Seeing the Unseen

Complete 3D scenes from a single image

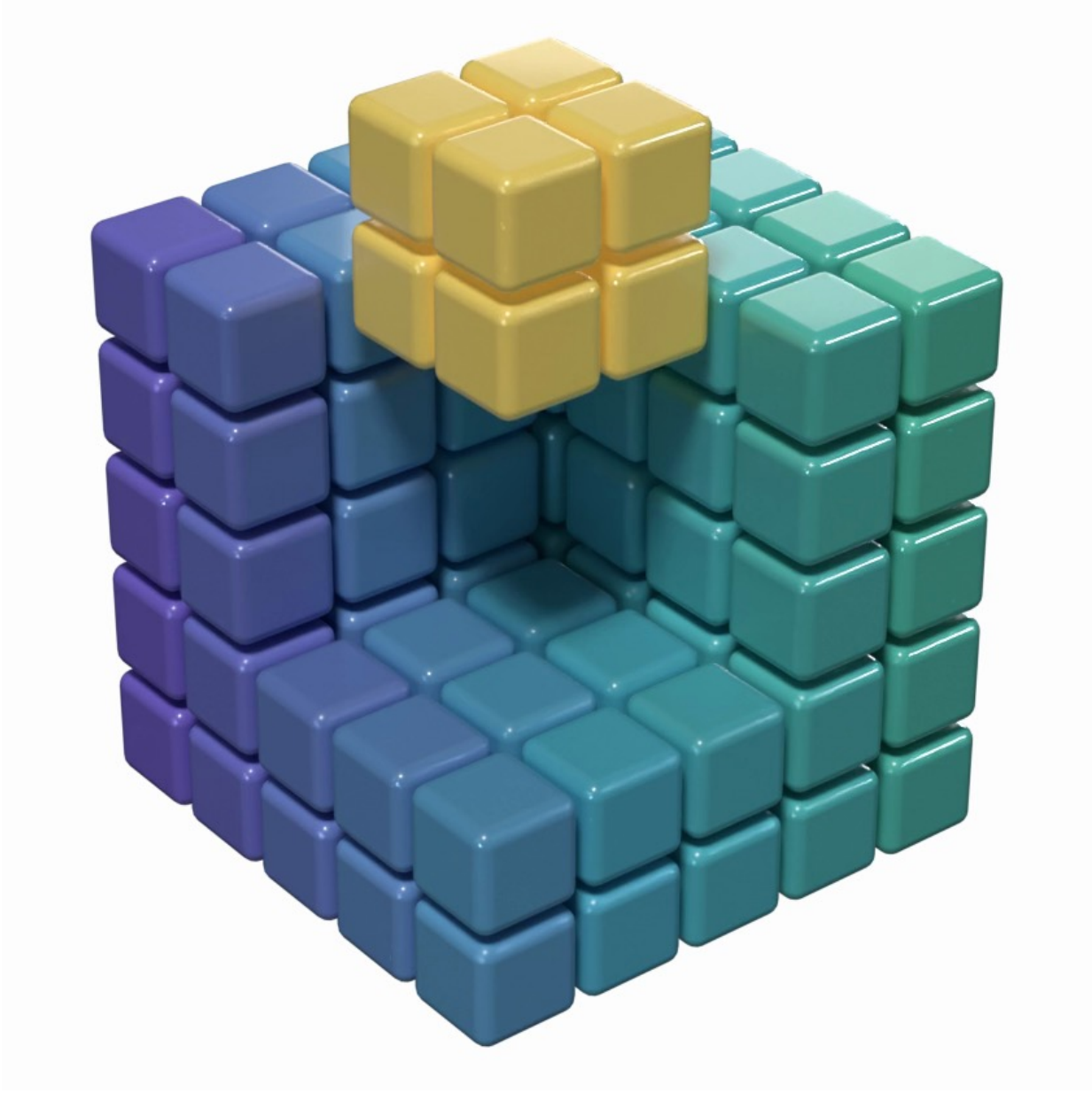
VolFill: Single-View Amodal 3D Scene Reconstruction with Volumetric Flow Matching

NeurIPS 2026

Tuan Duc Ngo

UMass Amherst

with Chuang Gan and Evangelos Kalogerakis



A photo shows only what faces the camera. We want the whole scene.



one photo

Photo: Brett Wharton / Unsplash



visible · from the photo

hidden · generated

Agents need the space they cannot see

01

Robots

plan paths and grasps behind furniture they have not seen yet

02

AR / VR

occlusion and physics need the full room, not one surface

03

Editing and world models

remove objects, move the camera, simulate the scene

> Humans infer hidden structure effortlessly. Machines mostly stop at the **visible surface**.

My research: perceiving the 4D world from images and video

Motion

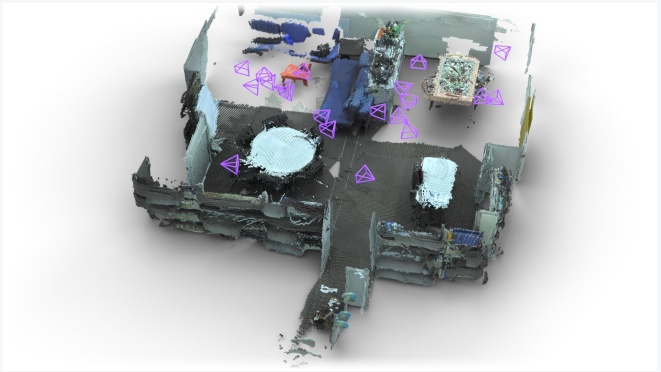
how every point moves



DELTA (ICLR 2025) · DELTAv2 (TPAMI)

Geometry

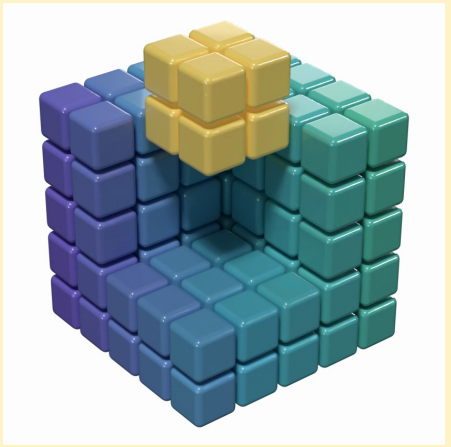
where every visible point is



DAGE (CVPR 2026)

The unseen

what lies behind the visible



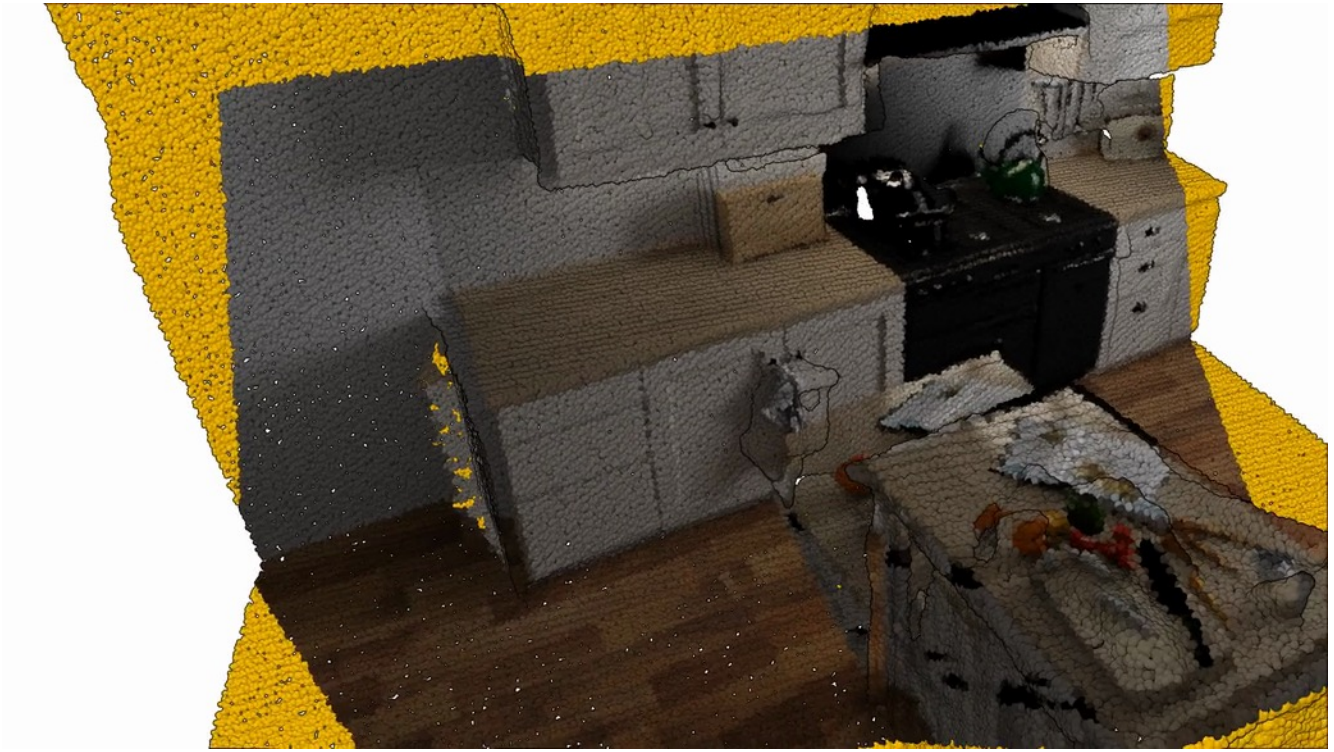
VolFill (NeurIPS 2026) · this talk

The task: from one photo, recover every surface in view, visible or hidden



Input: one RGB image

?



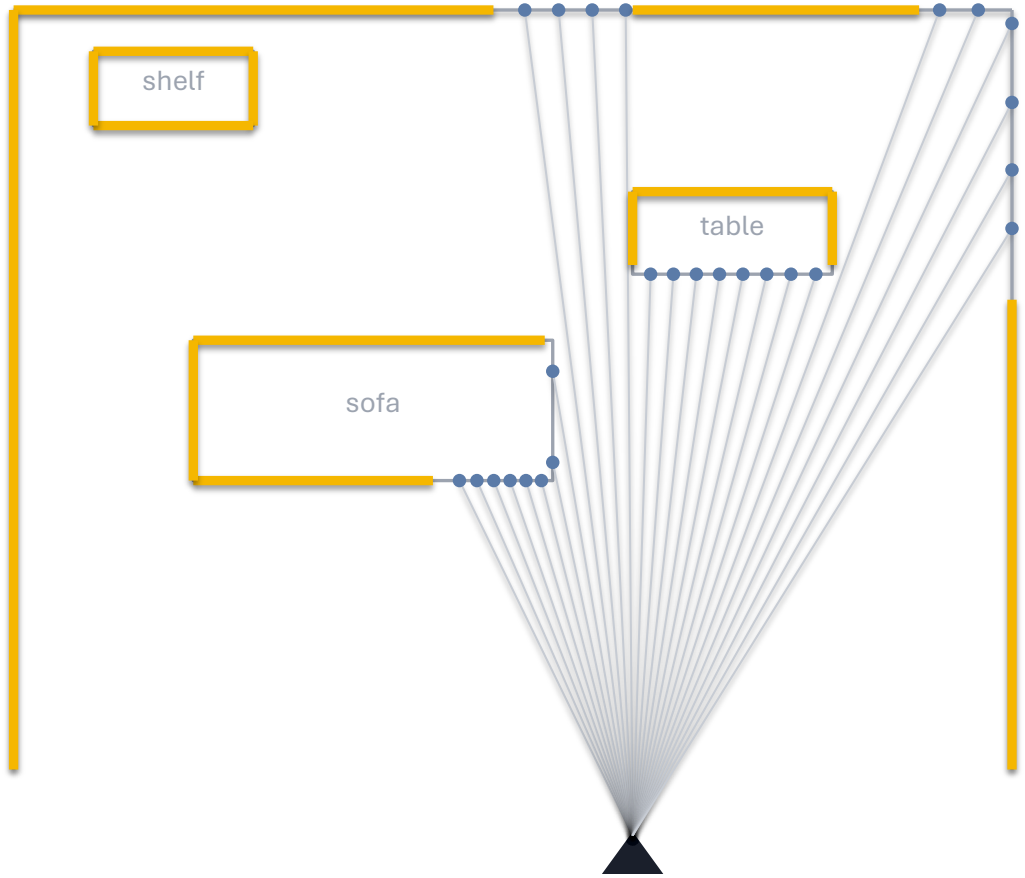
Output: the complete 3D scene in the camera's view

This is **amodal** reconstruction.

visible surfaces (photo colour)

hidden surfaces (gold)

Depth models see only the first surface along each ray



gold = surfaces no ray reaches



MoGe2's 3D points for one image, seen from the side

Methods that go beyond the visible return layered or noisy geometry

Layered rays

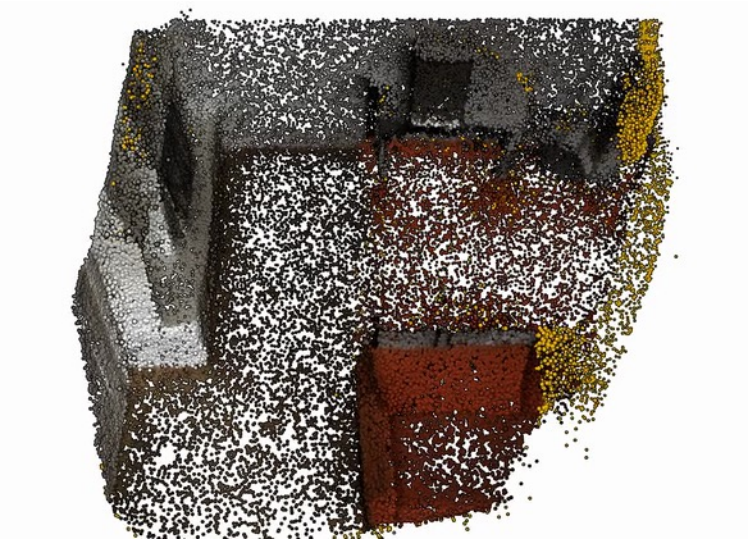
LaRI



a fixed number of hits per pixel → duplicated, broken walls

Point generation

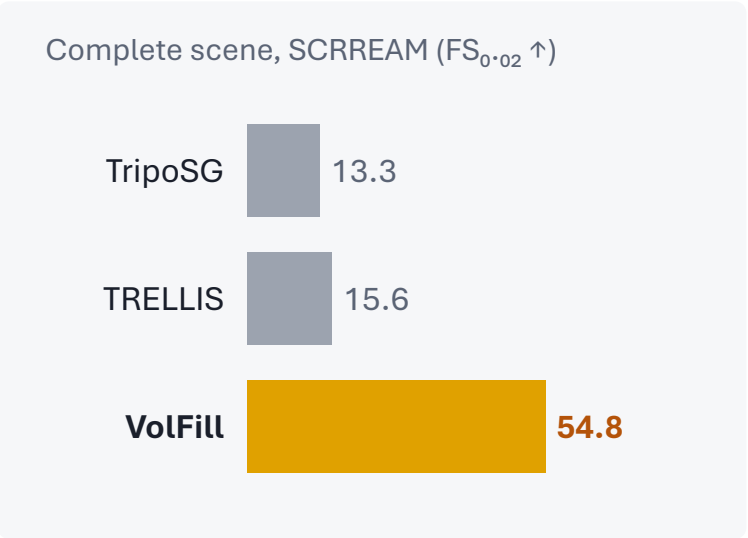
NOVA3R



an unordered point cloud → noise, floating surfaces

Object generators

TRELLIS · TripoSG



trained on single objects → a room breaks apart

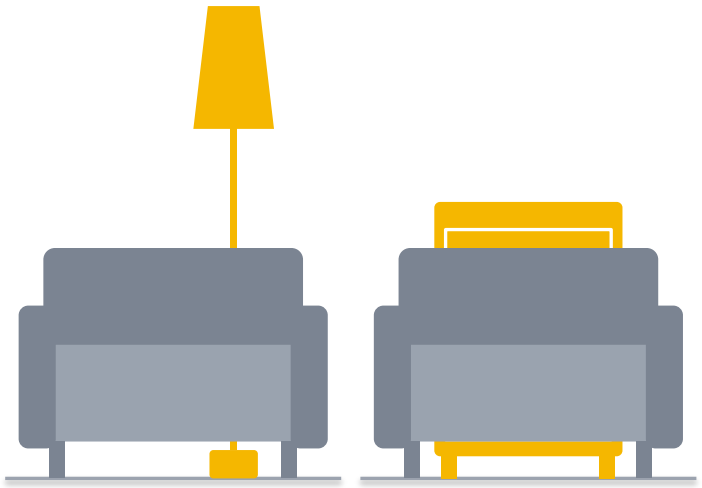
We want all three: complete, structured, and grounded in the photo

	Complete	Structured	Grounded
Depth / pointmap models MoGe2, DA3, DepthPro	✗	✓	✓
Layered rays LaRI	◐	✗	✓
Point generation NOVA3R	✓	✗	◐
Object generators TRELLIS, TripoSG	✗	✓	✗
VolFill (ours) this talk	✓	✓	✓

✓ yes ◐ partly ✗ no

Hidden geometry is ambiguous, so we generate it rather than regress it

Two plausible scenes



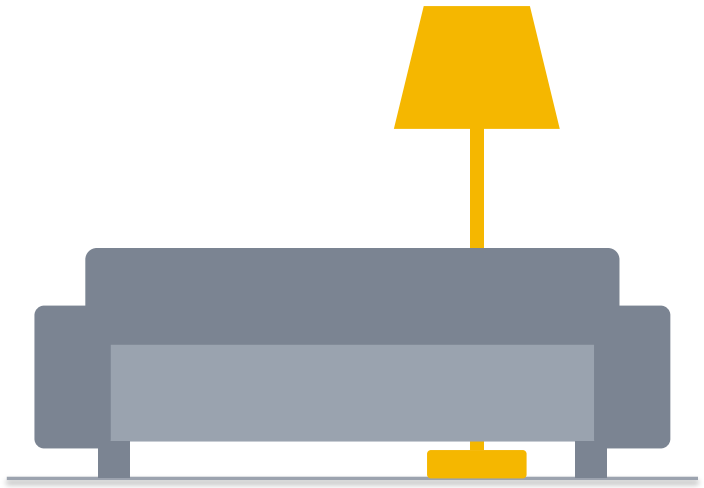
a lamp? a cabinet? both fit the photo

Regression → the average



neither object: a blurry ghost

Generation → one sharp sample



sharp and plausible

> So VolFill is **generative**: it samples a complete scene instead of averaging.

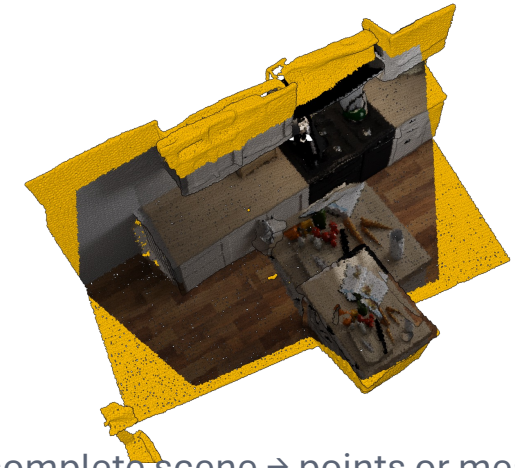
Complete **Structured** **Grounded**

VolFill: generate the whole scene as a 3D volume, guided by what the photo shows



photo + visible surfaces

Volumetric generator
samples a 3D volume of the whole scene



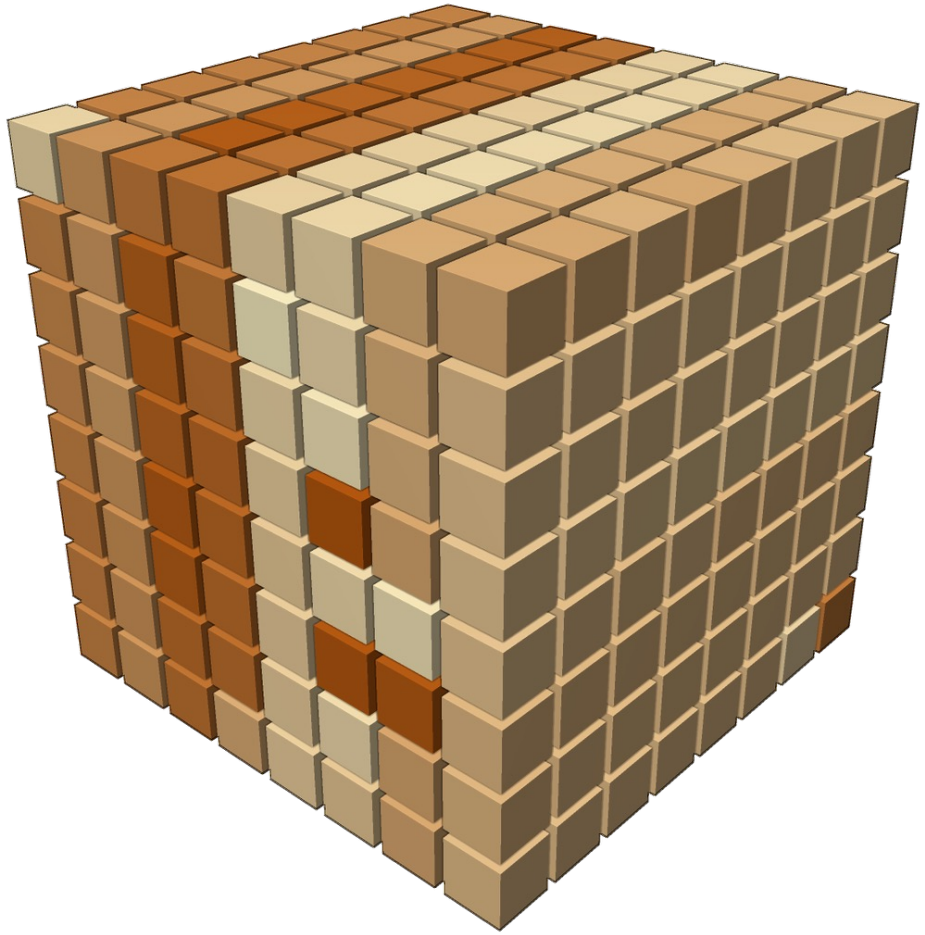
complete scene → points or mesh

1 Representation → **2 Compression** → **3 Generation** → **4 Grounding**

VolFill

Generate the whole scene as a 3D volume

NeurIPS 2026



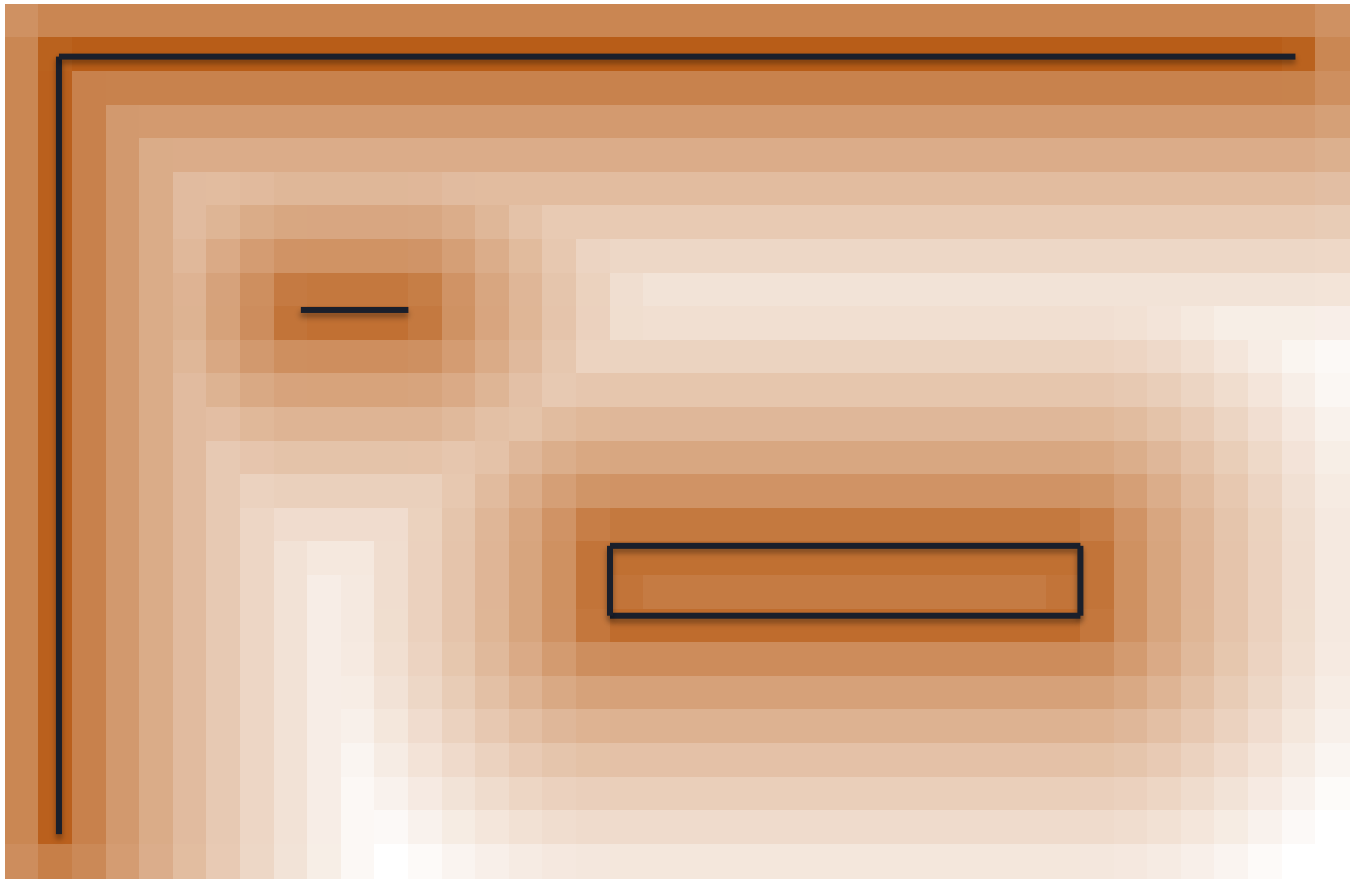
1 Representation → 2 Compression → 3 Generation → 4 Grounding

Represent the scene as a distance field, truncated near surfaces



1 Take a slice through the room: its surfaces

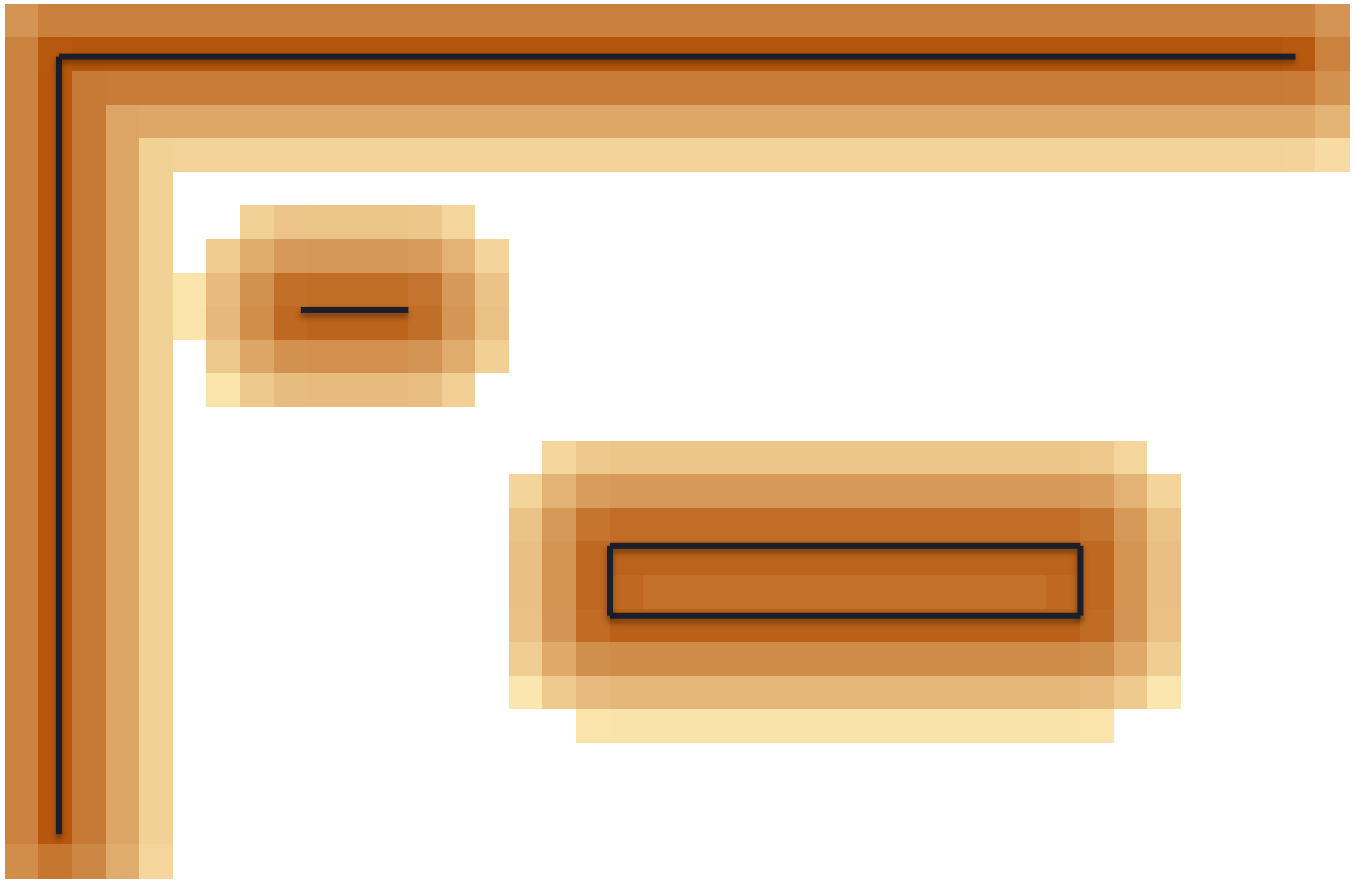
Represent the scene as a distance field, truncated near surfaces



1 Take a slice through the room: its surfaces

2 Every cell stores the distance to the nearest surface

Represent the scene as a distance field, truncated near surfaces



1 Take a slice through the room: its surfaces

2 Every cell stores the distance to the nearest surface

3 Keep only a thin band near the surfaces

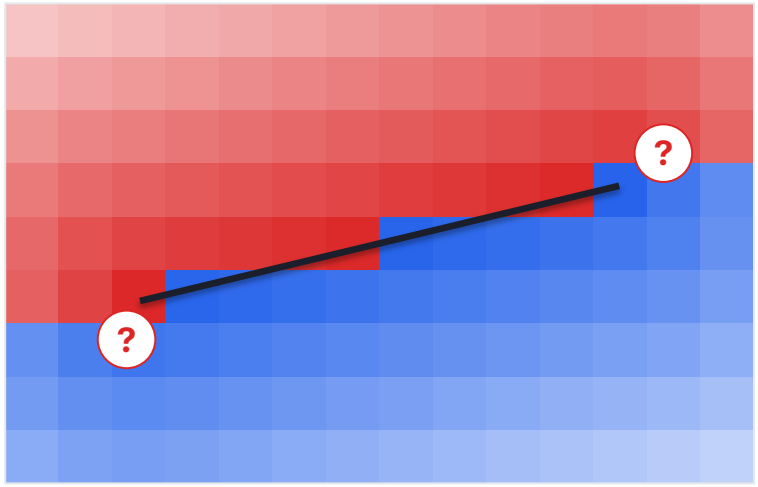
$$V(\boldsymbol{p}) \;=\; \min(\,\mathrm{dist}(\boldsymbol{p}, \mathrm{surfaces}), \tau\,)$$

distance to the nearest surface, capped at **τ = 3 voxels**

TUDF: truncated unsigned distance field

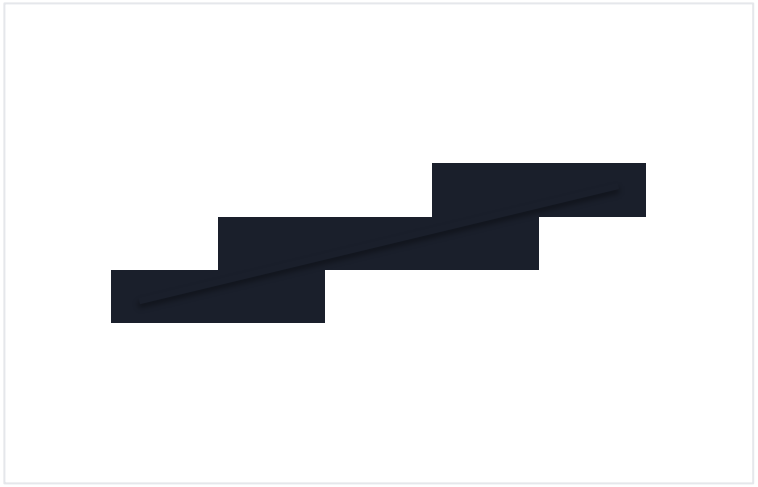
Unsigned distances suit indoor scenes, whose surfaces are open

Signed distance



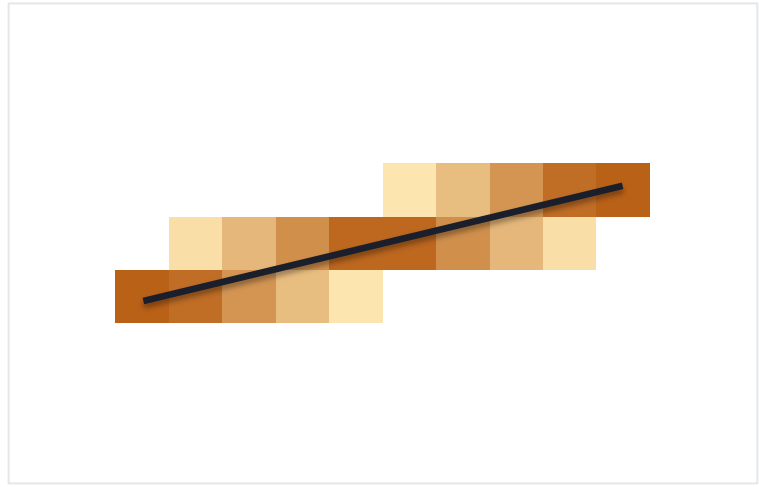
inside vs outside? undefined for an open wall

Occupancy



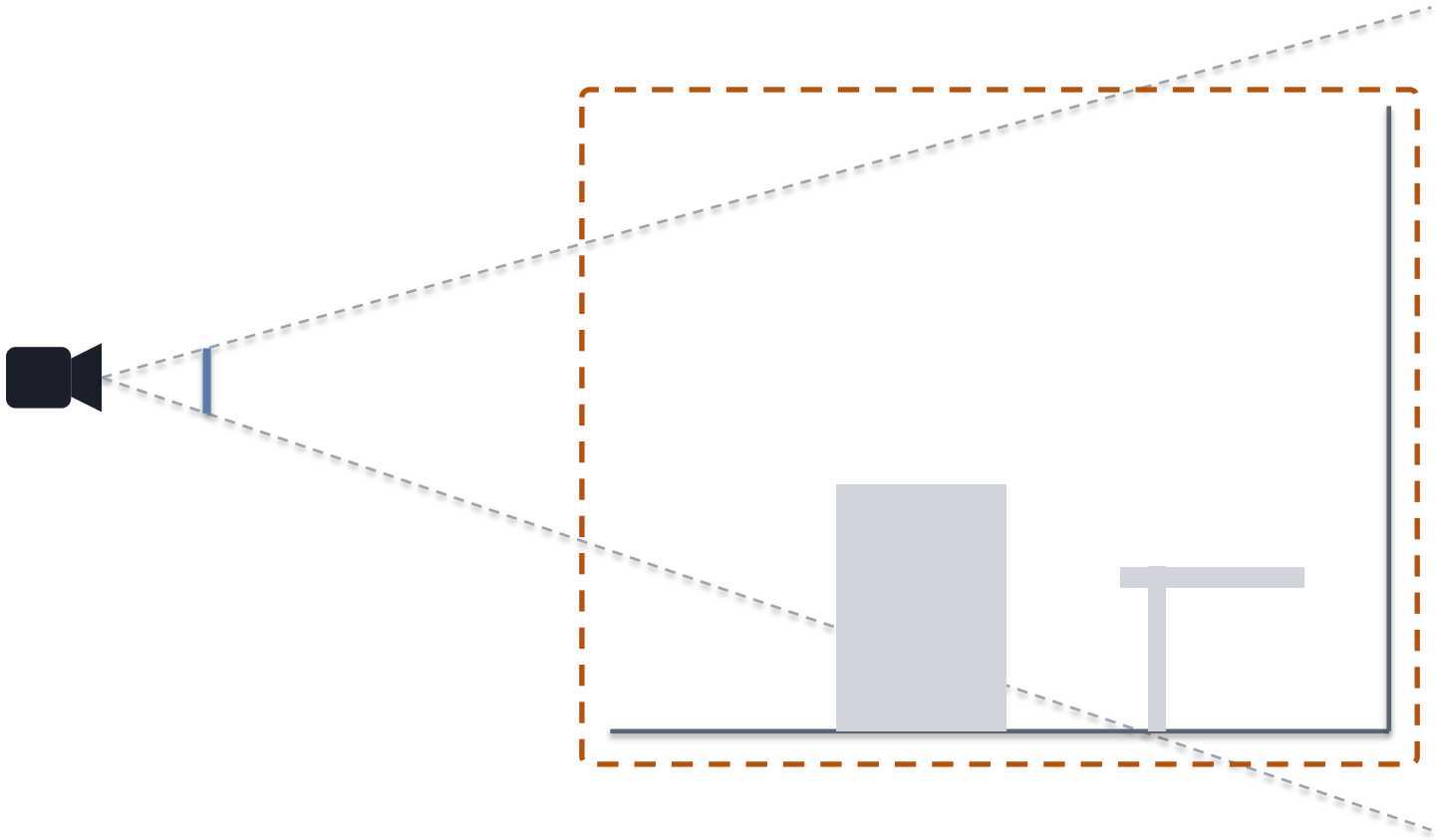
surface or not: no distance information

Unsigned distance (TUDF)



well defined for any surface; smooth near it

A 256^3 grid placed in the camera's view covers the scene



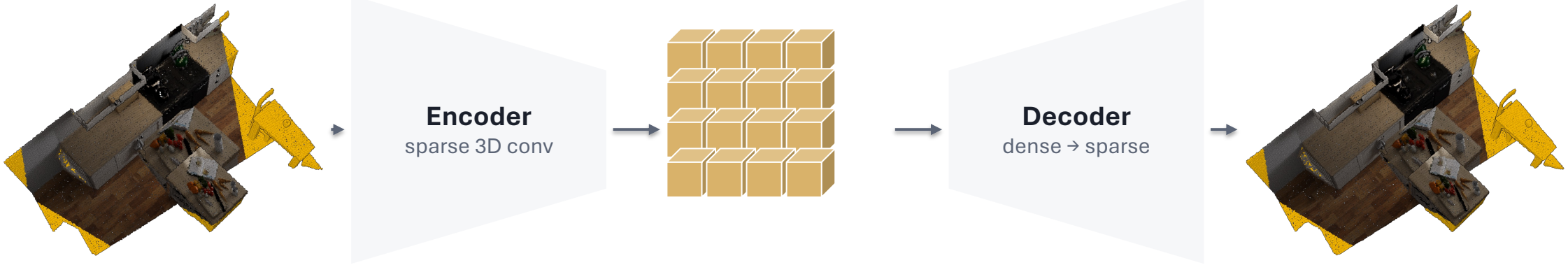
256^3

voxels per scene

≈ 16.8 million cells

Too many to generate directly → **compress it first**.

Compress the volume: a 3D VAE turns 16.8 M voxels into 65 K numbers



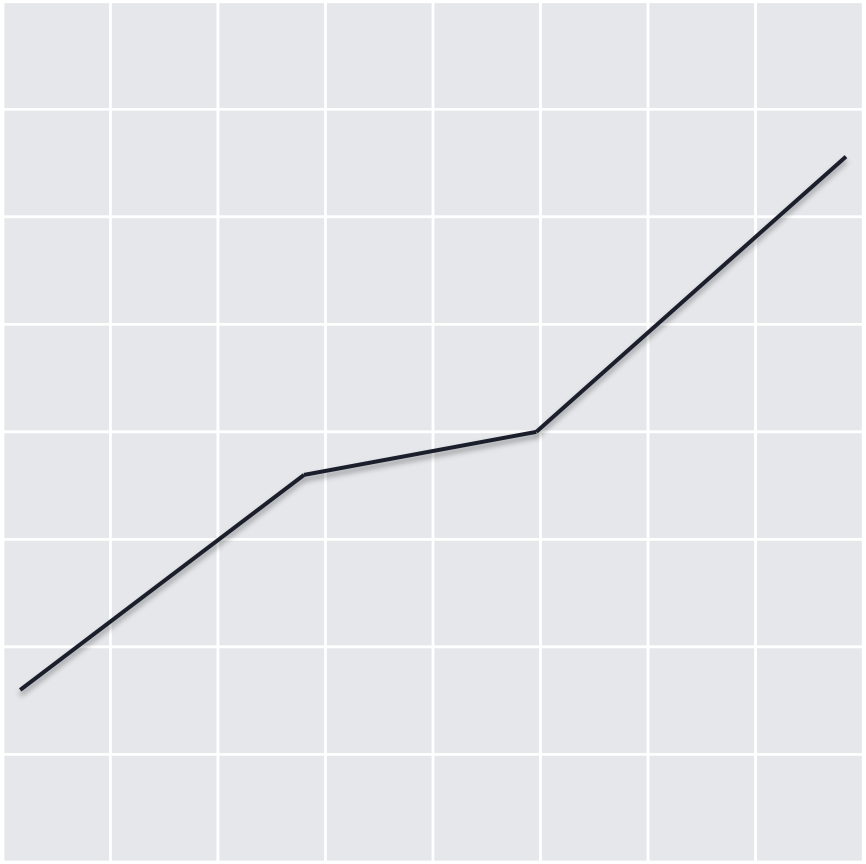
256^3 volume

$16^3 \times 16$ latent

reconstruction

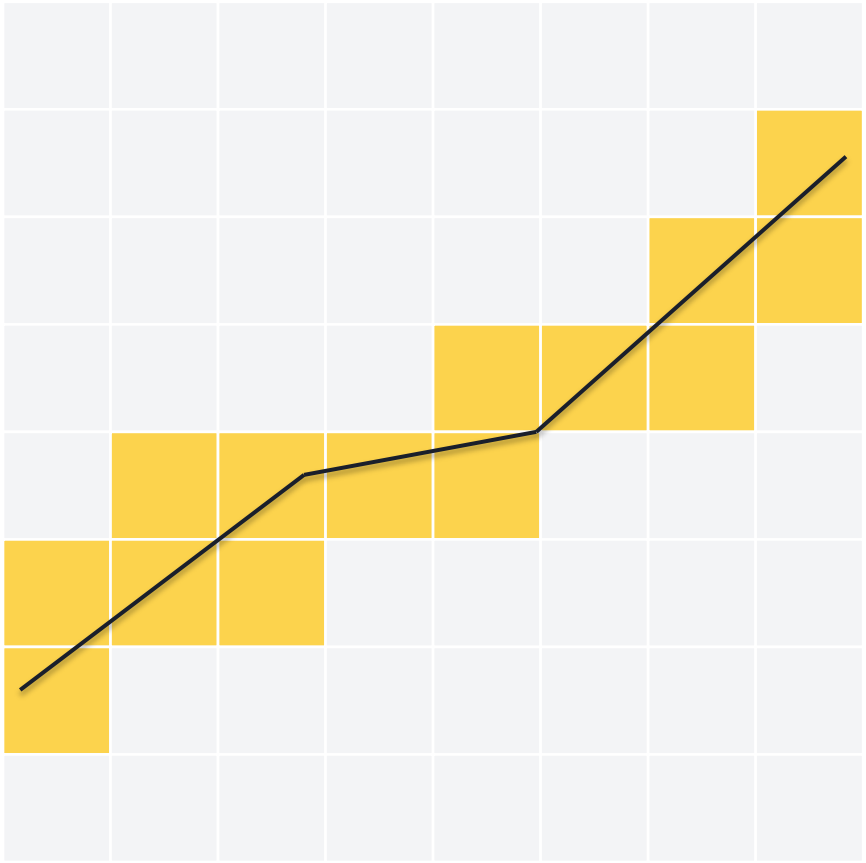
256× smaller

Decode densely where it is cheap, sparsely where it matters



1 Decode densely at low resolution ($16^3 \rightarrow 64^3$)

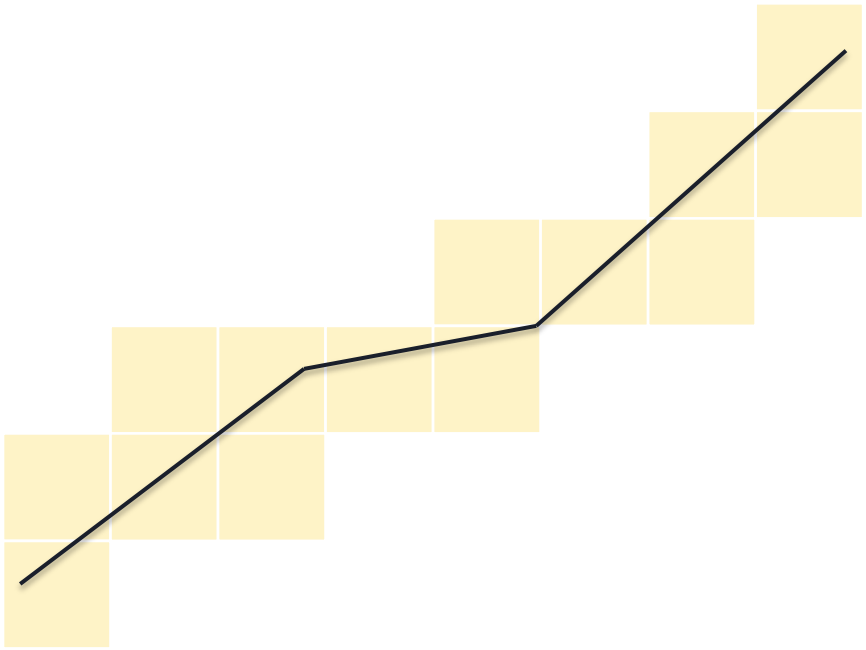
Decode densely where it is cheap, sparsely where it matters



1 Decode densely at low resolution ($16^3 \rightarrow 64^3$)

2 Predict which cells contain a surface

Decode densely where it is cheap, sparsely where it matters

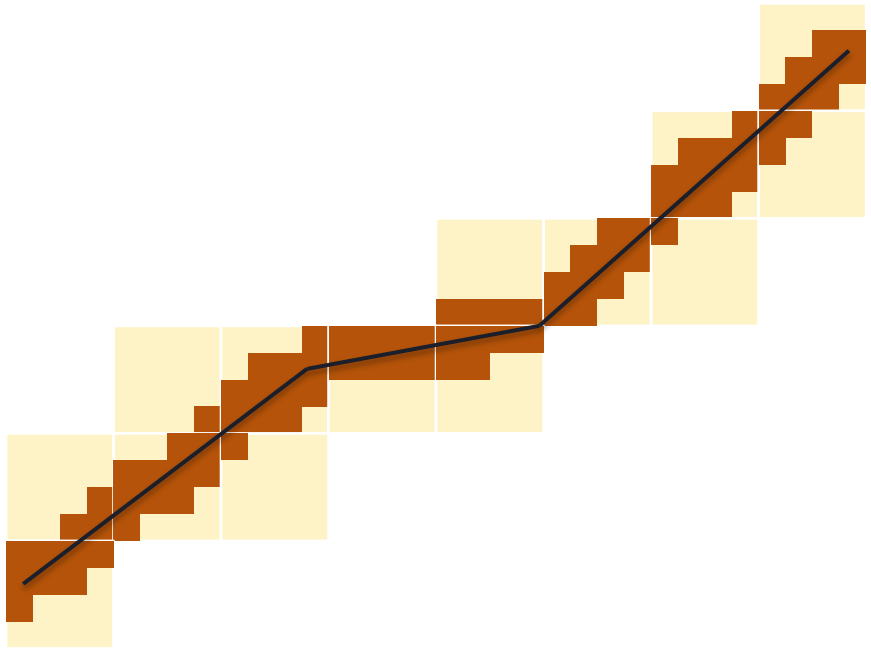


1 Decode densely at low resolution ($16^3 \rightarrow 64^3$)

2 Predict which cells contain a surface

3 Drop the empty cells

Decode densely where it is cheap, sparsely where it matters



1 Decode densely at low resolution ($16^3 \rightarrow 64^3$)

2 Predict which cells contain a surface

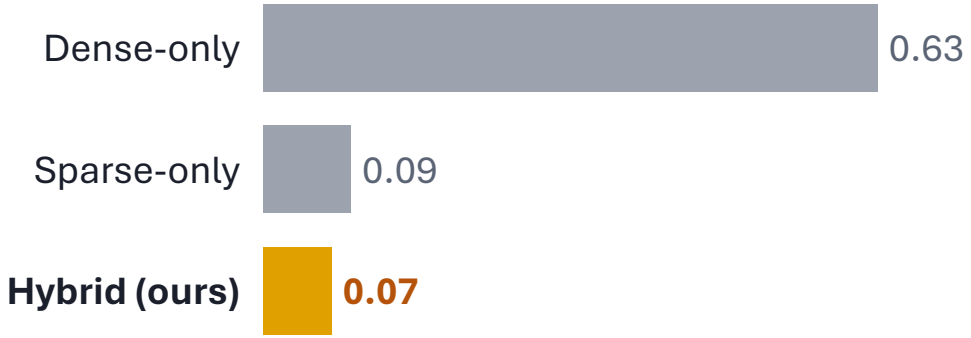
3 Drop the empty cells

4 Refine only near surfaces ($64^3 \rightarrow 256^3$)

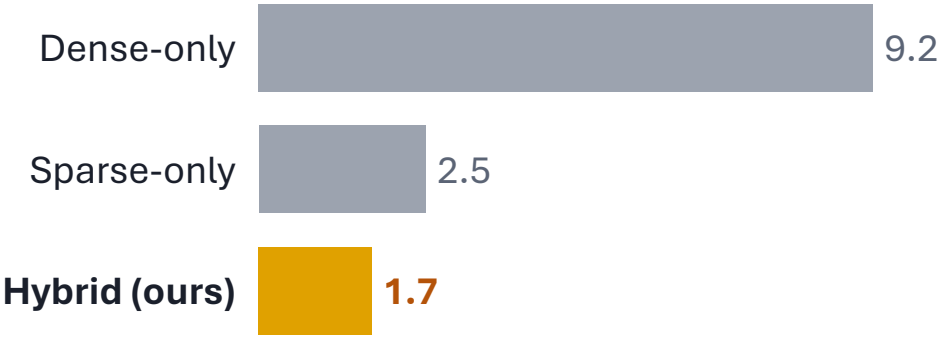
Complete Structured Grounded

The hybrid decoder is the fastest and lightest, with the lowest error

Decoding time (s) ↓



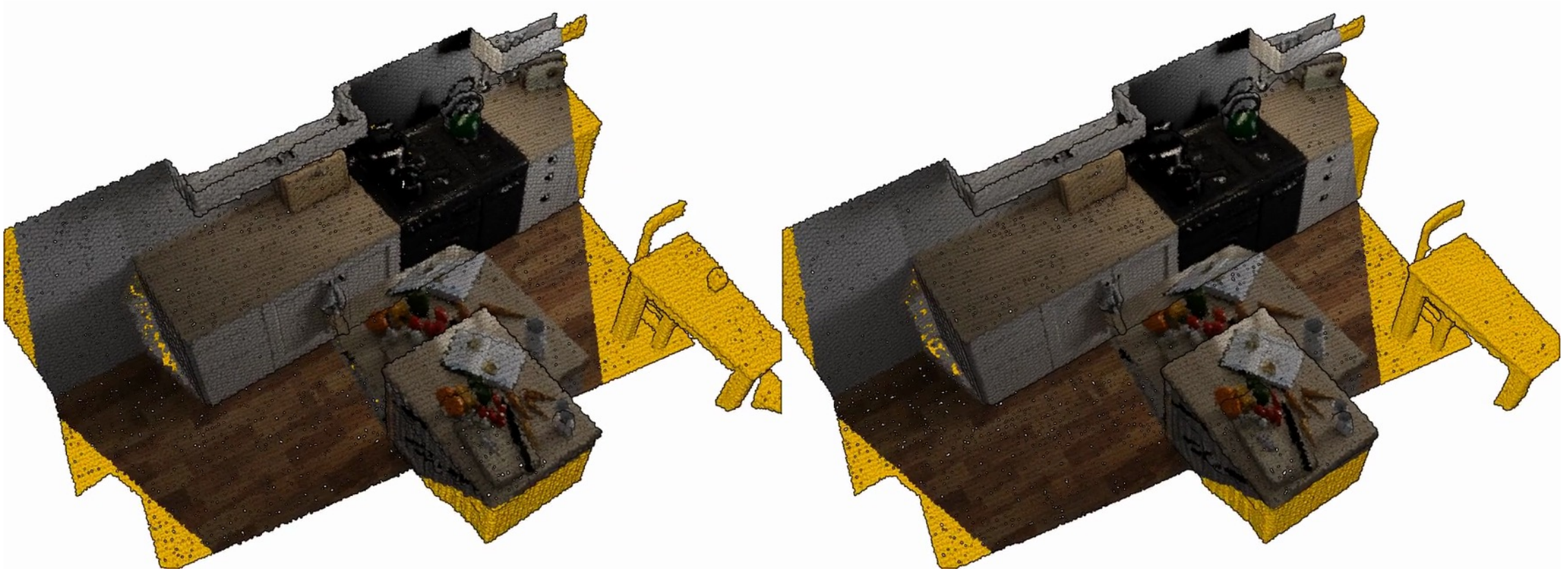
GPU memory (GB) ↓



	L1 distance ↓	Surface IoU ↑
Dense-only	0.49	93.2
Sparse-only	0.48	93.0
Hybrid (ours)	**0.43**	**93.5**

9× faster and **5× less memory** than dense decoding, and more accurate.

The compressed latent keeps the scene: reconstructions are near-identical

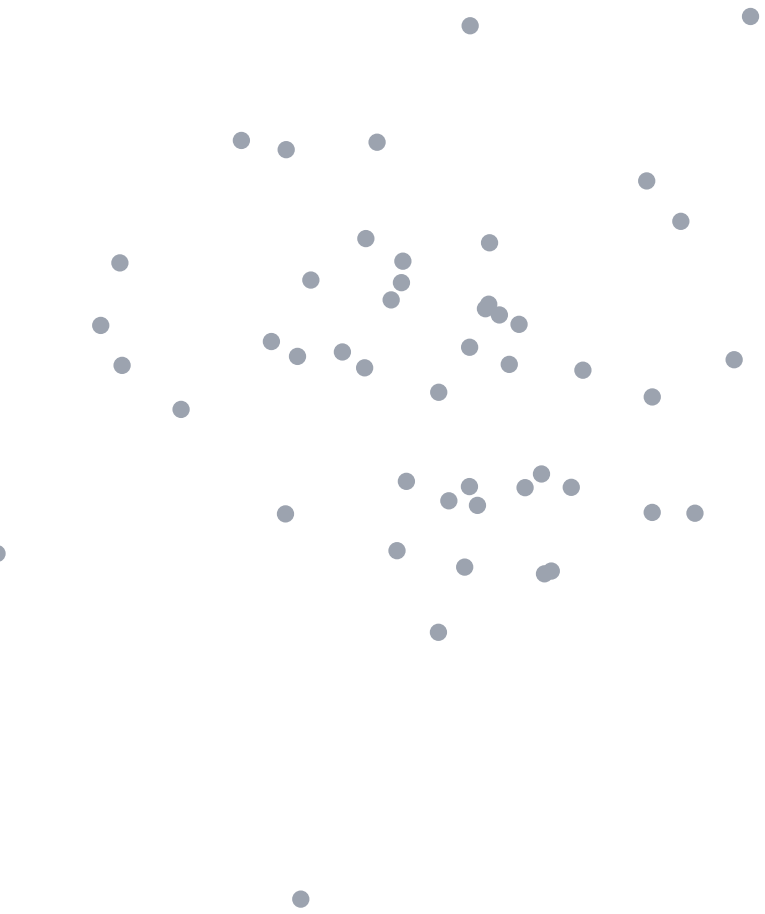


ground truth

VAE reconstruction

93.5% surface IoU between the ground truth and its reconstruction

Flow matching: learn to carry noise to data along straight paths

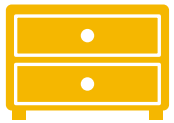




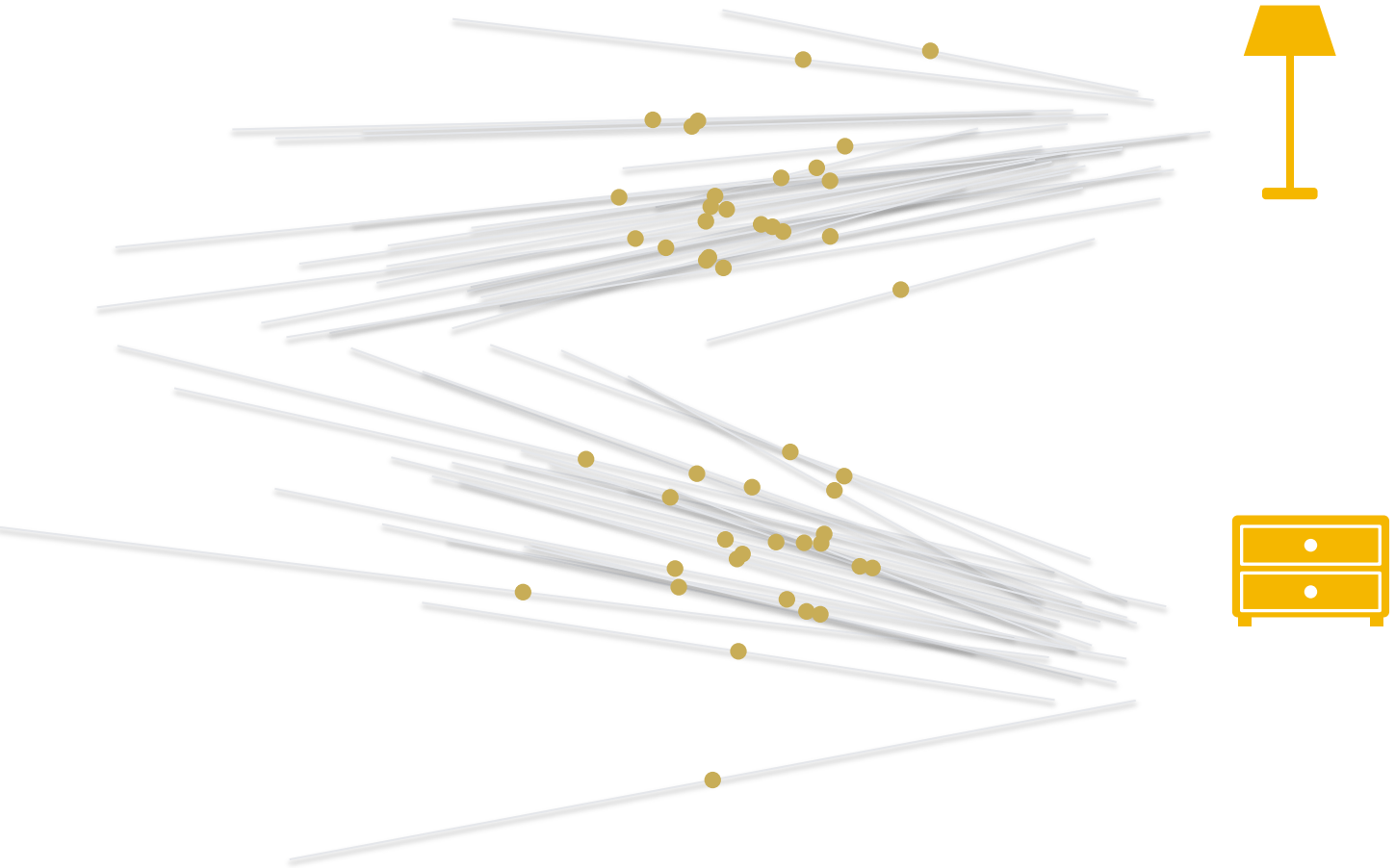
Start from random noise

every dot is one random sample: a whole scene, drawn as a single point

`t = 0.0`



Flow matching: learn to carry noise to data along straight paths



Follow a learned velocity field

a network predicts which way each sample should move

`t = 0.5`

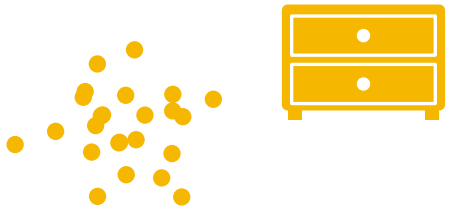
Flow matching: learn to carry noise to data along straight paths



Arrive at plausible scenes

here two modes: the lamp and the cabinet behind the sofa

`t = 1.0`



$$x_t = (1 - \mathrm{t}) \cdot \mathrm{noise} + \mathrm{t} \cdot \mathrm{data}$$

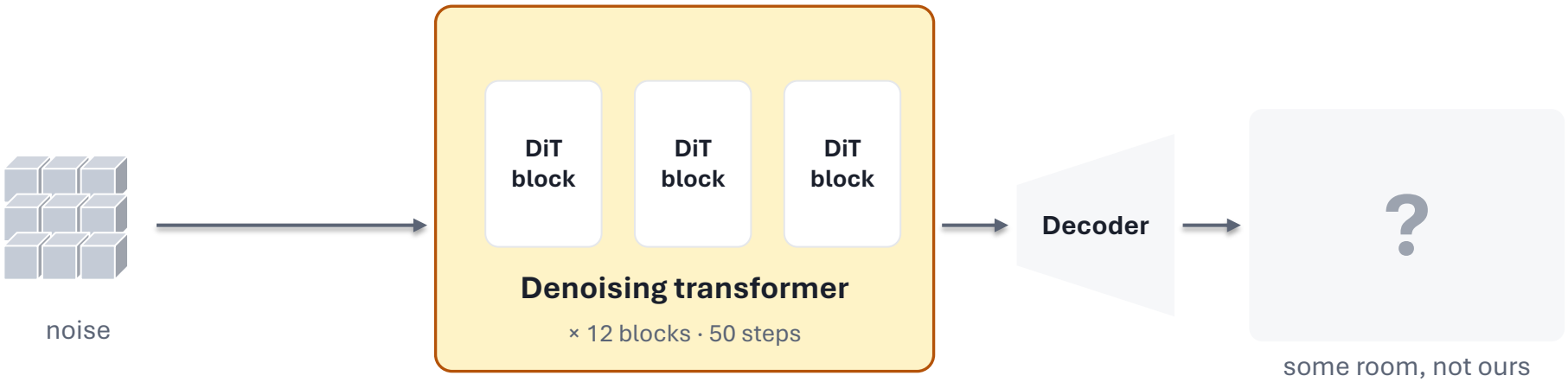
the network learns the direction **data – noise**

In VolFill's latent space, the room emerges coarse to fine in 50 steps

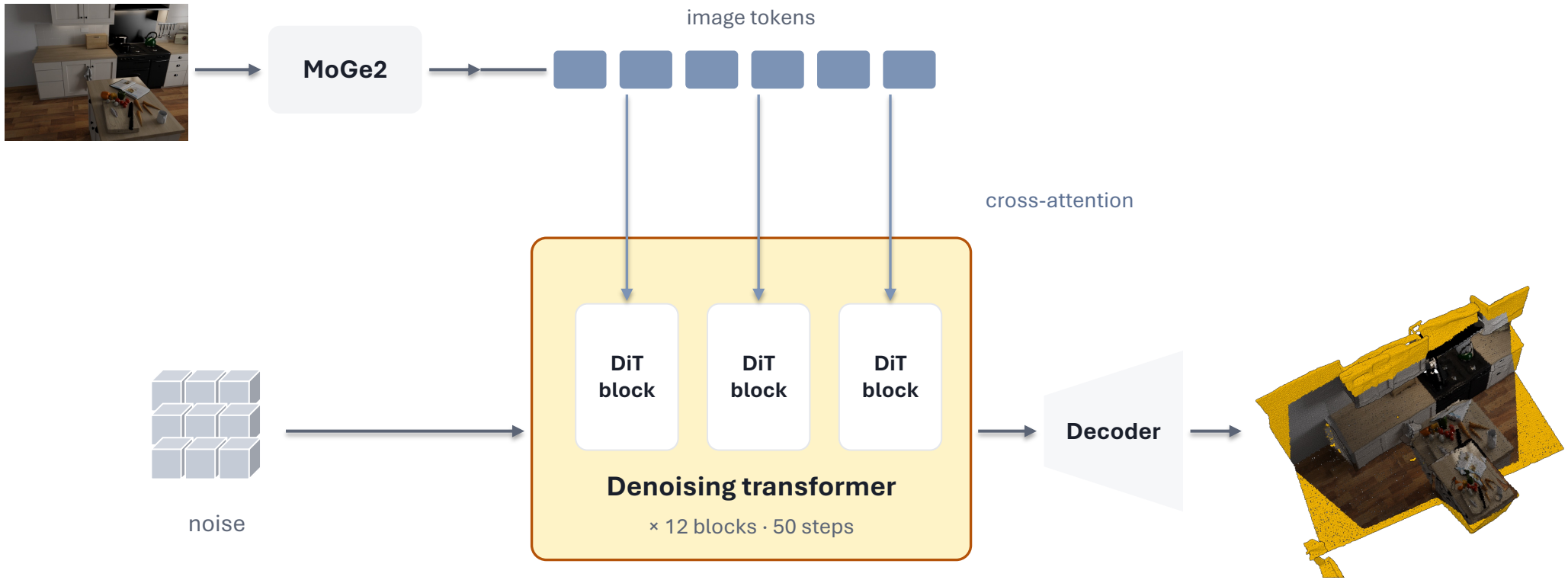


Layout first, details later · **50 steps** · trained on 142 K image–volume pairs (3D-FRONT + ScanNet++)

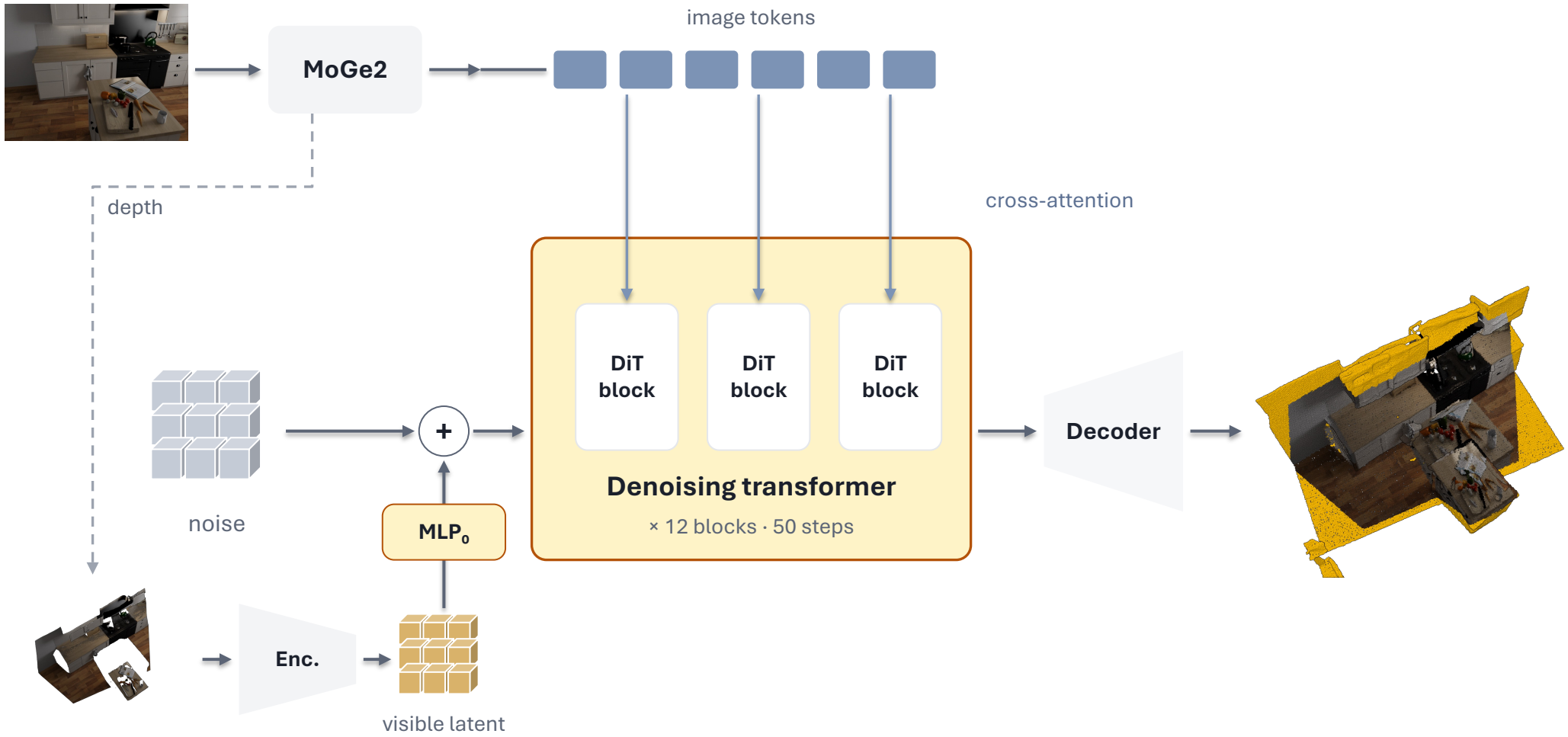
Without guidance, the generator invents an arbitrary room



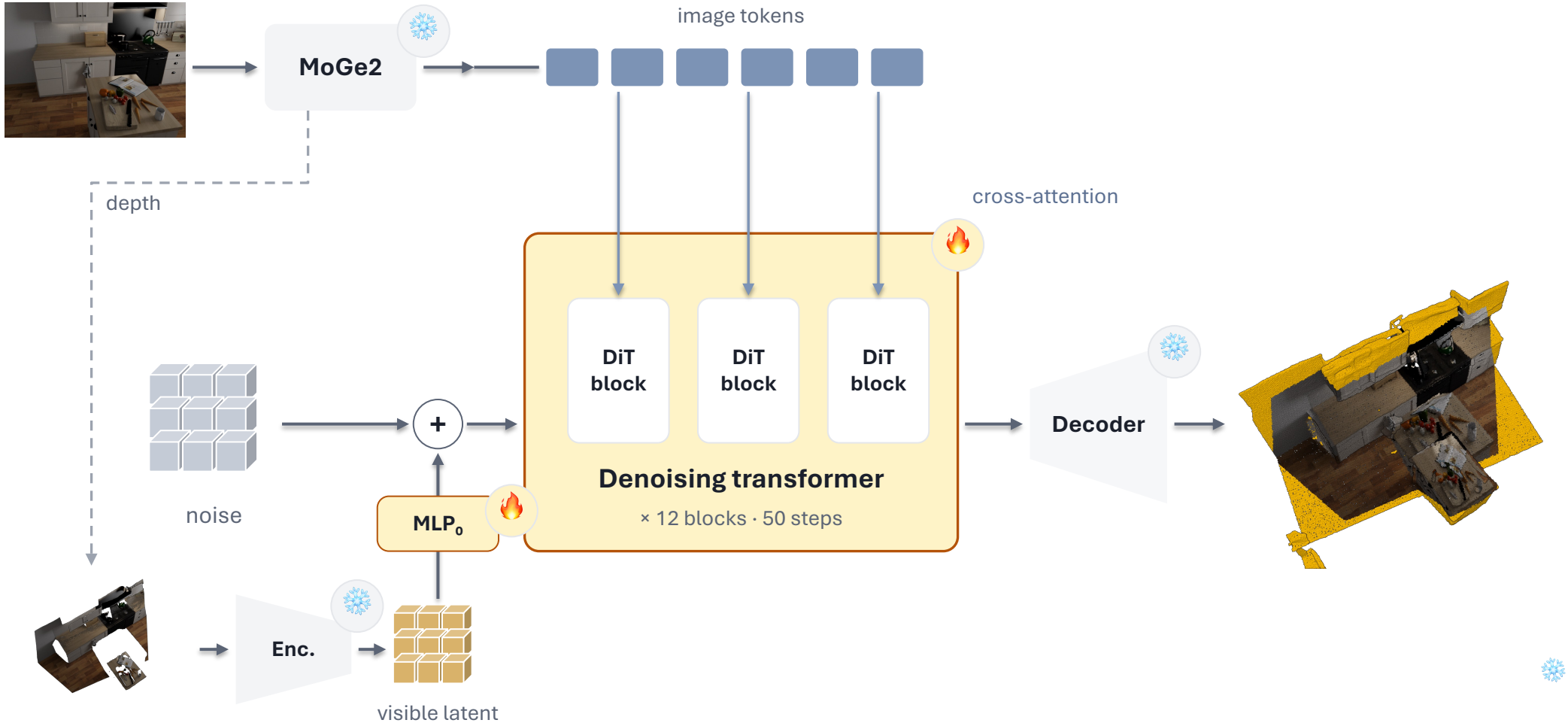
Grounding 1: image features from a frozen geometry model, via cross-attention



Grounding 2: the visible geometry, encoded and added to the noise



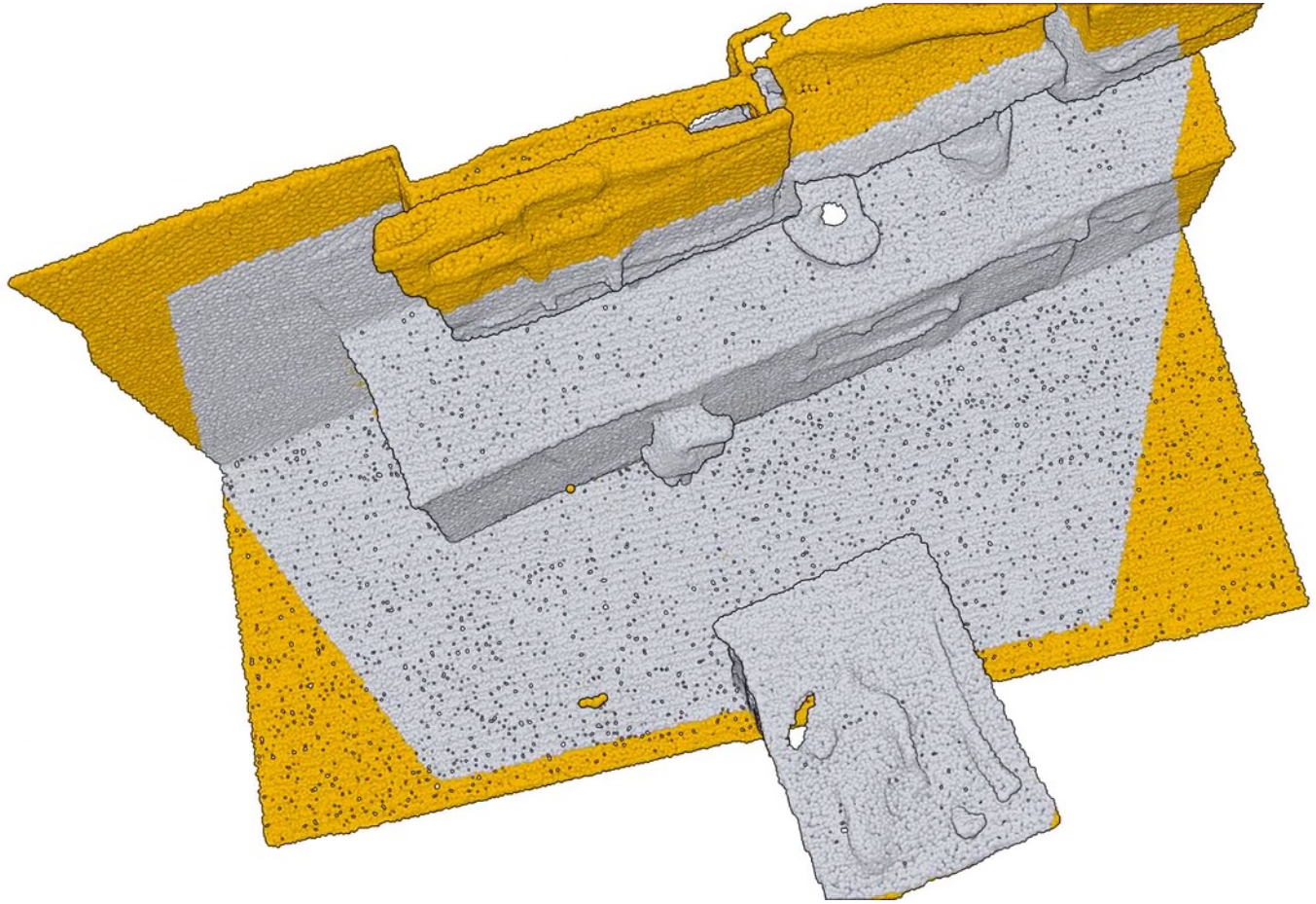
The full model: one photo in, a complete 3D scene out in 1.4 s



One photo in, the whole room out



input



only the generated part

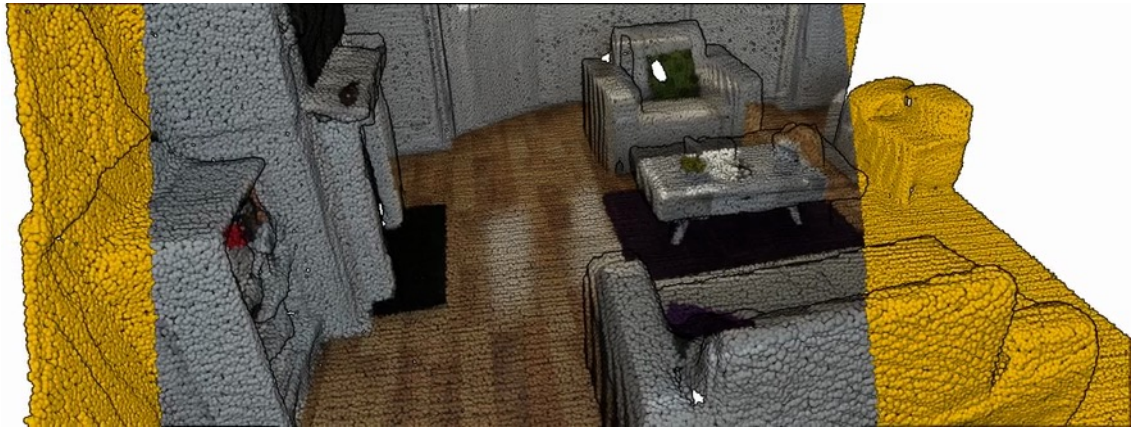
Depth models leave holes behind objects; VolFill fills them

MoGe2 (depth / pointmap)





VolFill (ours)



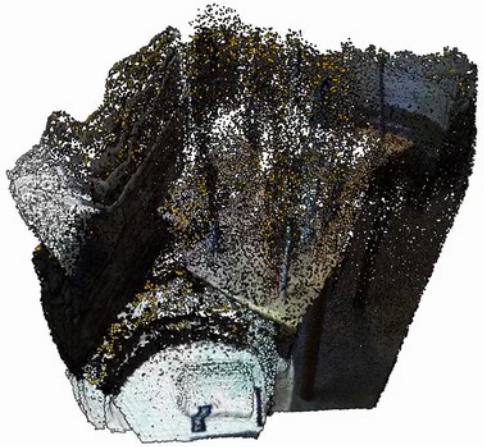


Against amodal methods: no duplicated layers, no floating noise

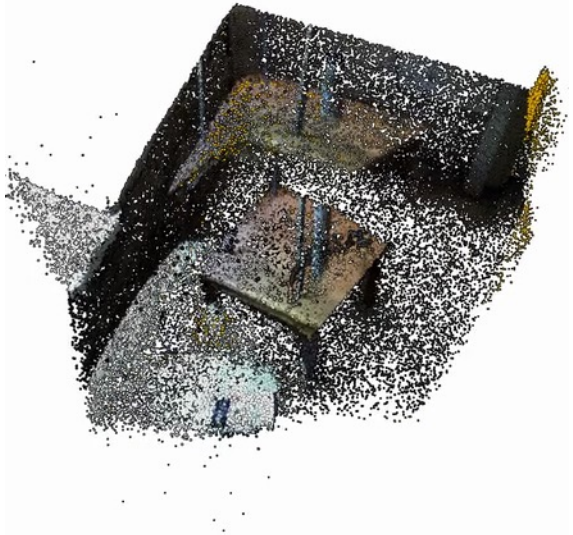
Input



LaRI



NOVA3R



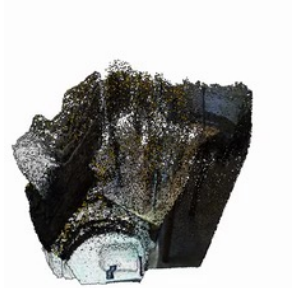
VolFill (ours)



A clean volume gives a clean mesh, directly



Baselines output points, not surfaces

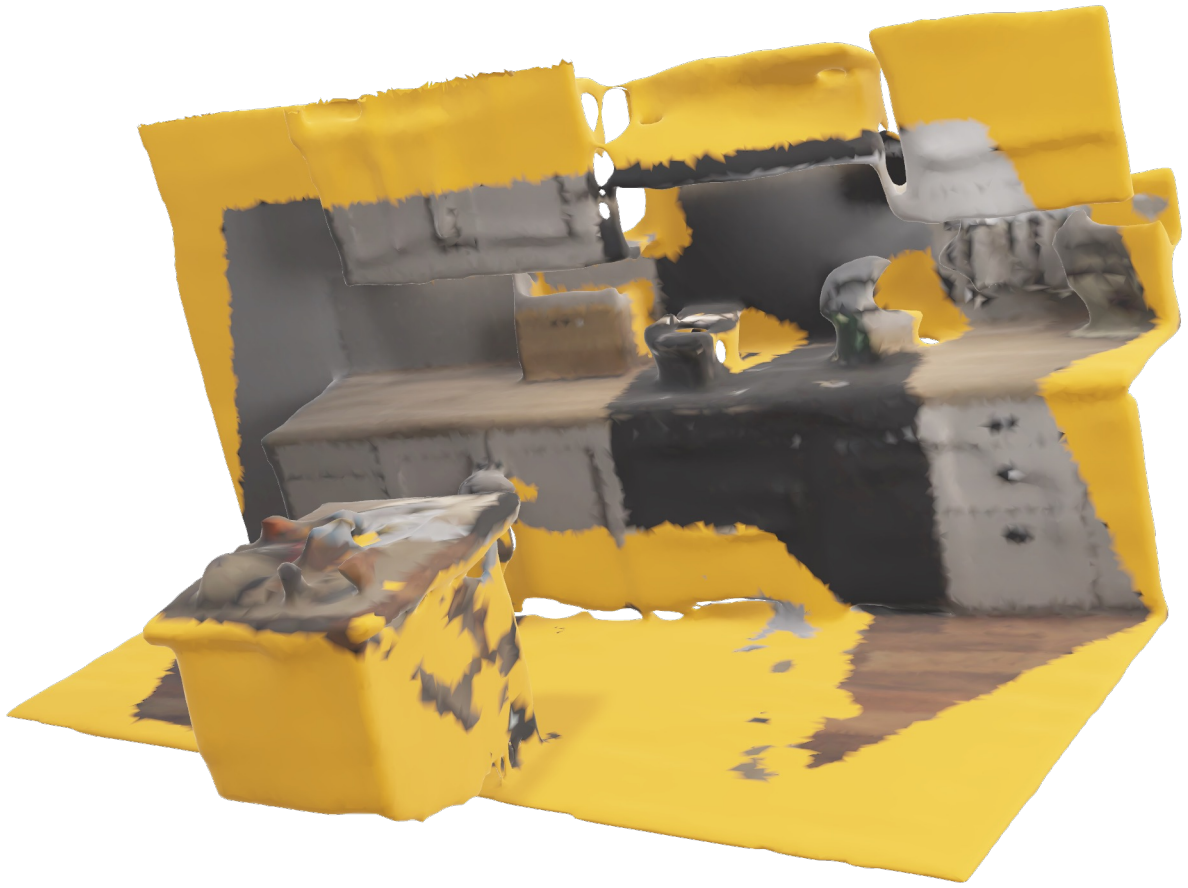


LaRI · layered points

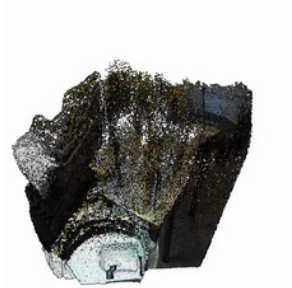


NOVA3R · noisy points

A clean volume gives a clean mesh, directly



Baselines output points, not surfaces



LaRI · layered points



NOVA3R · noisy points

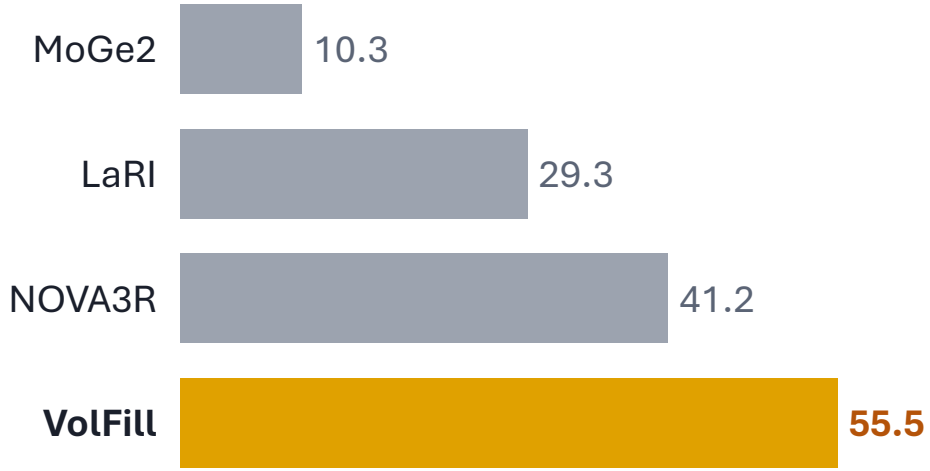
It generalizes to photos it has never seen



At a strict threshold, VolFill recovers the most hidden geometry on SCRREAM

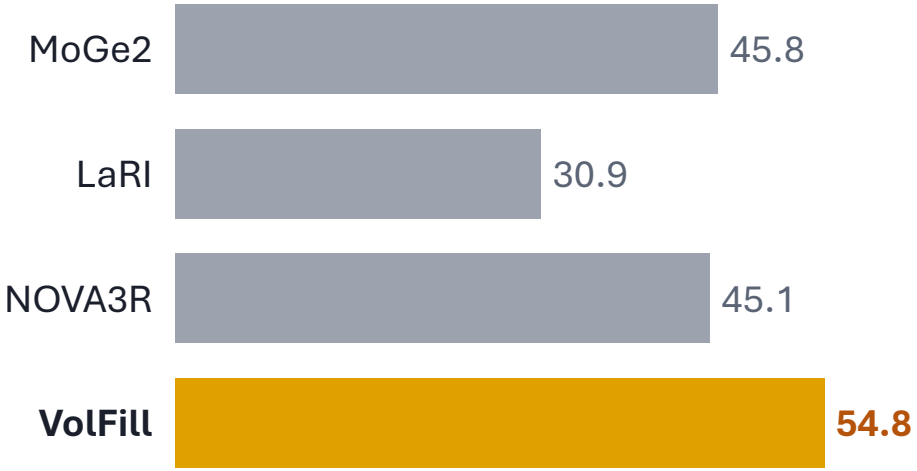
Hidden surface recovered (%) ↑

share of hidden ground truth within 0.02 of the prediction (APD)



Complete scene, F-score (%) ↑

precision and recall at 0.02 over the whole scene (FS)



Hidden surface recovered: **55.5%** vs 41.2% (NOVA3R) · SCREAM: 460 images of real scanned rooms

It holds on a second benchmark and at real-world scale

NRGB-D · hidden surfaces

6.97

Chamfer distance ↓ (average error) vs 8.31 NOVA3R, 13.04 LaRI

Metric scale · F-score

44.9

$FS_{0.02}$ ↑ vs 22.2 NOVA3R (2×)

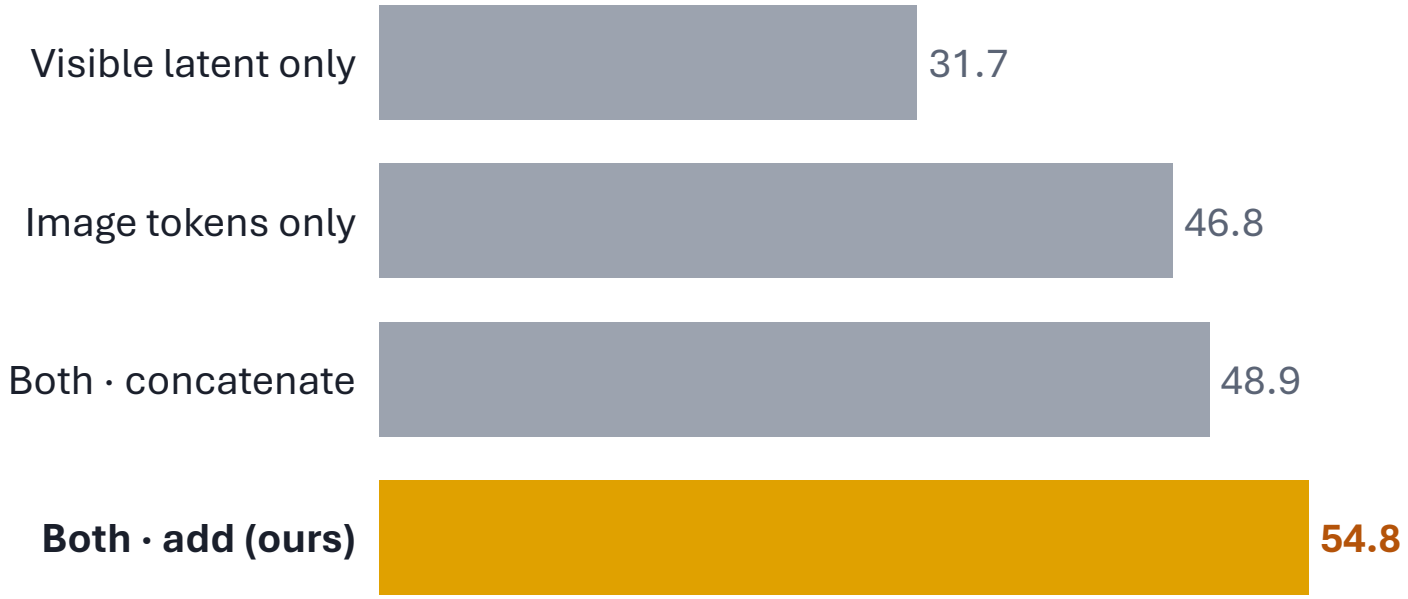
Realism · FPD

4.86

distance to real-scene statistics ↓ vs 16.50 NOVA3R (3.4× lower)

Both kinds of guidance are needed, and adding beats concatenating

Complete-scene F-score at 0.02, SCRREAM (%) ↑



Adding puts the visible geometry **in the same space** as the sample being denoised: generation becomes inpainting.

Even purely semantic features allow completion; 3D-aware features help most

Image tokens only: complete-scene F-score at 0.02, SCRREAM (%) ↑



> Completion works with any image features; **3D-aware features (MoGe2) work best**.

The remaining error is in generation, not in the representation

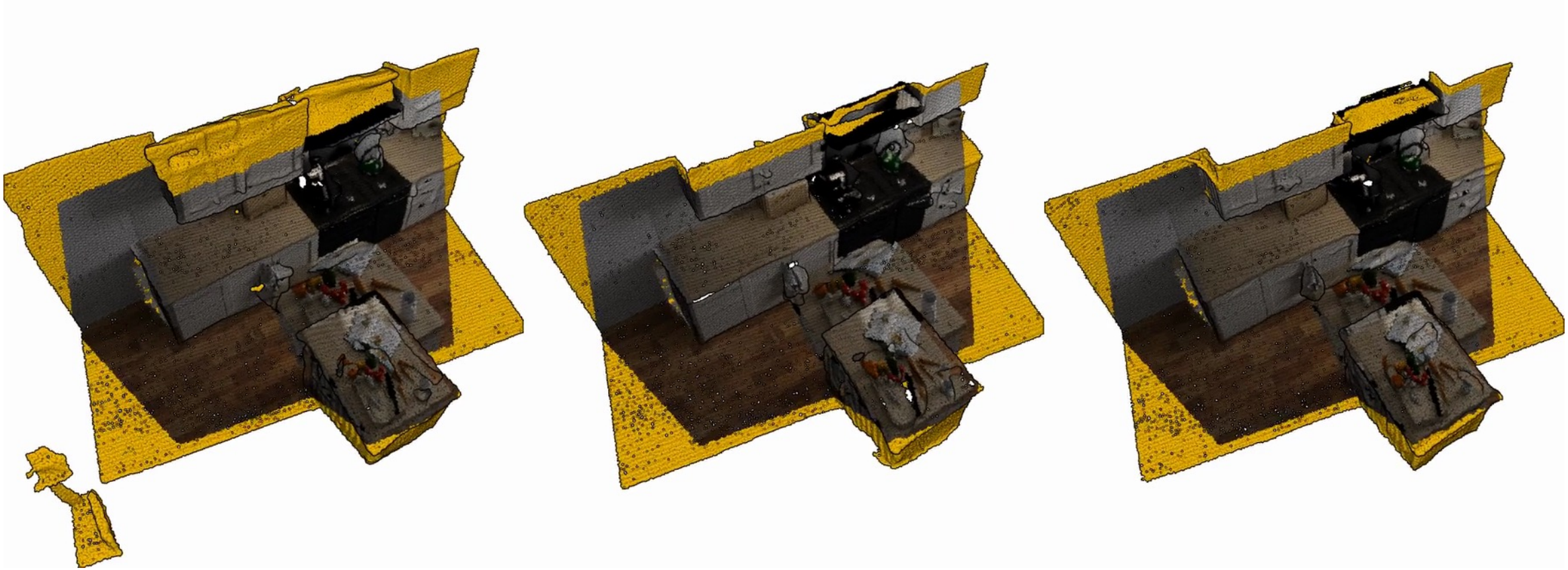
F-score at the strict 0.02 threshold, SCRREAM (%)



ground truth → 256^3 grid → surface · ground truth → VAE → visible surface · photo → VolFill → scene

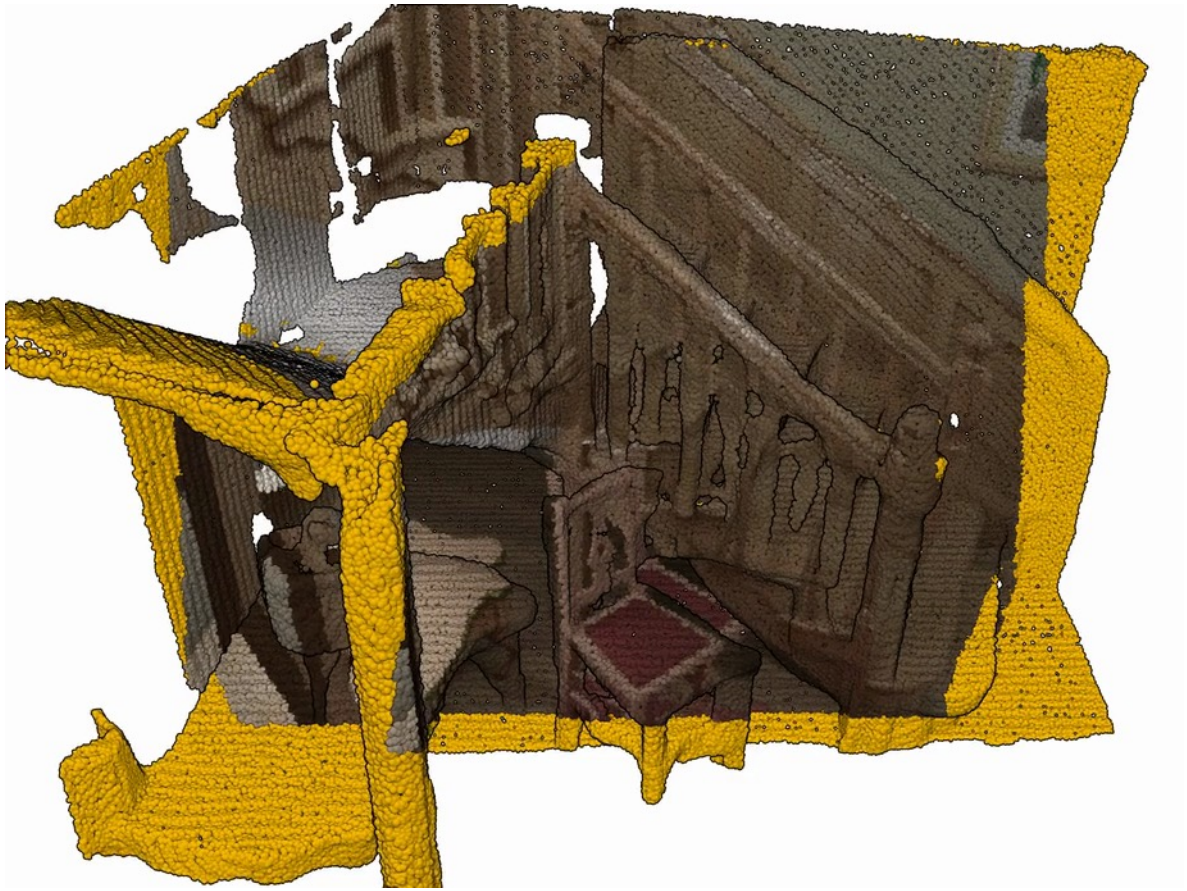
> Grid and VAE are near-lossless → improve the **generator and its grounding**.

Different seeds change the hidden parts, not the visible ones



5 seeds on SCRREAM: complete CD **3.11 ± 0.08** · hidden regions vary **1.3×** more than visible

Limitations: small holes, indoor-only training, and 1.4 s sampling



Small holes, softer visible surfaces

the visible part is re-generated, not copied

Indoor scenes only

outdoor and large scenes need more data and grid chunking

1.4 s per image

50 sampling steps; slower than one regression pass

Next: more data and resolution, more views, and toward 4D

01

Scale

more diverse data (outdoor) and finer grids; the VAE already runs at 512^3 zero-shot (IoU 0.87)

02

More views

the same formulation, from one photo to a few photos or a video

03

Toward 4D

unseen geometry (VolFill) + geometry from video (DAGE) + motion (DELTA)

Three takeaways



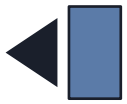
Generate the hidden part

ambiguity calls for sampling, not averaging



Use a volume

a truncated unsigned distance grid, compressed by a hybrid VAE



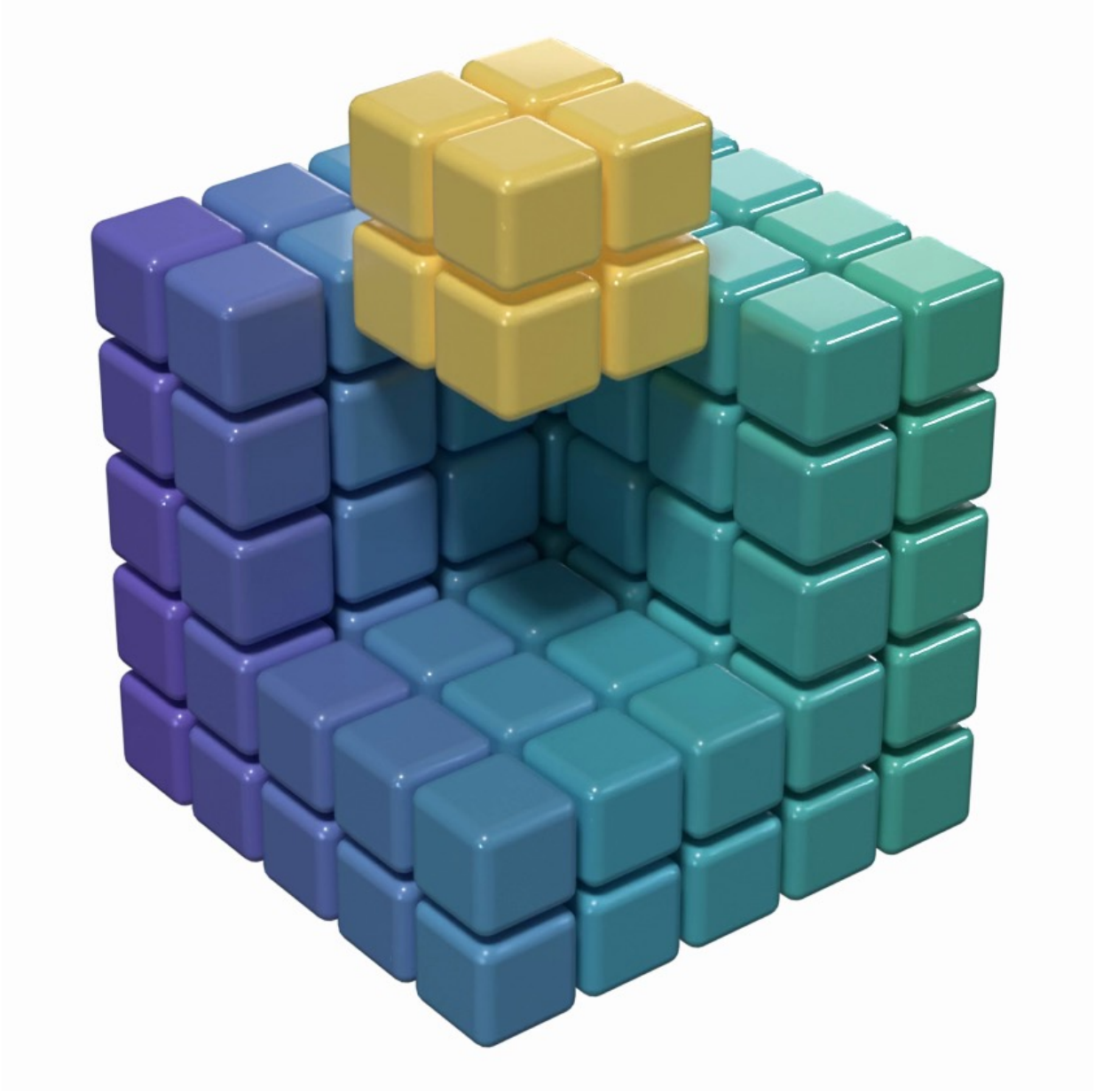
Inpaint, don't invent

image tokens + the visible latent, added to the noise

Thank you

Questions?

Project	**ngoductuanlhp.github.io/VolFill**
Code	**github.com/ngoductuanlhp/VolFill**
Paper	**arXiv 2605.31466**



Backup

Table 1 · SCRREAM (460 real images; $\times 10^2$)

	Visible			Occluded			Complete		
Method	**CD**	**APD0.1**	**APD0.02**	**CD**	**APD0.1**	**APD0.02**	**CD**	**FS0.1**	**FS0.02**
TripoSG†	13.75	44.74	10.08	10.73	55.34	13.37	11.10	53.56	13.30
TRELLIS†	11.63	54.05	15.00	9.65	60.70	17.52	10.72	56.69	15.59
DUSt3R	4.46	92.13	30.81	23.21	38.43	8.40	10.85	69.90	24.17
VGGT	3.04	96.98	45.62	17.40	44.60	10.04	6.77	80.45	36.77
DA3	2.37	98.98	54.34	16.16	48.31	10.76	6.16	82.86	43.10
DepthPro	3.85	95.10	33.30	15.31	48.66	9.71	6.90	80.36	28.44
MoGe2	**2.28**	**99.38**	**57.42**	15.29	50.09	10.30	5.74	83.96	45.84
LaRI	3.43	95.48	41.61	5.23	85.25	29.31	5.19	85.43	30.85
NOVA3R	3.20	96.77	43.75	3.56	**94.86**	41.21	3.43	**95.26**	45.12
VolFill	2.84	96.10	56.73	**3.46**	92.70	**55.53**	**3.03**	95.03	**54.83**

† trained on object-centric data. Bold = best per column.

Table 2 · NRGB-D (137 images, 9 scenes; $\times 10^2$)

	Visible			Occluded			Complete		
Method	**CD**	**APD0.1**	**APD0.05**	**CD**	**APD0.1**	**APD0.05**	**CD**	**FS0.1**	**FS0.05**
DUSt3R	4.46	92.13	69.08	23.21	38.43	22.18	10.85	69.90	51.61
VGGT	3.25	96.51	81.63	21.38	42.15	26.35	9.44	73.85	58.91
DA3	2.69	**98.68**	89.24	21.65	43.34	28.15	8.88	74.87	63.52
DepthPro	4.38	92.56	67.41	22.19	41.69	24.64	10.03	71.30	51.60
MoGe2	**2.62**	98.46	**89.55**	21.12	43.31	28.43	9.31	75.35	61.93
LaRI	4.09	93.77	73.51	13.04	64.53	45.82	9.79	68.37	43.47
NOVA3R	5.05	89.63	64.44	8.31	80.50	58.55	8.19	73.99	49.92
VolFill	4.52	90.06	69.22	**6.97**	**81.41**	**62.48**	**6.44**	**84.35**	**63.92**

Table 3 · metric-scale geometry

Method	CD ↓	$FS_{0.02}$ ↑	FPD ↓
LaRI	11.43	11.47	–
NOVA3R	6.56	22.23	16.50
VolFill	**3.88**	**44.90**	**4.86**

Table 4 · VAE decoder ablation

Decoder	L1 ↓	IoU ↑	Time (s) ↓	Mem. (GB) ↓
Dense-only	0.49	93.2	0.63	9.2
Sparse-only	0.48	93	0.09	2.5
Hybrid (ours)	0.43	93.5	0.07	1.7
w/ GT occupancy	0.44	93.5	0.07	1.7
w/ non-learned mask	1.79	74	0.08	1.8

Table 5 · conditioning and foundation-model ablations (SCRREAM)

Configuration	CD ↓	$FS_{0.02}$ ↑
Visible latent only	6.80	31.72
MoGe2 tokens only	3.81	46.80
Both (concat)	3.66	48.94
Both (add, ours)	3.03	54.83

Image features	CD ↓	$FS_{0.02}$ ↑
DINOv2	5.46	35.98
VGGT	4.32	42.30
MoGe2	3.81	46.80

Metrics

- **CD**: Chamfer distance; one-way (ground truth → prediction) on the visible / occluded subsets, two-way on the complete scene
- **APD_γ**: % of ground-truth points within γ of the prediction (coverage / recall)
- **FS_γ**: F-score at threshold γ (precision and recall), complete scene
- **FPD**: Fréchet point distance between predicted and real scenes in a pretrained 3D feature space (Uni3D)
- Visible vs occluded split from the ground-truth visibility mask; predictions aligned with Sim(3) except in Table 3

Data, training and compute

- **Data**: 3D-FRONT 18k rooms → 96k image–TUDF pairs; ScanNet++ fused multi-view depth → 46k pairs
- **Grid**: 256^3, $\tau = 3$ voxels, frustum-aligned box from the visible geometry (robust percentiles + 10% padding)
- **VAE**: sparse encoder, dense $16^3 \to 64^3$ + occupancy head + sparse $64^3 \to 256^3$ decoder; L1 + BCE + Dice + KL
- **DiT**: 12 blocks, width 768, 12 heads, AdaLN, QK-norm; rectified flow, logit-normal t; CFG drop 0.1; 50 Euler steps, guidance 3.0
- **Compute**: 2 GPUs (A6000 / L40S); VAE 3 days, DiT 2.5 days; ≈1,680 GPU hours in total incl. ablations

Q&A · Why are visible surfaces less sharp than MoGe2's?

- The scene is generated jointly; the visible part enters as **soft** guidance, so it is re-synthesised, not copied
- Not the representation: GT through the VAE keeps 99.8% of the visible surface ($FS_{0.02}$)
- **Training-free fix**: overwrite the near-surface band with MoGe2's visible TUDF → visible CD 2.85 → 2.31 (MoGe2 2.28), occluded CD 3.53 → 3.21

Numbers from the rebuttal re-run (SCRREAM, $\times 10^2$); paper Table 1: visible CD 2.84, occluded CD 3.46.

Q&A · Is 256^3 too coarse? How was NRGB-D curated?

- **Resolution**: GT → 256^3 TUDF → surface reproduces SCRREAM with $FS_{0.02} = 1.00$; 512^3 adds nothing there
- The VAE runs zero-shot at 512^3 (surface IoU 0.87, 7.5 GB): finer grids matter for larger, outdoor scenes
- **NRGB-D curation**: keep frames whose occluded-to-visible voxel ratio ≥ 0.2 → 137 samples from 9 scenes; script and list released
- **Seeds**: 5 seeds, complete CD 3.11 ± 0.08, $FS_{0.02}$ 54.2 ± 0.5