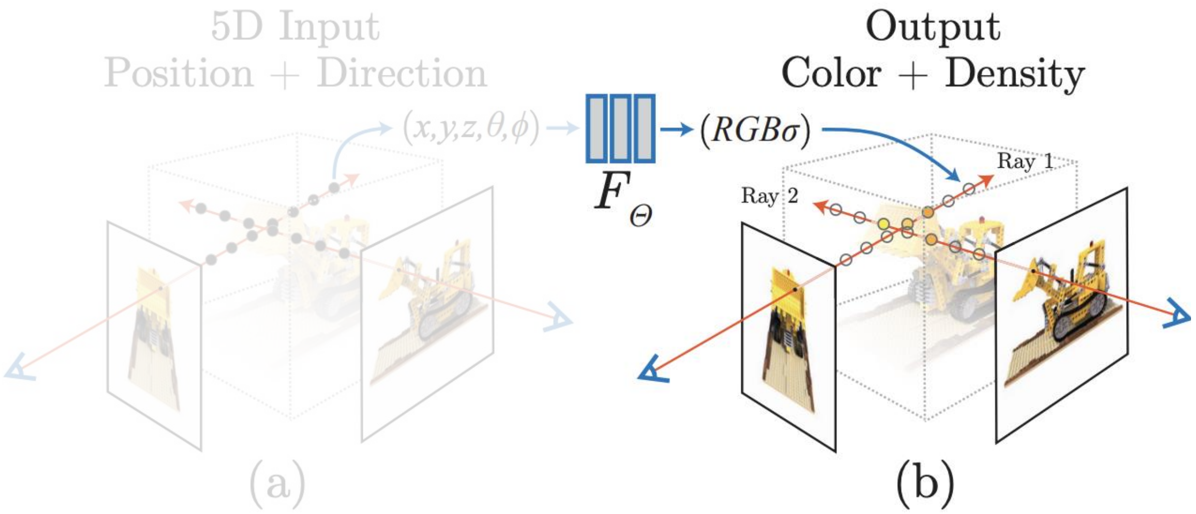


Video credit: NeRF [Mildenhall et al. 2020]

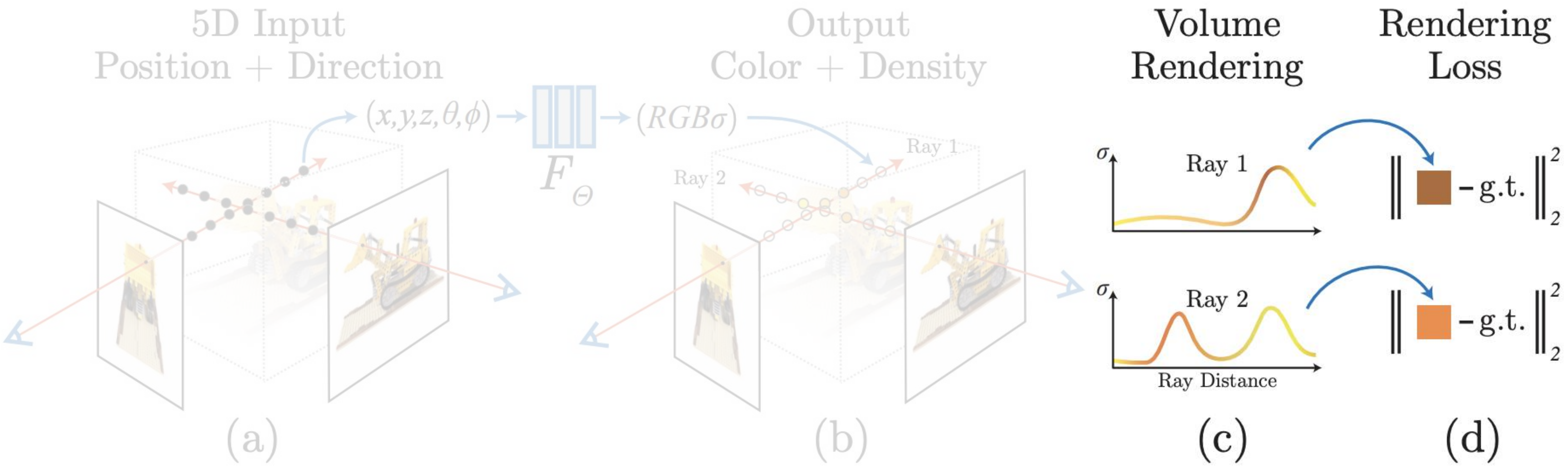
Neural Radiance Fields (NeRFs)



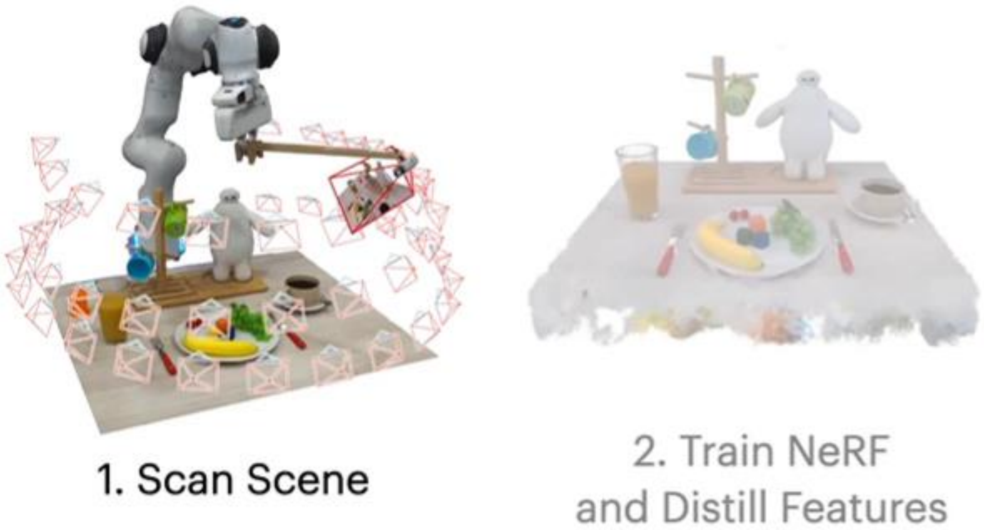
Neural Radiance Fields (NeRFs)



Neural Radiance Fields (NeRFs)



NeRF for Grasping



Distilled Feature Fields, Shen et al.

Summary so far

1) Because voxel grids are memory inefficient, we can use coordinate-based MLPs to store data efficiently

2) NeRF’s volumetric rendering enables photorealistic rendering

3) Downstream applications include robotics, semantic grounding etc.

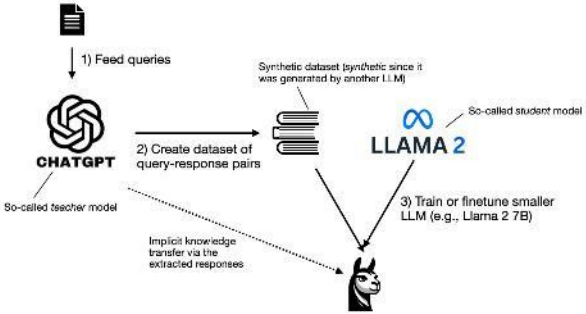
Part 2: Foundation Models

Large Models trained on massive datasets

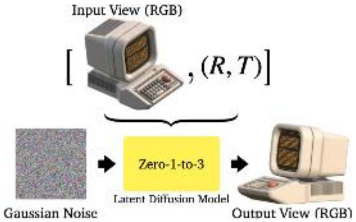
Foundation Model

8x 222B parameters

8B parameters



Distilled Model



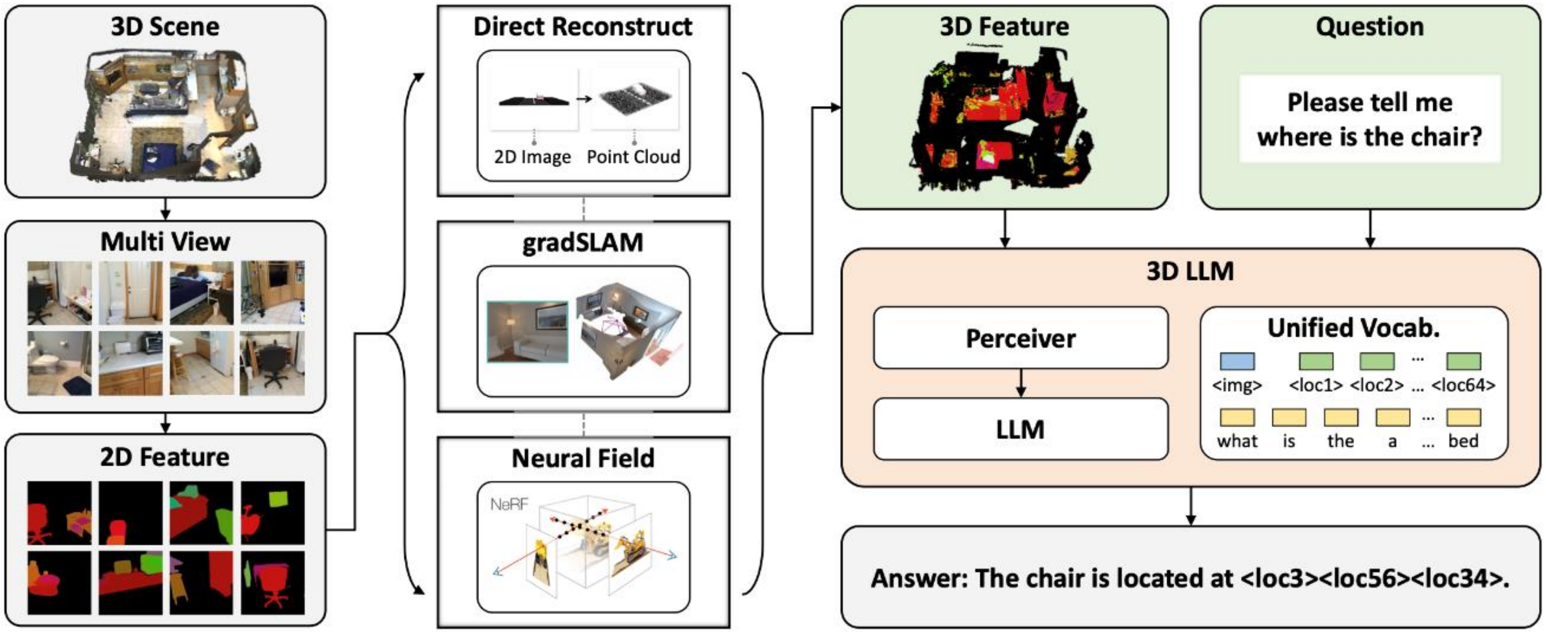
Finetuned Model

Visual Instruction tuning

Better capability

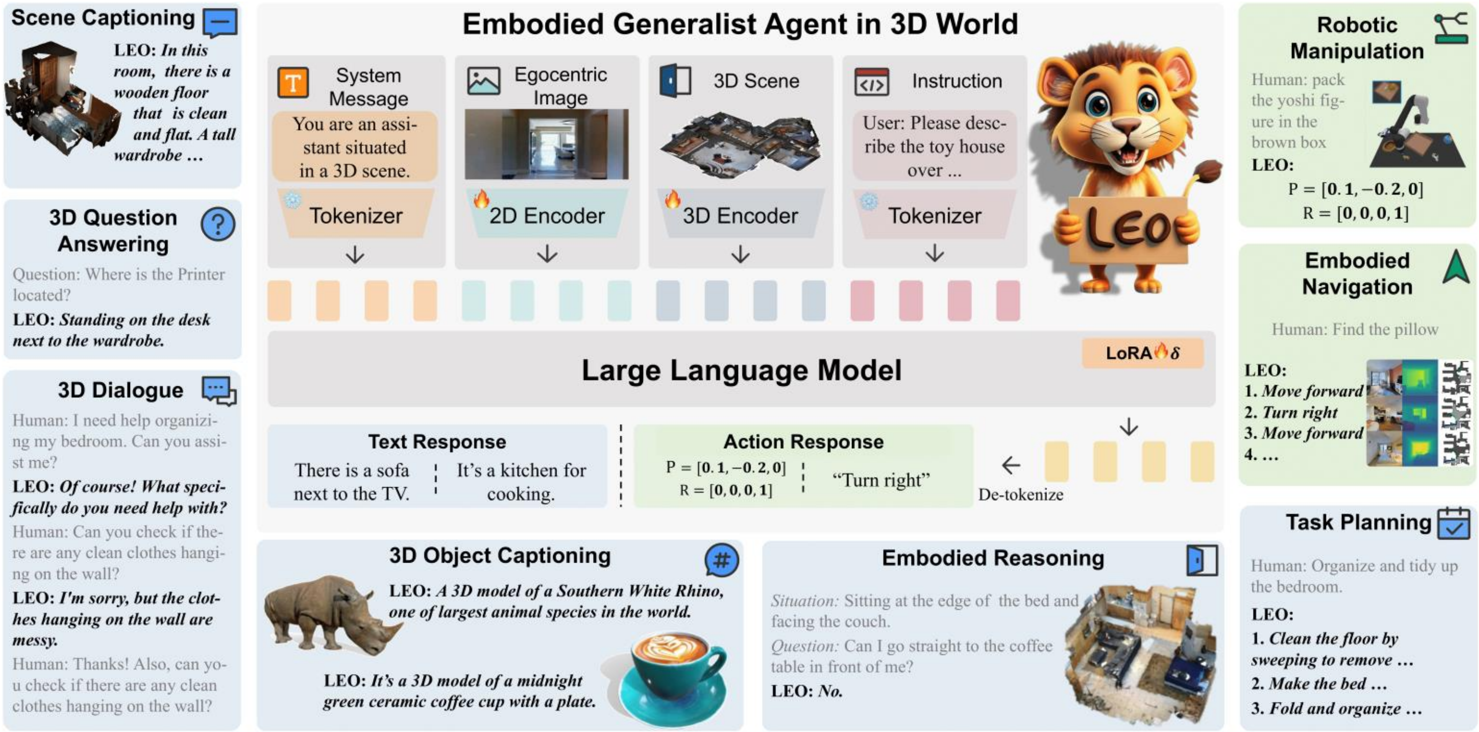
Phi-2 Microsoft, Zero123 Liu et al, Llava Lu et al.

3D LLM



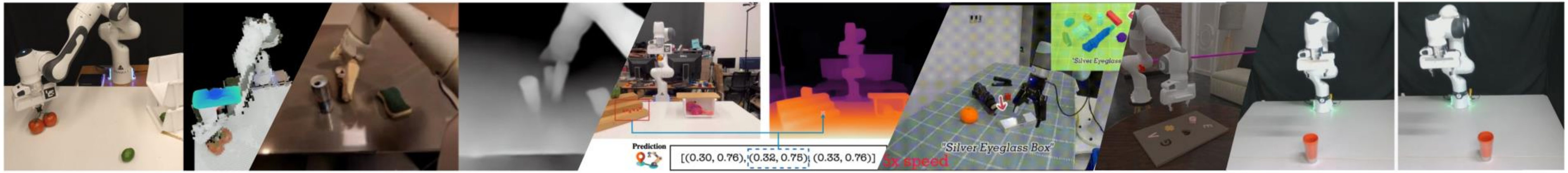
Yining et al.3D LLM, Neurips 2023

Embodied Foundation Model

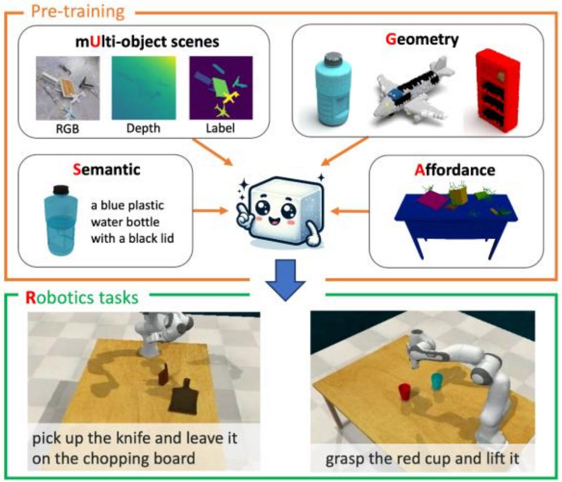


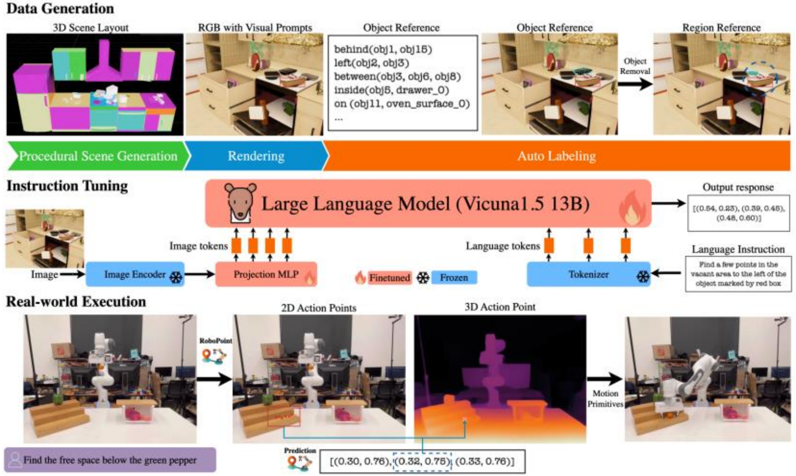
Huang et al. LEO, ICML 2024

What about 3D + Robotics?



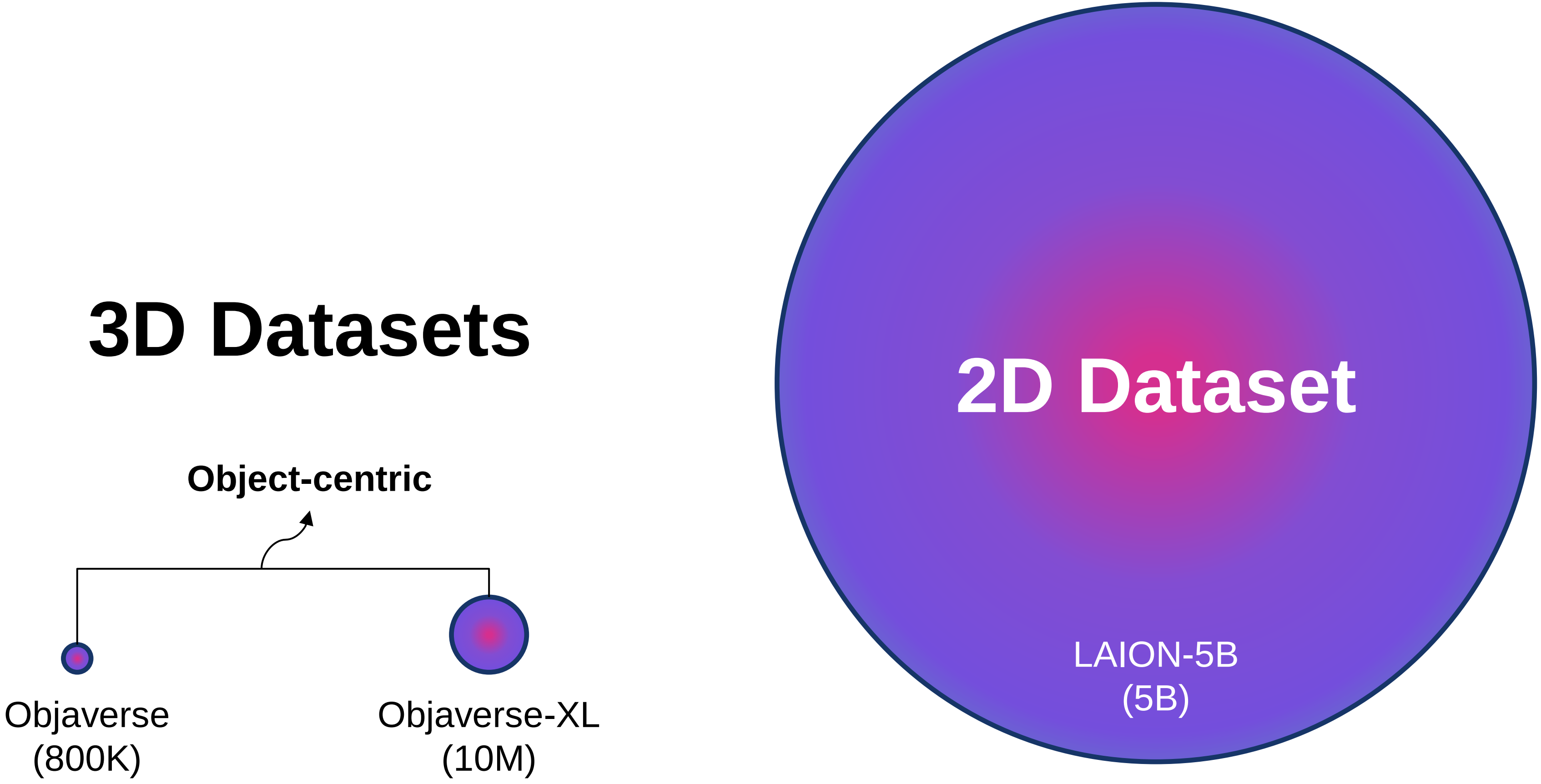
Policy Learning **VLMs and VLA** **Representations** **Simulation/Benchmarks** **Pretraining**





SUGAR CVPR 2024, RoboPoint arXiv 2024

Why Neural Fields Matter for 3D Foundation Models



Summary so far

1) Foundation models are essential due to various reasons i.e. saving resources

2) 3D vision is starting to see some decent foundation models

3) Foundation models can be pulled into smaller more meaningful models through finetuning or model distillation

Part 3: How to build towards 3D Foundation Models

ShAPO: Implicit Representations for Shape Appearance and Pose Optimization



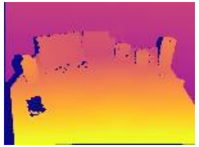


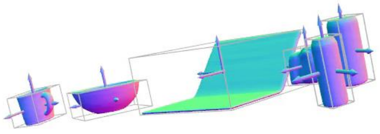

Zubair Irshad   Sergey Zakharov   Rares Ambrus   Thomas Kollar   Zsolt Kira   Adrien Gaidon

European Conference in Computer Vision 2022

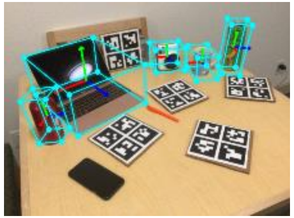
Motivation

Input

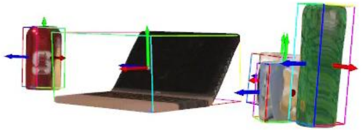




3D Shape

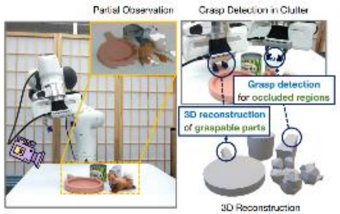


6D pose and size



Appearance

holistic category-level
3D object understanding



Robotics Grasping [GIGA RSS'21]

Category-level Manipulation [NDF'21]

Asset Creation [HHBD ICCV'19]

Object Reconstruction and Pose Estimation (Current Paradigm)

Key highlights (Prior Methods):

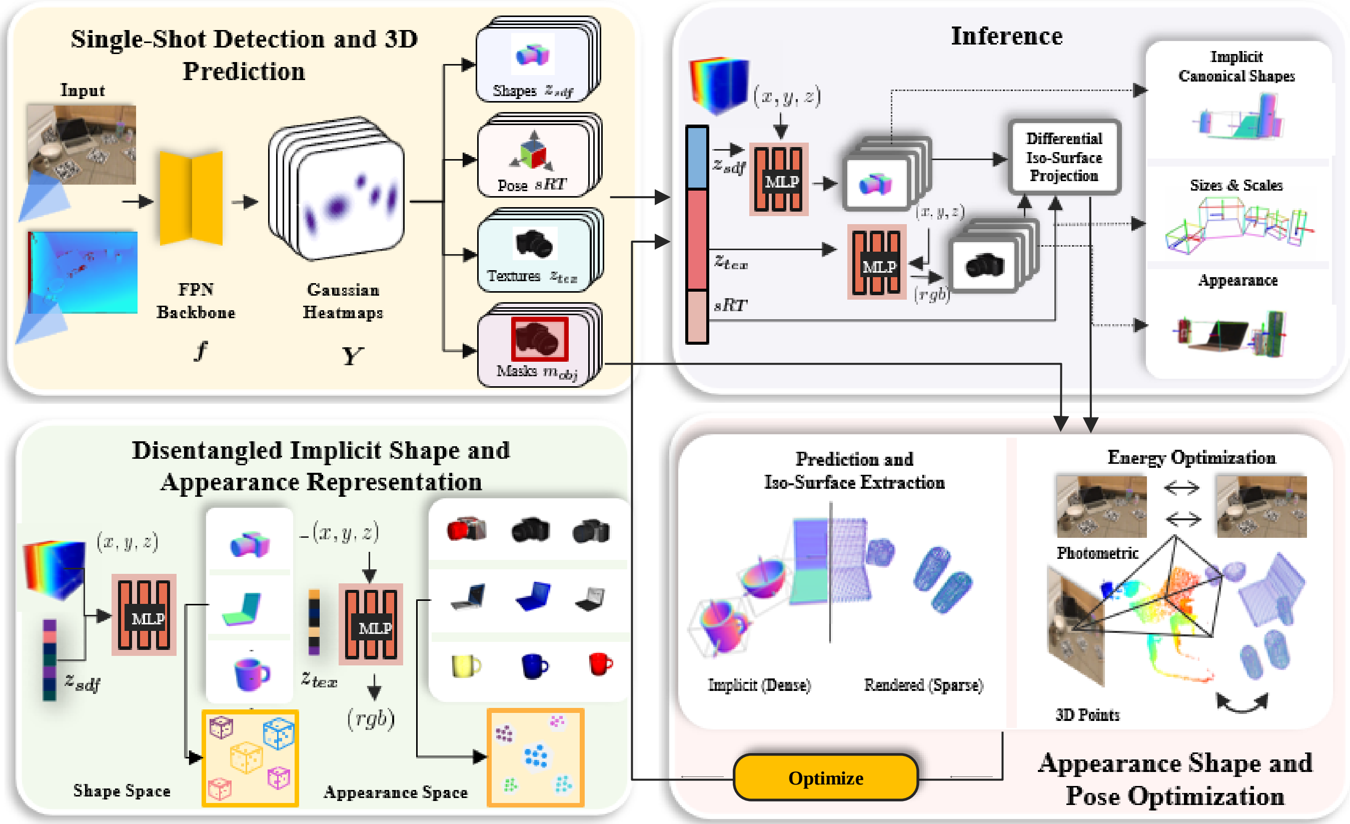
- Anchor-Based
- Disjoint shape reconstruction and object-centric scene context
- Slow reconstruction
- Category-specific reconstruction and 6D pose and size estimation

Key highlights (Our proposed):

+ Anchor-free 3D Contribution 1
+ Joint shape reconstruction and object-centric scene context
+ Fast (Real-time) reconstruction Contribution 2
+ Category agnostic reconstruction and 6D pose and size estimation Contribution 3

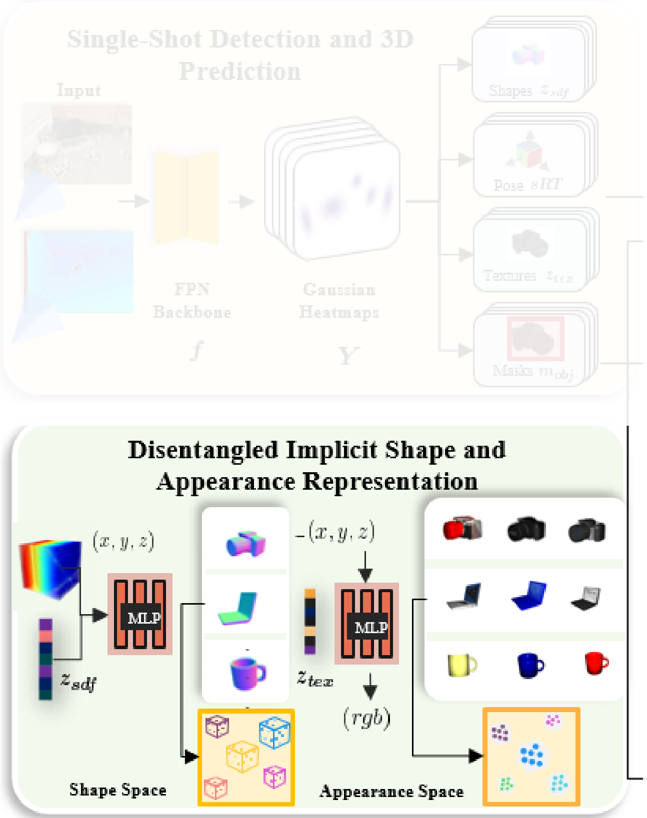
0.05FPS → 40FPS

Architecture



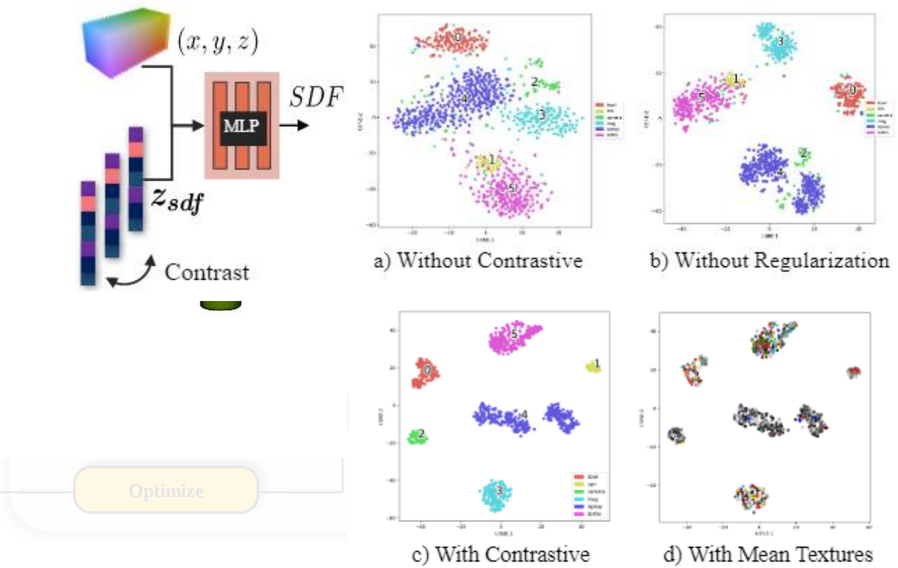
[Ref] M.Z.Irshad, et al, " ShAPO : Implicit Representations for Multi Object Shape Appearance and Pose Optimization, **ECCV2022**

Shape and Appearance Prior Database



Key highlights:

- Represents geometry as continuous SDF
  - $G(\boldsymbol{x}, \boldsymbol{z}_{sdf}) = s : z_{sdf} \in \mathbb{R}^{64}, s \in \mathbb{R}$
- Represents appearance as Texture Field
  - $t_\theta : \mathbb{R}^3 \rightarrow \mathbb{R}^3$



[Ref] M.Z.Irshad, et al, " ShAPO : Implicit Representations for Multi Object Shape Appearance and Pose Optimization, **ECCV2022**

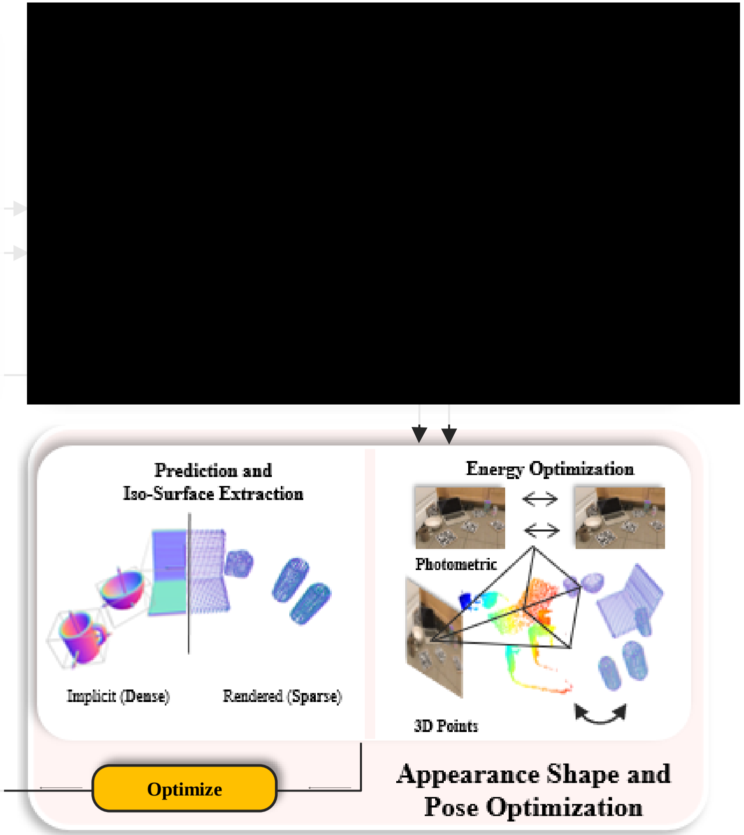
Efficiently optimizing Shape and Texture

Differentiable iso-surface projection:

- Trivial Solution: Threshold the points based on SDF value, Non-Differentiable
- Alternate solution: Utilize gradients and normal values (Ours)

$$n_i = \frac{\partial G(x_i; \mathbf{z}_{sdf})}{\partial x_i}$$

$$p_i = x_i - \frac{\partial G(x_i; \mathbf{z}_{sdf})}{\partial x_i} G(x_i; \mathbf{z}_{sdf})$$

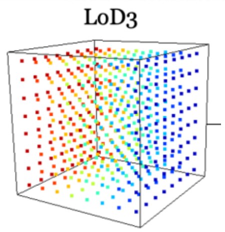


Animation: SDF Label CVPR, 2020

Efficiently optimizing Shape and Texture

Octree-based point sampling:

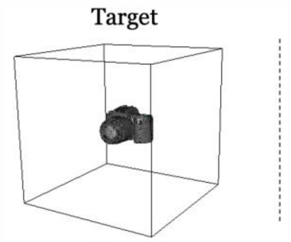
- Brute Force Solution: Extremely inefficient
- Sampling 216000 ~= 1600 surface points (0.7%)
- Solution: Coarse-to-fine sampling
- LoD3 to LoD7



Efficiently optimizing Shape and Texture

Octree-based point sampling:

- Brute Force Solution: Extremely inefficient
- 603 points = 216000 ~= 1600 surface points (0.7%)
- Solution: Coarse-to-fine sampling
- LoD3 to LoD7



Quantitative Results

Takeaway: *Establish a new* ***SOTA*** *for 6D Pose and Size Estimation, while* ***adding textures*** *to the representation!*

Metrics: ***Detection*** *(Intersection over Union, IOU@2525, IOU@50)* ***Pose Estimation*** *(Rotation, translation accuracy)*

Table 2: **Quantitative comparison of 6D pose estimation and 3D object detection on NOCS** [41]: Comparison with strong baselines. Best results are highlighted in **bold**. $*$ denotes the method does not report IOU metrics since size and scale is not evaluated. We report metrics using nocs-level class predictions for a fair comparison with all baselines.

Method	CAMERA25						REAL275					
	IOU25	IOU50	5°5 cm	5°10 cm	10°5 cm	10°10 cm	IOU25	IOU50	5°5 cm	5°10 cm	10°5 cm	10°10 cm
1 NOCS [41]	91.1	83.9	40.9	38.6	64.6	65.1	84.8	78.0	10.0	9.8	25.2	25.8
2 Synthesis* [3]	-	-	-	-	-	-	-	-	0.9	1.4	2.4	5.5
3 Metric Scale [23]	93.8	90.7	20.2	28.2	55.4	58.9	81.6	68.1	5.3	5.5	24.7	26.5
4 ShapePrior [37]	81.6	72.4	59.0	59.6	81.0	81.3	81.2	77.3	21.4	21.4	54.1	54.1
5 CASS [2]	-	-	-	-	-	-	84.2	77.7	23.5	23.8	58.0	58.3
6 CenterSnap [15]	93.2	92.3	63.0	69.5	79.5	87.9	83.5	80.2	27.2	29.2	58.8	64.4
7 CenterSnap-R [15]	93.2	92.5	66.2	71.7	81.3	87.9	83.5	**80.2**	29.1	31.6	64.3	70.9
8 **ShAPO (Ours)**	**94.5**	**93.5**	**66.6**	**75.9**	**81.9**	**89.2**	**85.3**	79.0	**48.8**	**57.0**	**66.8**	**78.0**

Table 3: **Quantitative comparison of 3D shape reconstruction on NOCS** [41]: Evaluated with **CD** metric (10^{-2}). Lower is better.

Method	CAMERA25							REAL275						
	Bottle	Bowl	Camera	Can	Laptop	Mug	Mean	Bottle	Bowl	Camera	Can	Laptop	Mug	Mean
1 Reconstruction [37]	0.18	0.16	0.40	0.097	0.20	0.14	0.20	0.34	0.12	0.89	0.15	0.29	0.10	0.32
2 ShapePrior [37]	0.34	0.22	0.90	0.22	0.33	0.21	0.37	0.50	0.12	0.99	0.24	0.71	0.097	0.44
3 CenterSnap	0.11	0.10	0.29	0.13	0.07	0.12	0.14	0.13	0.10	0.43	0.09	0.07	0.06	0.15
3 **ShAPO (Ours)**	0.14	0.08	0.2	0.14	0.07	0.11	0.16	0.1	0.08	0.4	0.07	0.08	0.06	0.13

Ablation Analysis

Takeaways:

1. *LoD7 has the higher accuracy while LoD6 gives the best* ***speed/accuracy trade-off***
2. *PSNR* ***improves after optimization*** *and finetuning confirming iterative optimization helps fine-tuning*

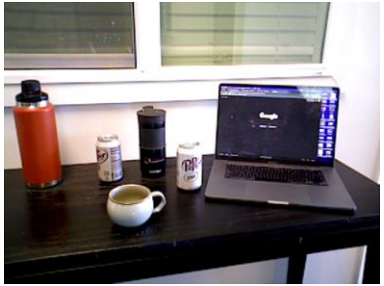
Table 4: **Generalizable Implicit Representation Ablation**: We evaluate the efficiency (point sampling/time(s)/memory(MB)) and generalization (shape(CD) and texture(PSNR) reconstruction) capabilities of our implicit object representation as well as its sampling efficiency for different levels of detail (LoDs) and compare it to the ordinary grid sampling. All ablations were executed on NVIDIA RTX A6000 GPU.

Grid type	Resolution	Point Sampling		Efficiency (per object)		Reconstruction	
		Input	Output	Time (s)	Memory (MB)	Shape (CD)	Texture (PSNR)
Ordinary	40	64000	412	10.96	3994	0.30	10.08
	50	125000	835	18.78	5570	0.19	12.83
	60	216000	1400	30.51	7850	0.33	**19.52**
OctGrid	LoD5	1521	704	**5.53**	**2376**	0.19	9.27
	LoD6	5192	3228	6.88	2880	**0.18**	**13.63**
	LoD7	20246	13023	12.29	5848	0.24	16.14

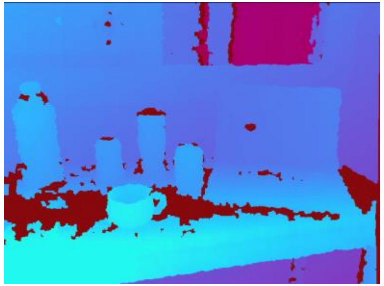
Table 1: **Texture quality ablation.** We compare texture quality using the PSNR metric between three modalities: network prediction, optimization, and fine-tuning of the t_θ network.

	Inference	Optimization	Fine-tuning
PSNR	11.41	20.64	24.32

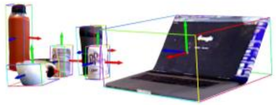
Qualitative Results (In-the-wild on HSR Robot)



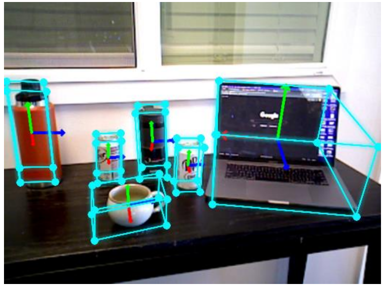
RGB



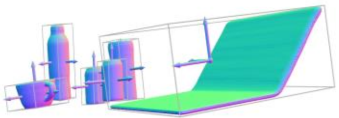
Depth



Appearance Reconstruction



6D pose and size



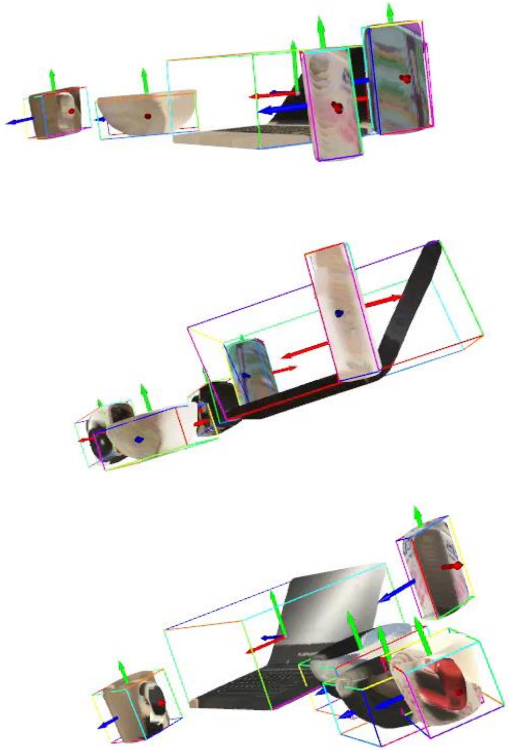
3D Shape

Testing Results on **Xtion Pro Live** Camera on HSR Robot

Qualitative Results

NOCS REAL275

Input



Summary so far

1) Categorical 3D models can model a large number of categories of objects

2) Combining them with detection makes them efficient retrievers

3) Scaling to thousands of categories is still a slight challenge

NeRF-MAE

Masked AutoEncoders for Self-Supervised 3D Representation Learning for Neural Radiance Fields

European Conference on Computer Vision, ECCV 2024
also appeared at CVPR Neural Rendering Intelligence Workshop, 2024


Zubair Irshad


Sergey Zakharov


Vitor Guizilini

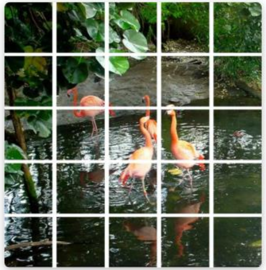

Adrien Gaidon


Zsolt Kira


Rares Ambrus

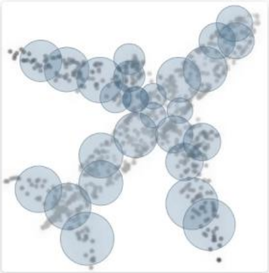
What is representation Learning?

MAE



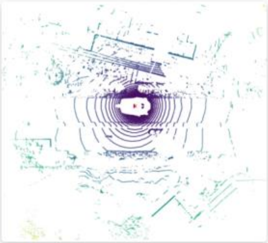
Data: 2D Images
Representation: 2D

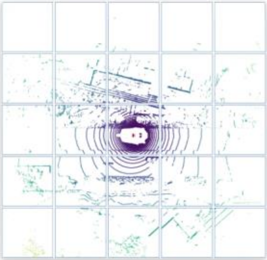
Point-MAE



Data: 3D PointClouds
Representation: 3D

Voxel-MAE





Data: Lidar PointClouds
Representation: 3D

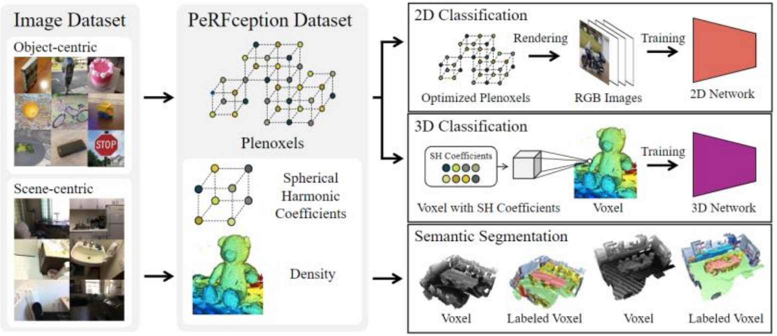
MAE, CVPR2022 | Point-MAE, ECCV2022 | Voxel-MAE, WACV2023

Neural Fields beyond showcasing high rendering quality

Language-Embedded Radiance Fields
(LeRF, Kerr et al)

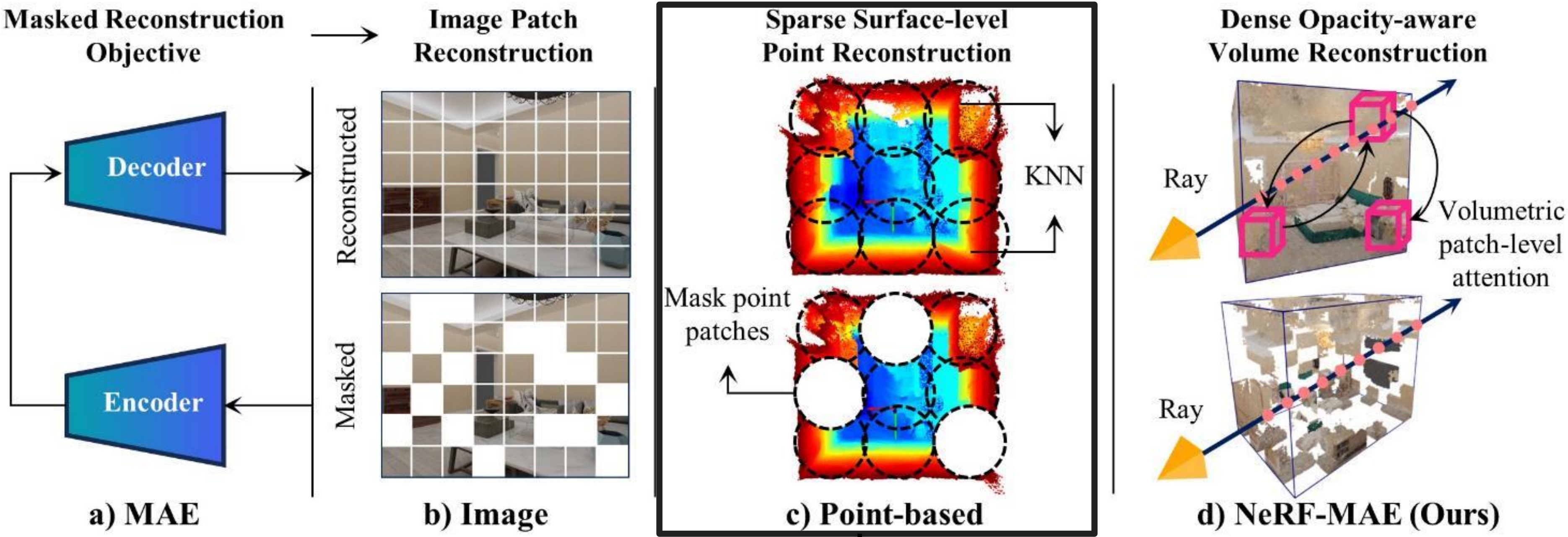
Inferring Accurate Geometry
(NeRFMeshing, Rakotosaona et al)

Open-world Manipulation
(F3RM, Shen et al)



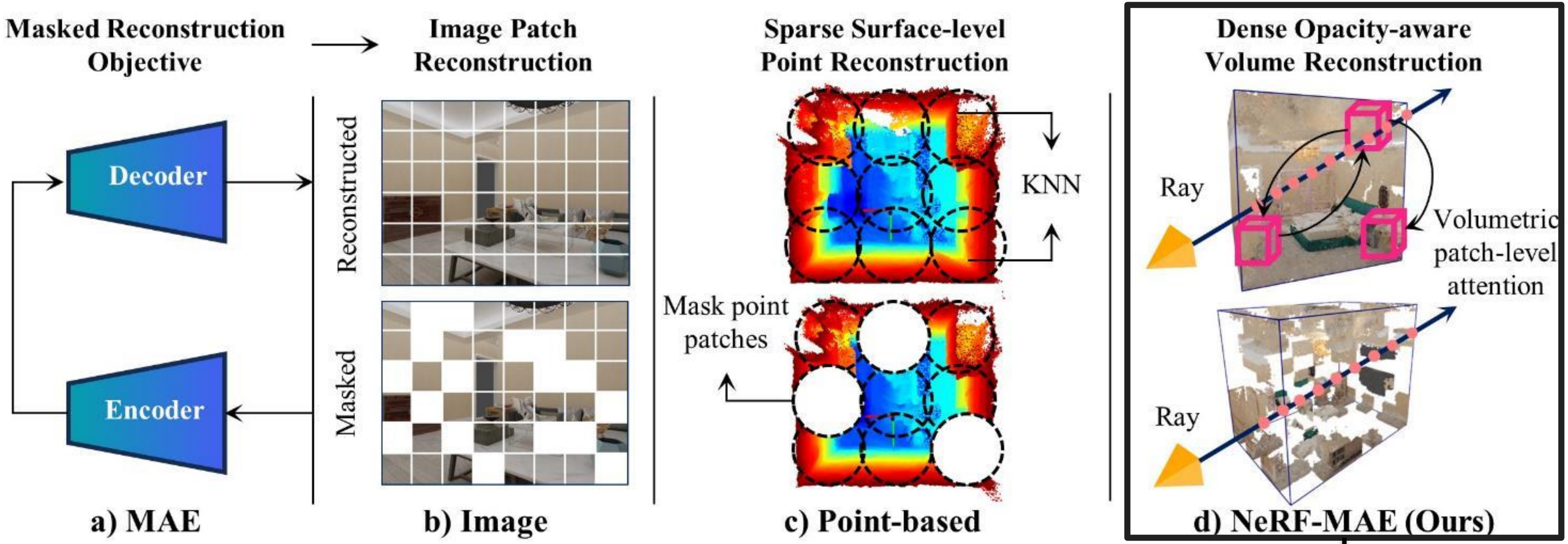
Efficient Data Storage
(PerFception, Jeong et al)

Existing 3D MAE architectures vs NeRF-MAE



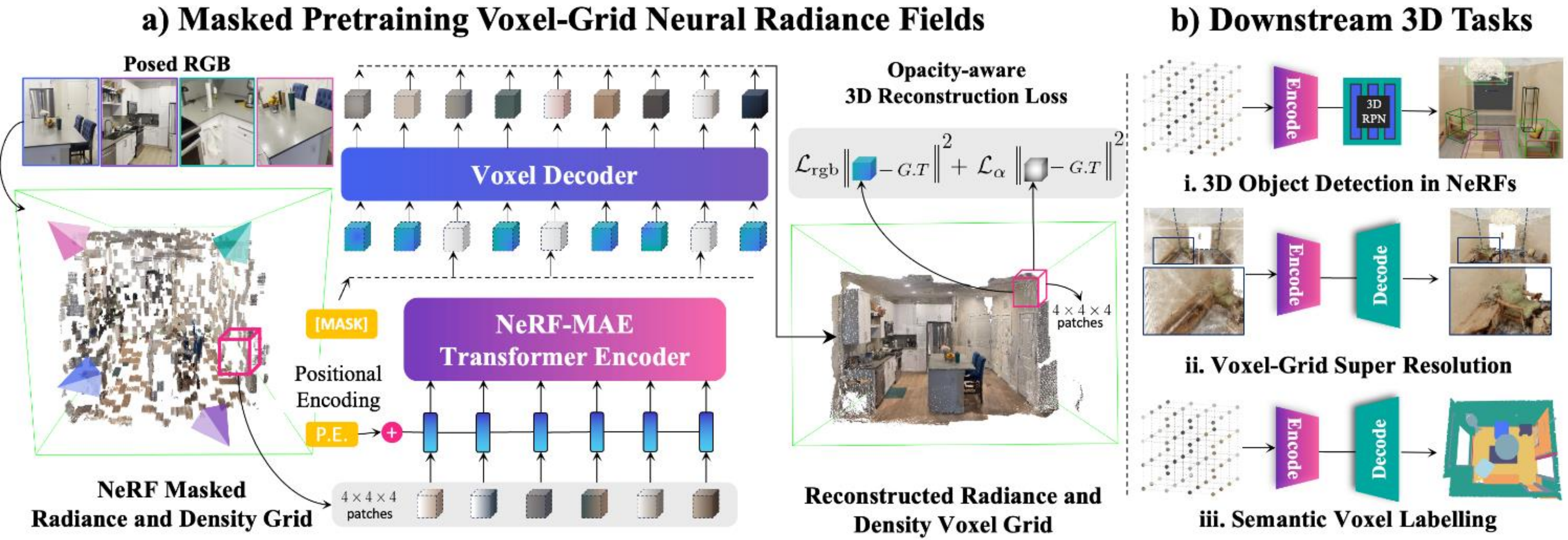
✗ Model Surface Level Information | ✗ Irregular data Structures | ✗ Uneven information density

Existing 3D MAE architectures vs NeRF-MAE

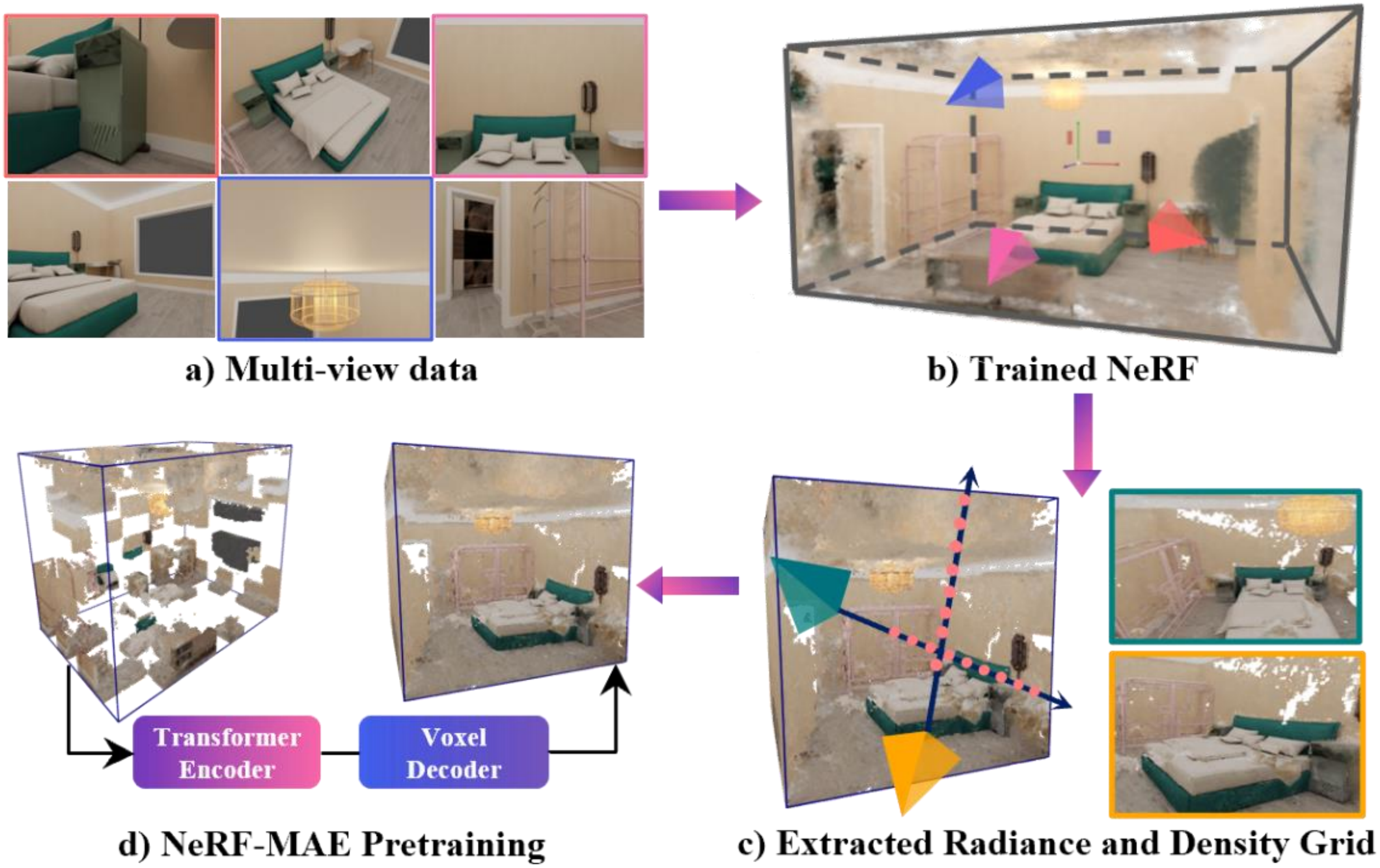


✓ High Information Density | ✓ Regular unbiased Sampling | ✓ Spatial data redundancy

Architecture

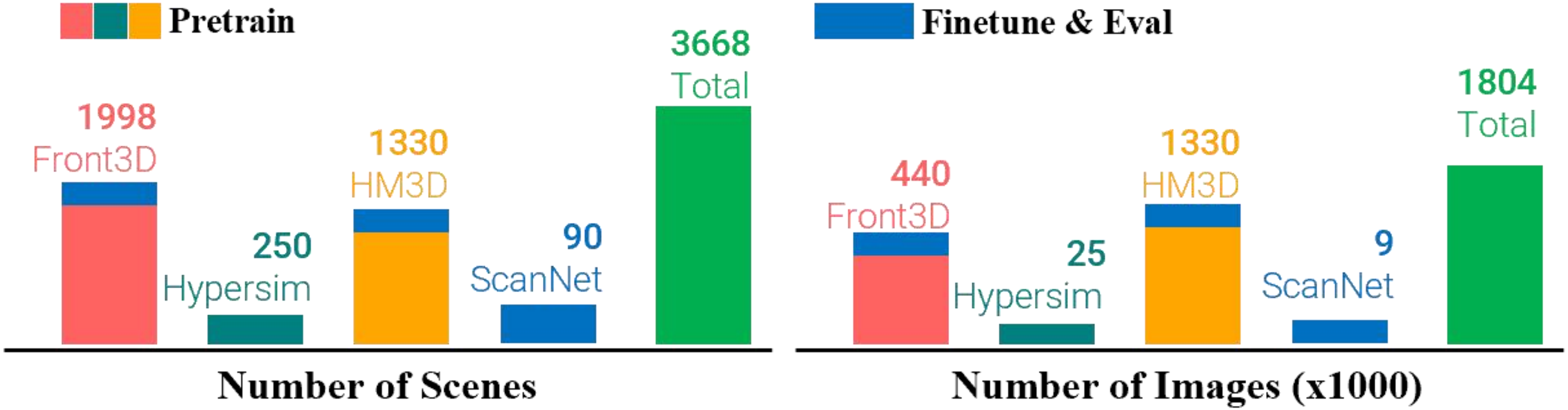


Data preprocessing flow for large-scale 3D pretraining

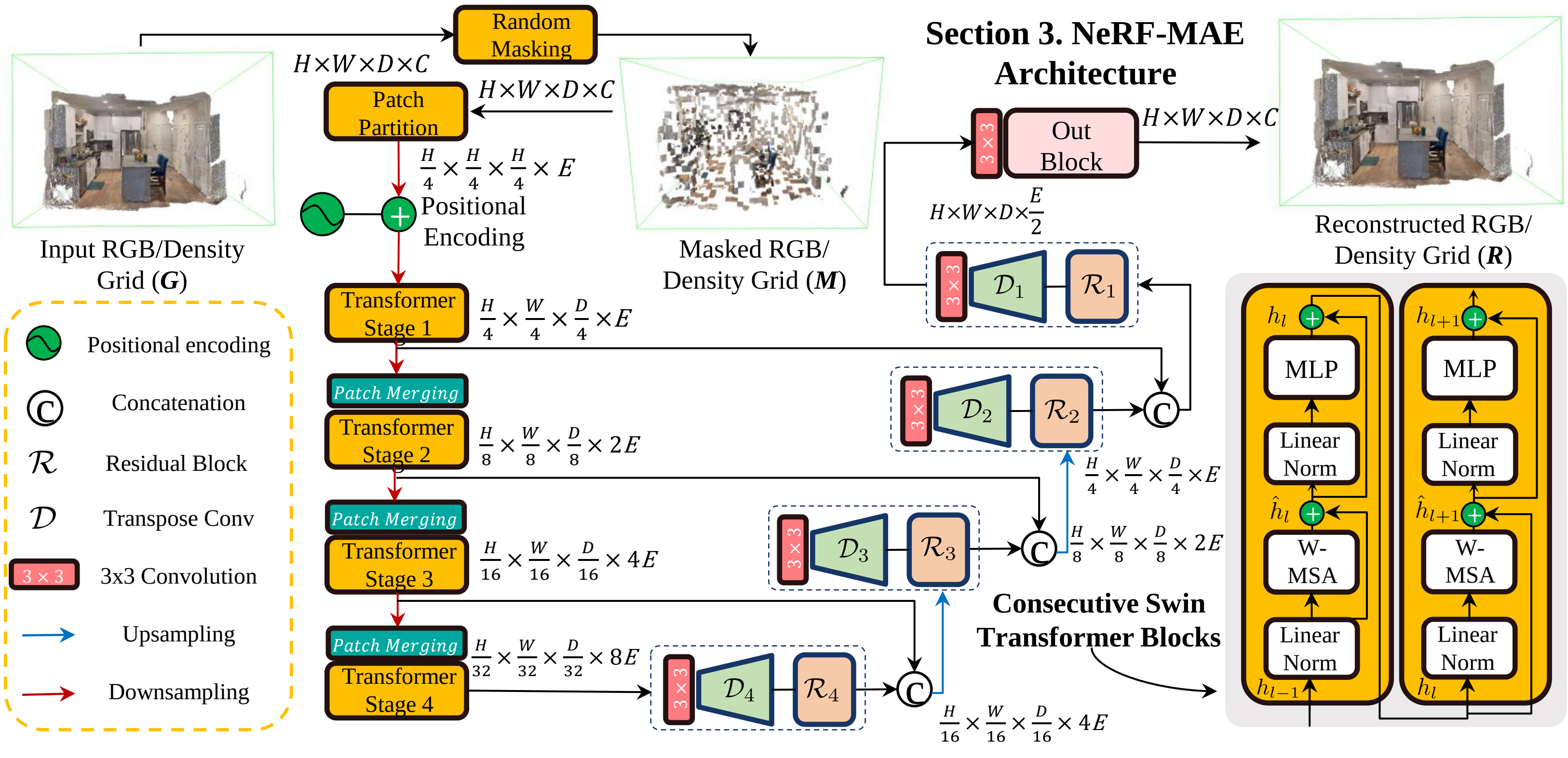


a) Multi-view Dataset Setup

b) NeRF-MAE Data Mix & Statistics

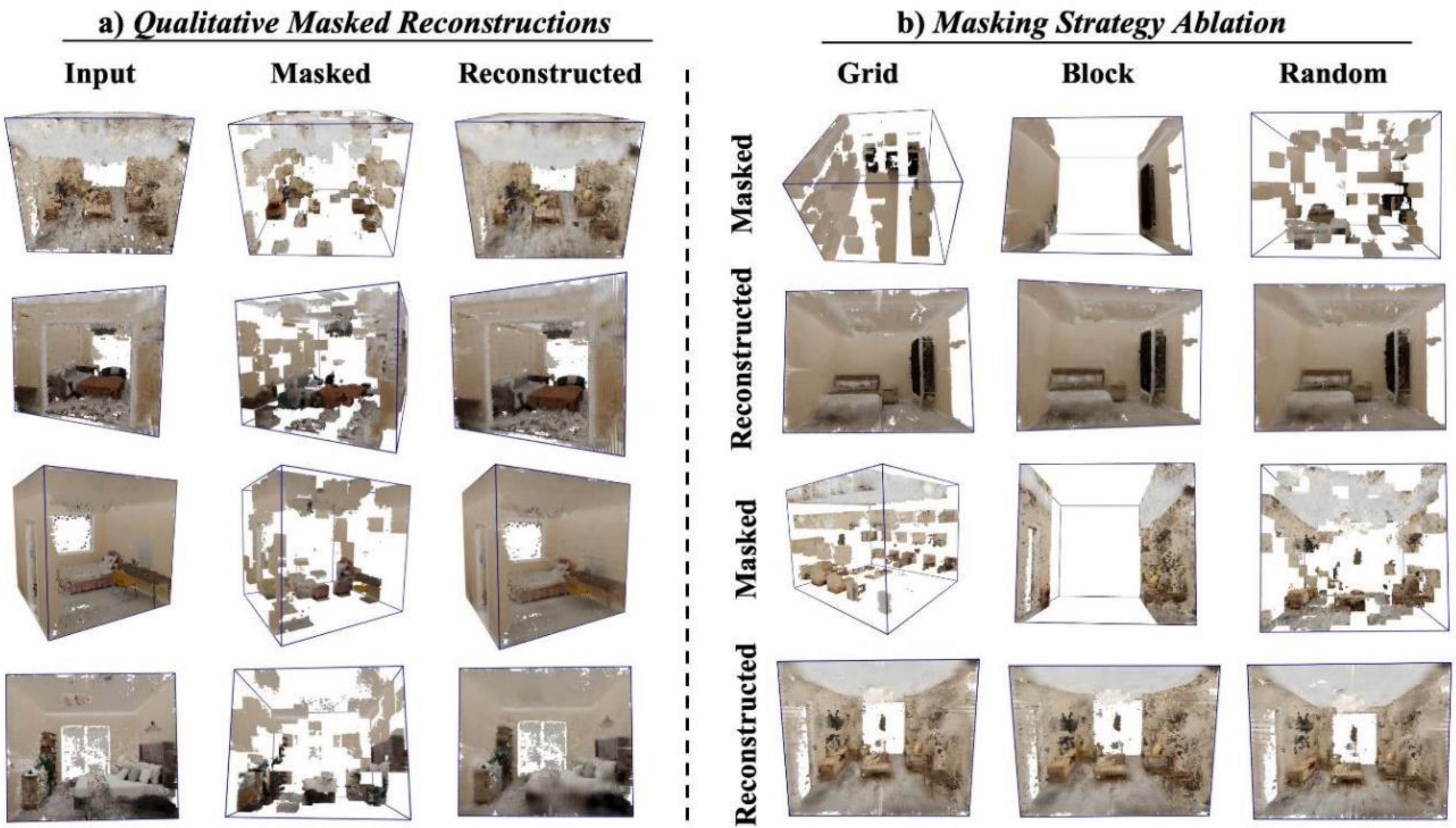


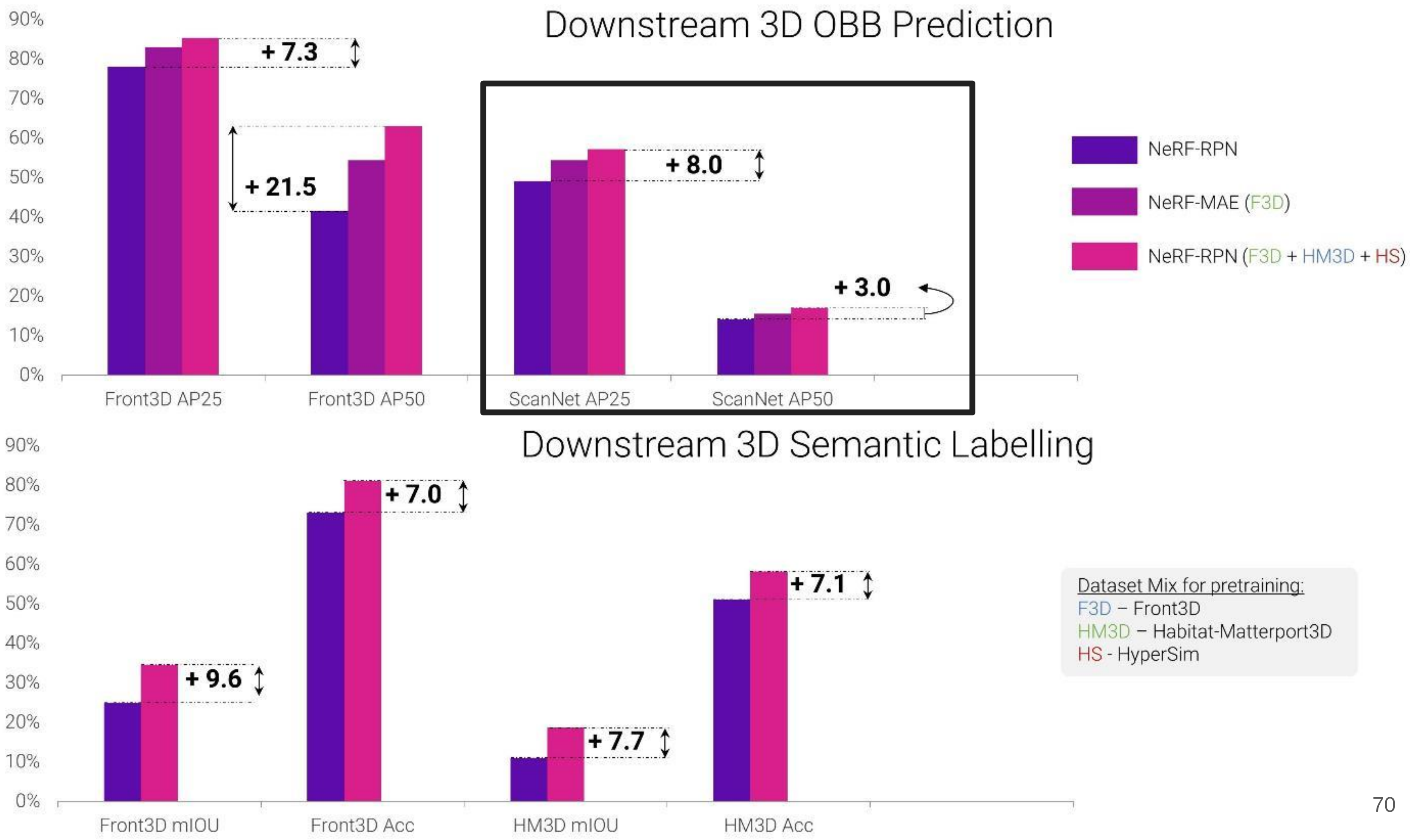
Key Idea: Pretrain a Single Transformer model using masked reconstruction objective



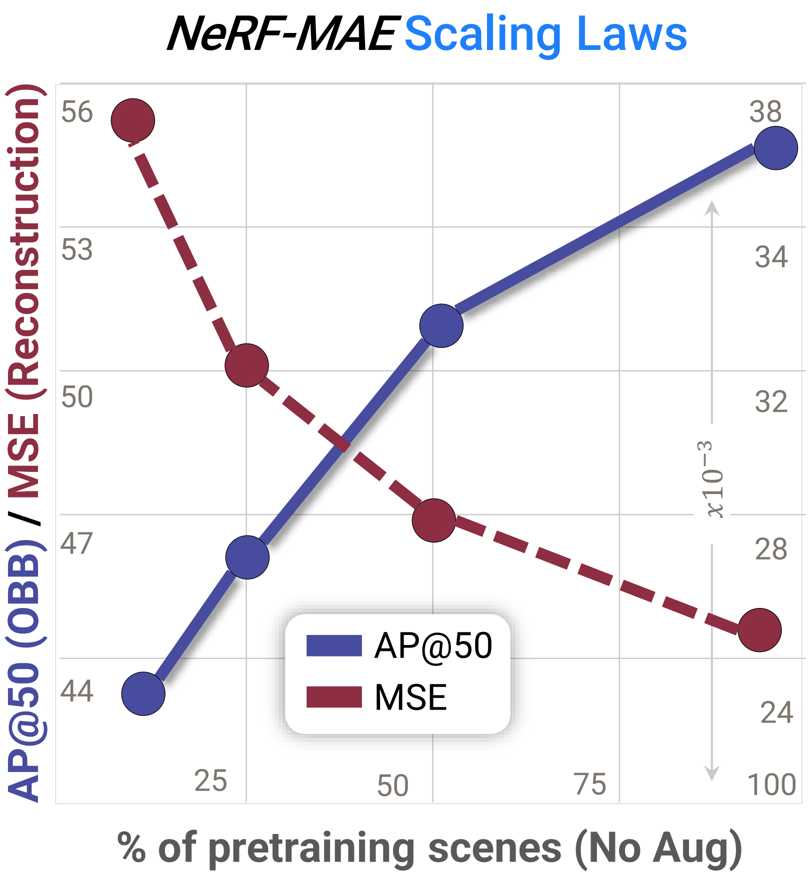
Key Takeaway: Large Model + Large-Scale data = Good Representations

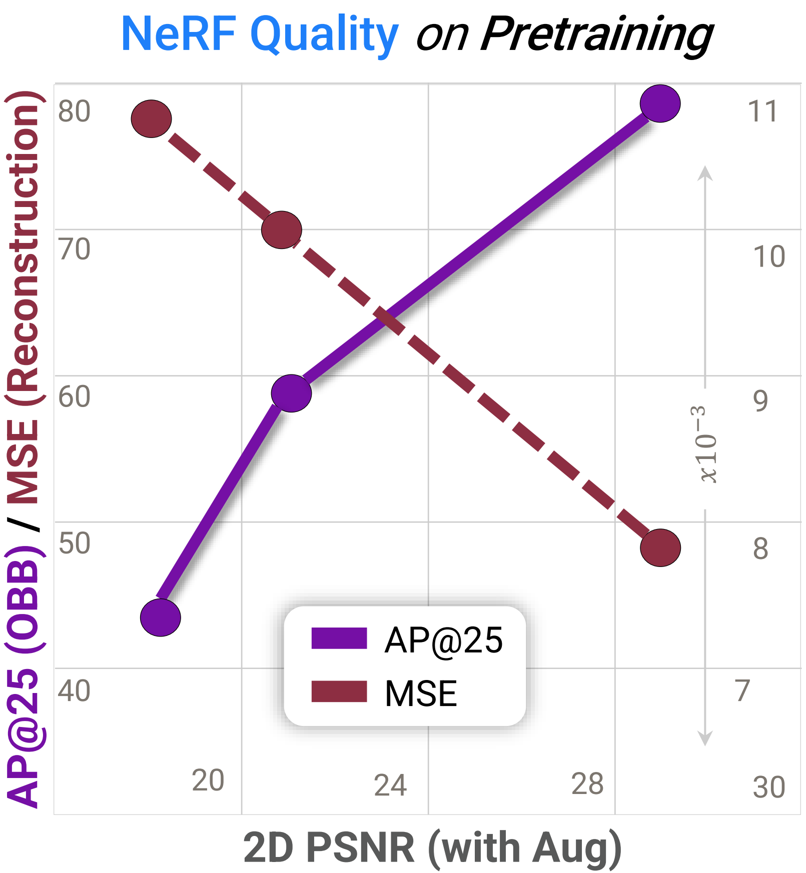
Qualitative Masked Reconstruction Results



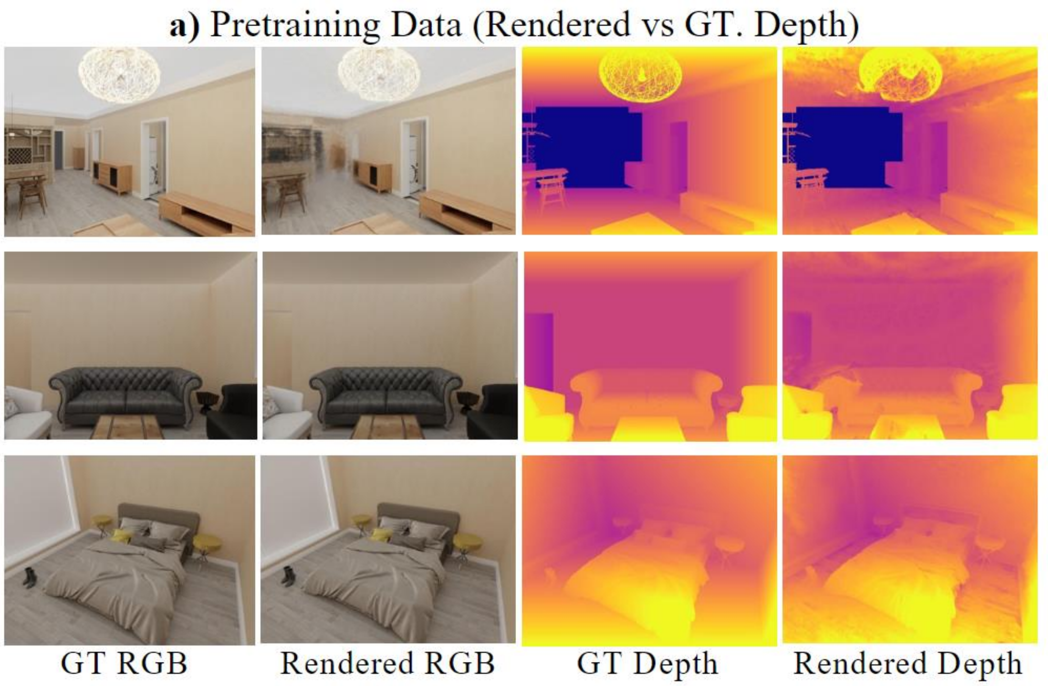

Downstream 3D OBB Prediction
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
+ 7.3
+ 21.5
+ 8.0
+ 3.0
Front3D AP25
Front3D AP50
ScanNet AP25
ScanNet AP50
NeRF-RPN
NeRF-MAE (F3D)
NeRF-RPN (F3D + HM3D + HS)
Downstream 3D Semantic Labelling
+ 9.6
+ 7.0
+ 7.7
+ 7.1
Front3D mIOU
Front3D Acc
HM3D mIOU
HM3D Acc
Dataset Mix for pretraining:
F3D – Front3D
HM3D – Habitat-Matterport3D
HS - HyperSim

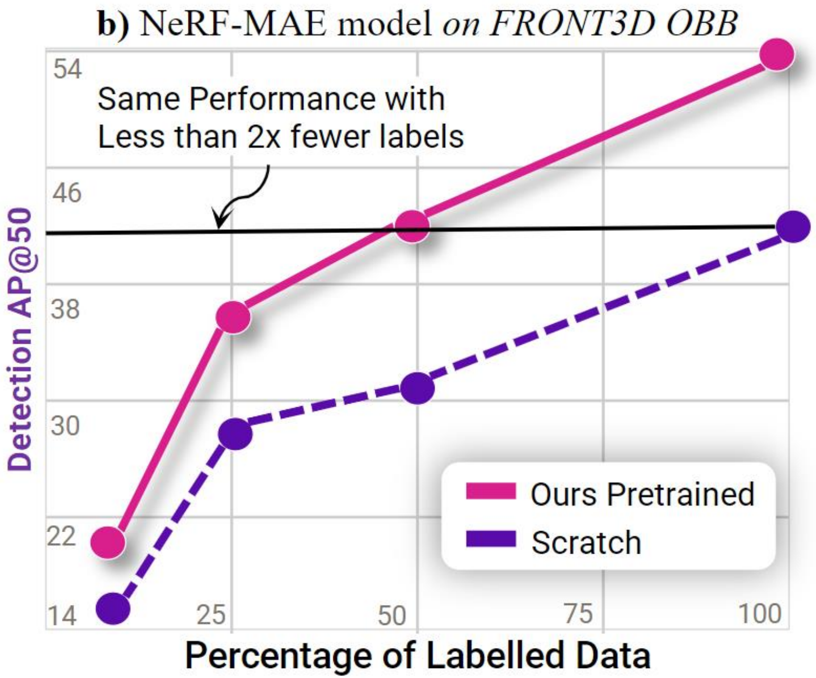
Quantitative Results



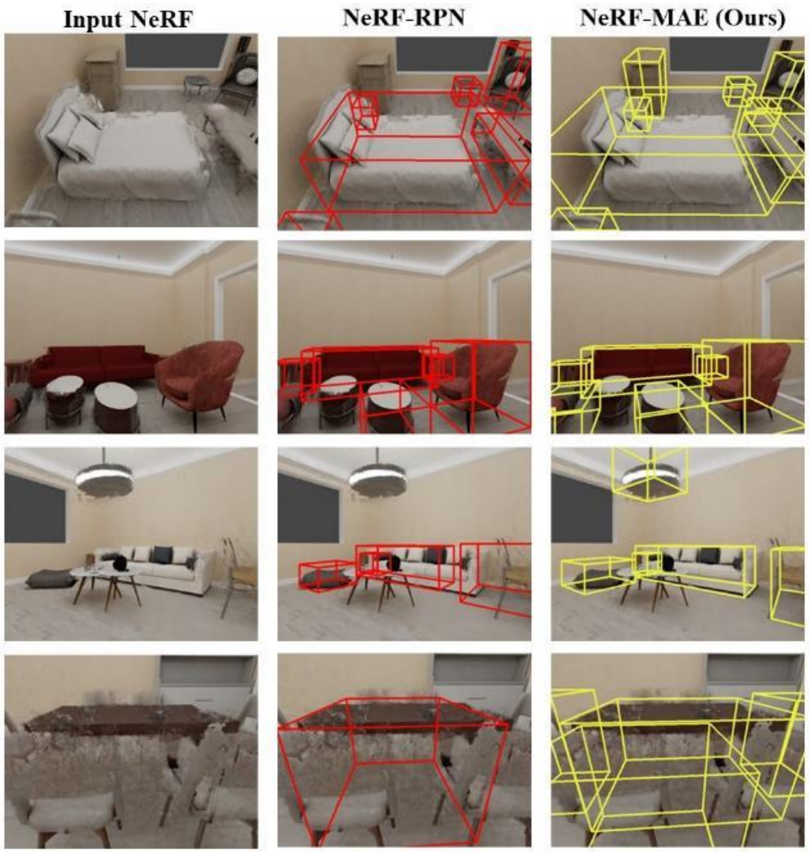


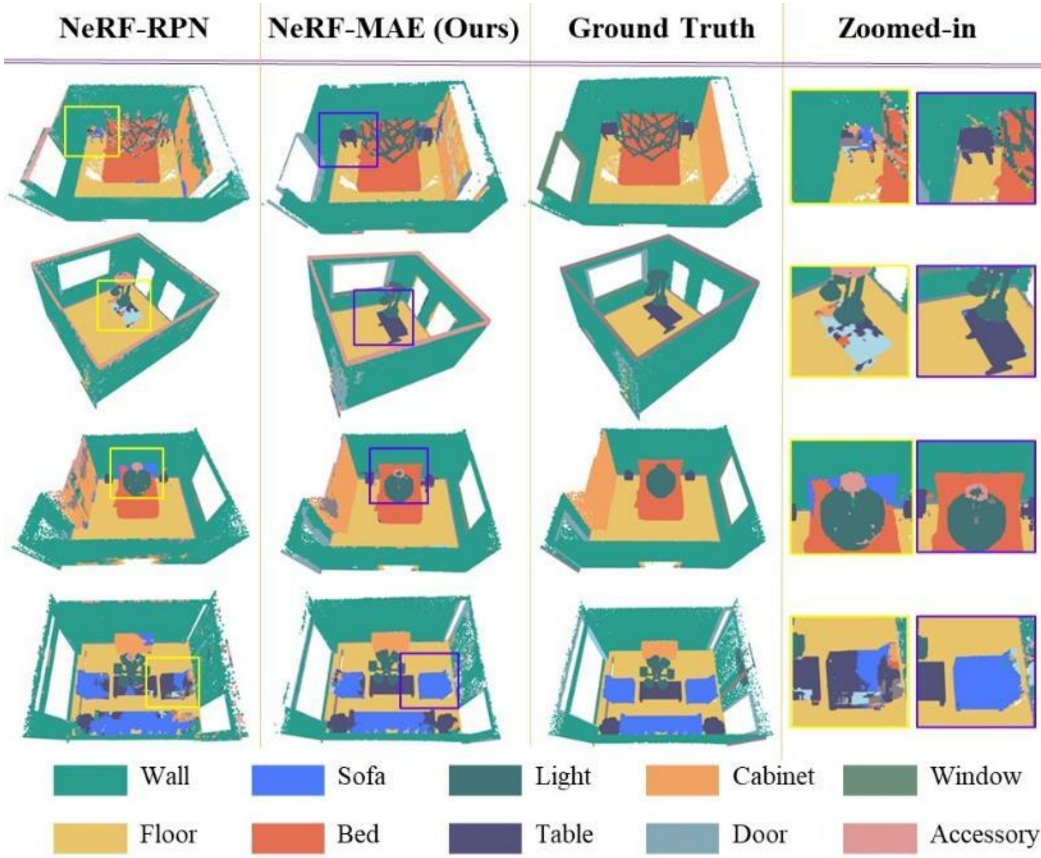
Results Analysis





Qualitative Results





Summary so far

Good:

1) Early signs of life of 3D foundation models only utilizing posed 2D data

2) Scaling helps here too

Bad:

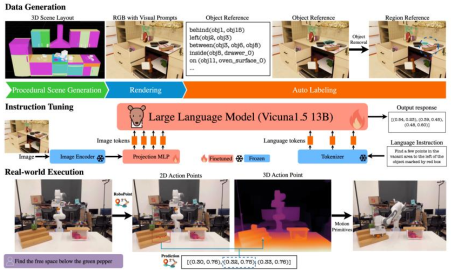
1) No neural rendering + masking communication which could be important for geometric downstream tasks

2) Single modality currently. Language/Audio as input?

Current/Future Work

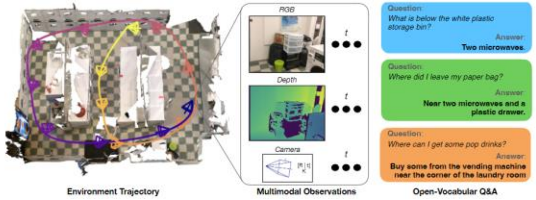
3D Vision and Robotics Foundation Models

- Trained on massive datasets on large compute
- Depth/poses/calibration is the key factors
- Most likely use an LLM due to the world knowledge it has obtained



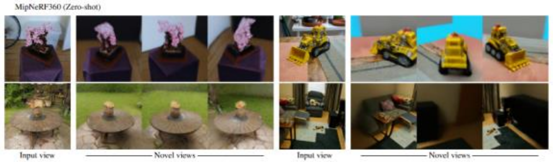
Benchmarking Robotics Foundation Models

- Need a common evaluation to validate the performance
- Robo-QA or spatial understanding could be early signs of success



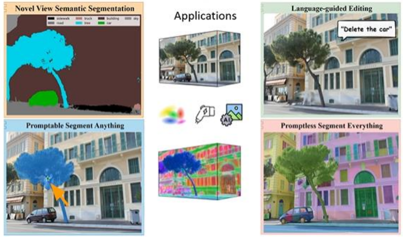
Data Augmentation through NVS and Diffusion Models

- We have other foundation models like ZeroNVS or Diffusion Models, so why not utilize them off-the-shelf for data augmentation?



Distilling 2D Foundation Models to 3D

- Distill powerful 2D models trained on billions of internet scale datapoints
- Some examples: semantic distillation into NeRFs



RoboPoint, arXiv 2024 | OpenEQA, CVPR2024 | ZeroNVS, CVPR 2024 | Feature3DGS, CVPR 2024

Thank you!

Zubair Irshad
Research Scientist
Toyota Research Institute

09/8/2024

zubairirshad.com