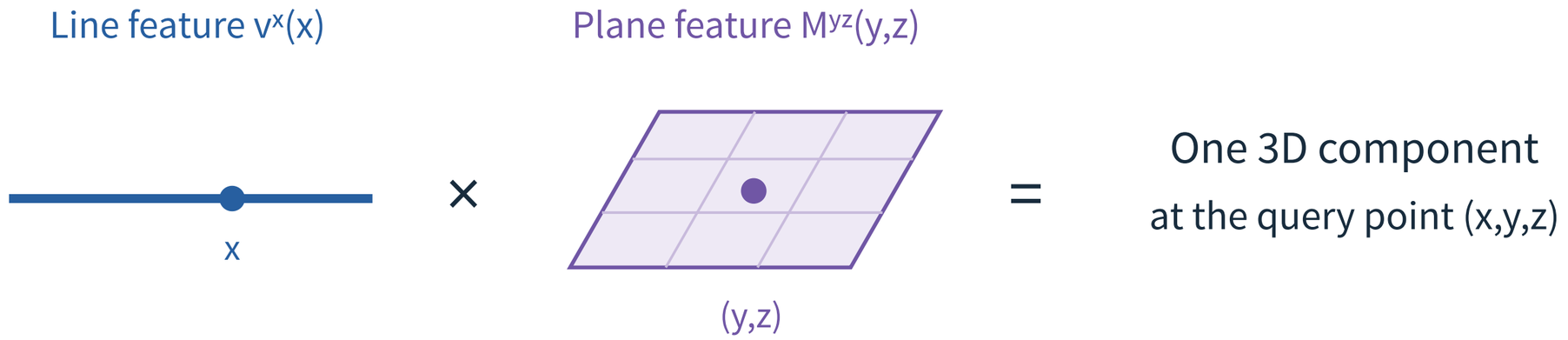
A 3D feature field built from lines and planes



$$T(x,y,z) = \sum_r [v_r^x(x)M_r^{yz}(y,z) + v_r^y(y)M_r^{xz}(x,z) + v_r^z(z)M_r^{xy}(x,y)]$$

T: feature grid · r: component index. Sum components in all three orientations.

Density and appearance come from compact factors. FourieRF controls the detail stored in each factor.

FourieRF: bandwidth and optimization

$$b_t = b_0 + t\Delta, \quad \Delta = (1 - b_0) / N$$

Conceptual linear schedule in the main method. N is its duration.

What changes

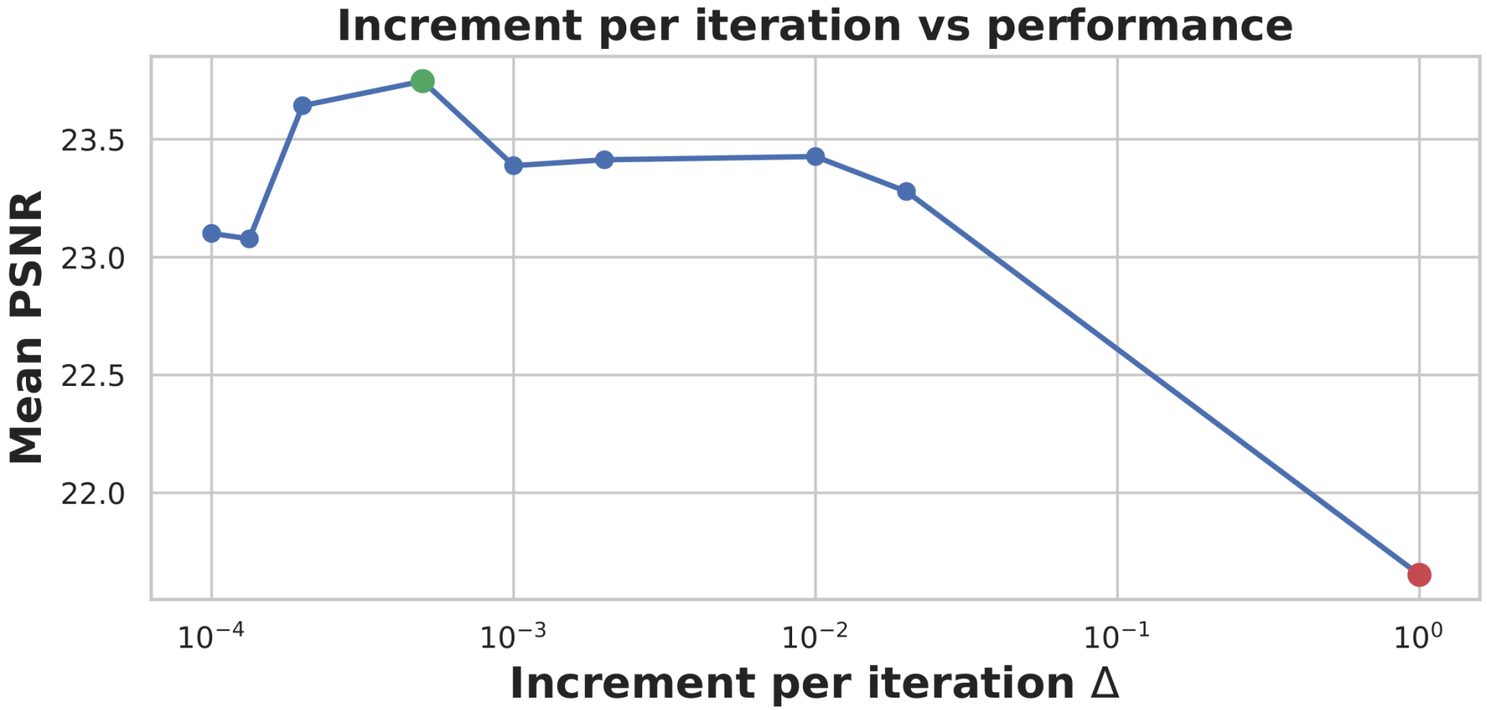
- Filter every 1D and 2D tensor factor before field evaluation.
- Use the prescribed mask at iteration t while differentiating the image loss.
- Remove positional encodings from feature and view-direction decoder inputs.

What optimizes the field

- Training-image supervision and total variation on density and appearance features.
- Synthetic scenes: AdamW weight decay 0.2.
- Real scenes: density-feature L1 penalty weighted 10^{-4}, with zero weight decay.

The source has conflicting initial-bandwidth values and an inconsistent synthetic increment. Use the released implementation for exact reproduction.

The rate of detail release matters



Blender, six input views. The frequency-increment axis is logarithmic.

Rapid release weakens regularization. Very slow release can underfit the fixed budget.

FourieRF: protocol and scope

Evaluation	Input views	FourieRF PSNR
Blender: 8 scenes	4 / 6	21.728 / 23.927 dB
LLFF: 8 scenes	3 / 6 / 9	19.303 / 23.595 / 25.011 dB

- Blender follows ZeroRF’s evaluation setup; LLFF follows RegNeRF’s.
- 10,000 iterations; VM tensor factors in the main comparison.
- No claim of recovering truly unobserved surfaces.

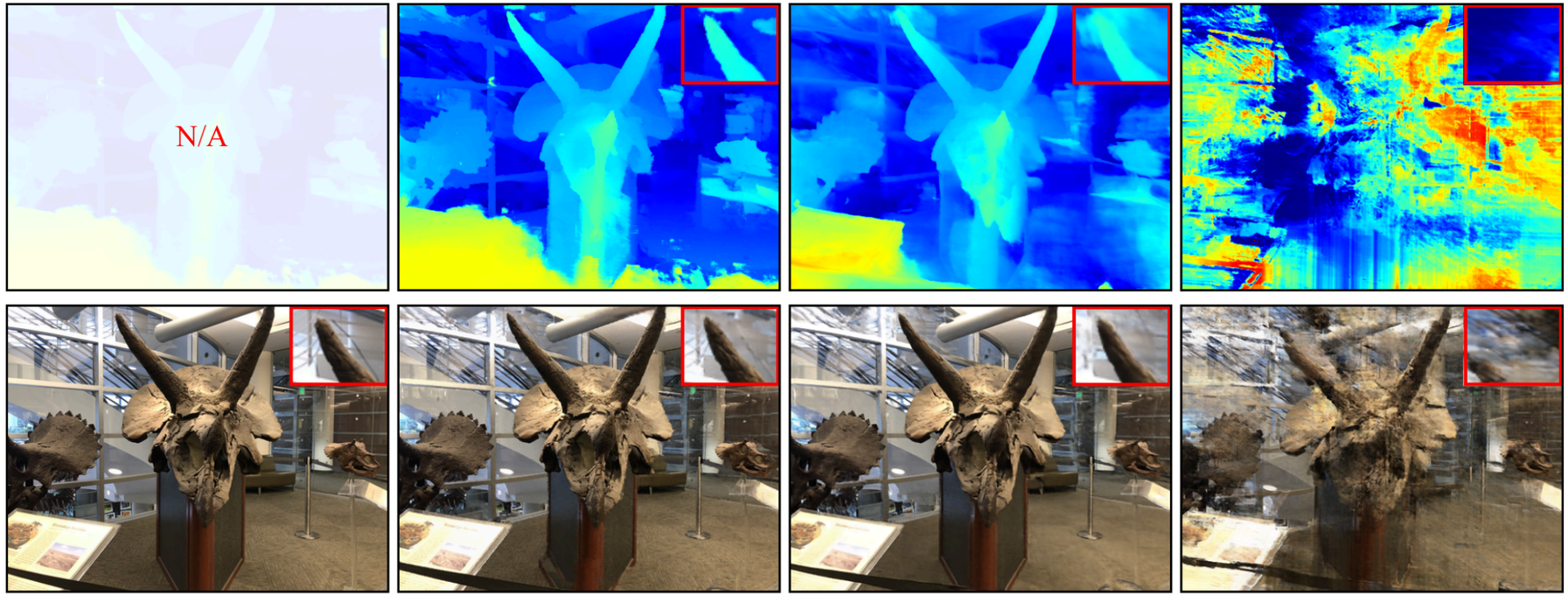
FourieRF: interpreting the timing ratio

Method	Relative training time ↓	4-view Blender PSNR ↑
TensoRF	1.000	—
FourieRF-VM	**0.930**	**21.728 dB**
ZeroRF	5.181	21.940 dB
FreeNeRF	35.710	—

5.57× training speedup over ZeroRF in this setup, with 0.212 dB lower PSNR.

10,000-iteration Blender timing comparison; TensoRF normalized to 1. RTX 4090.

FourieRF: real-scene comparison



LLFF example: ground truth, FourieRF, FreeNeRF, ZeroRF. The reference has no ground-truth depth.

Competitive reconstruction does not imply the best PSNR or verified depth.

MILo: losses and geometric evaluation

Ablation	Tanks & Temples F1 ↑
Baseline	0.41
+ Mesh depth	0.46
+ Mesh normals	0.44
+ Anti-erosion	0.44
+ Interior regularization	0.47
GOF integration	0.49

Normal supervision improves selected surfaces visually even when rounded F1 decreases.

MILo: optimization and evaluation boundaries

Iterations	Training stage
0–3,000	Photometric fitting and aggressive densification
3,000–8,000	Normal regularization; Gaussian refinement
8,000–18,000	Mesh supervision and selected pivots

Delaunay update: 500 iterations. Occupancy-label update: 200 iterations.

Mesh-rendering quality complements direct geometry metrics; it does not certify unseen surfaces.

MILo: preserving the surface and cleaning its interior

Preserve selected Gaussian centers

$$\mathcal{L}_{\text{erosion}} = \sum_{g \in G_{\text{Del}}} \max(0, f_{\mu g})$$

G_{Del}: Gaussians selected for triangulation

$f_{\mu g}$: signed value at a Gaussian center

Penalize positive center values. Keep a negative sample inside thin structures.

Suppress unsupported interior surfaces

$$\mathcal{L}_{\text{interior}} = \sum_p o_p \, H(\sigma(-f_p), o_p)$$

$o_p = 1$ when pivot p lies behind all visible mesh depths

H: binary cross-entropy · σ: sigmoid

Supervise only interior-labeled pivots. Refresh labels every 200 iterations.

$$\mathcal{L} = \mathcal{L}_{\text{vol}} + \lambda_{\text{MD}}\mathcal{L}_{\text{MD}} + \lambda_{\text{MN}}\mathcal{L}_{\text{MN}} + \lambda_{\text{erosion}}\mathcal{L}_{\text{erosion}} + \lambda_{\text{interior}}\mathcal{L}_{\text{interior}}$$

$\lambda_{\text{MD}} = \lambda_{\text{MN}} = 0.05 \cdot \lambda_{\text{erosion}} = \lambda_{\text{interior}} = 0.005$ in the reported implementation.

A zero crossing gives a differentiable surface vertex

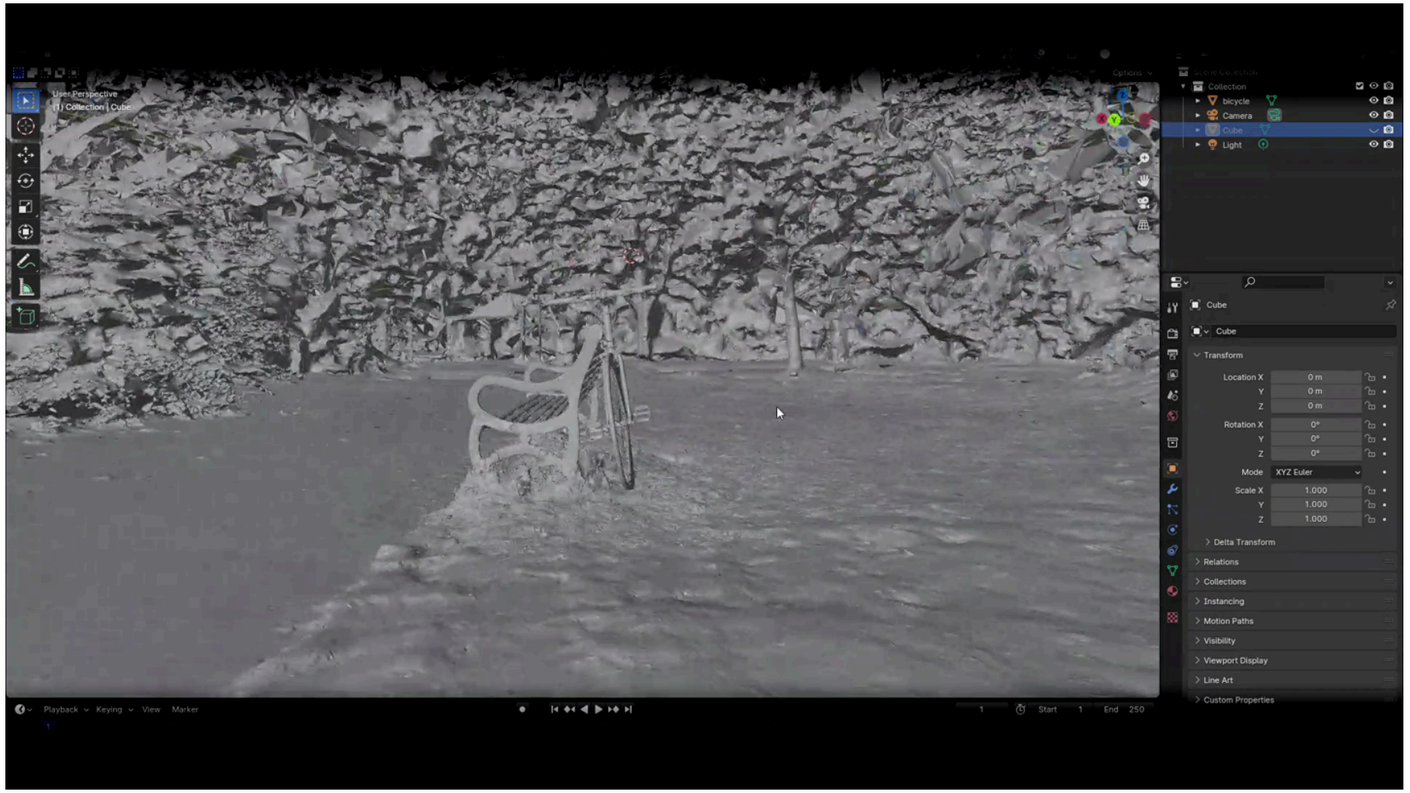
$$v = (f_a p_\beta - f_\beta p_a) / (f_a - f_\beta)$$

Moving either pivot or changing its signed value moves the extracted surface.

Mesh losses → **Surface vertices v** → **Signed values f**
Gaussian geometry μ, R, s

Arrows show the gradient path. Discrete Delaunay connectivity updates every 500 iterations.

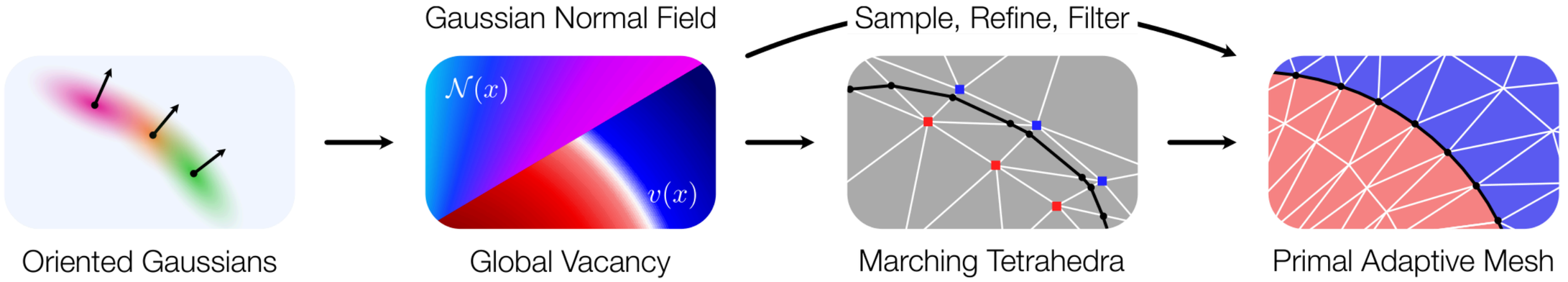
Inspecting recovered thin structures



Original supplementary video: exported bicycle and bench mesh in Blender.

Changing the viewpoint exposes details that a single attractive rendering can conceal.

Spokes: oriented fields and surface extraction



Two pivots per Gaussian
Use the center and an outward-displaced point for tetrahedral surface extraction.

Primal Adaptive Meshing (PAM)
Resample and project vertices to control mesh resolution independently of Gaussian count.

Tanks & Temples · mean F1 ↑ · six scenes

Method	Uniform sampling	Virtual scan
MILo	0.34	0.42
Spokes (2 pivots)	**0.48**	**0.53**

Both methods evaluated with the Spokes paper's protocols.

Vacancy and normal fields are estimated in practice. The table reports the two-pivot variant; PAM is optional.

MILo: fewer vertices and smaller meshes

Method	Mesh vertices ↓	Mesh size ↓	Training time ↓
GOF	16.49 M	600.0 MB	93 min
RaDe-GS	14.75 M	592.0 MB	42 min
MILo base	**4.36 M**	**179.6 MB**	**50 min**
MILo dense	6.89 M	276.1 MB	110 min

Tanks & Temples resource table; RTX 4090. Base/dense use the RaDe-GS backbone. Legacy vertex-based F1: MILo 0.47 (base) and 0.49 (dense); GOF 0.46, RaDe-GS 0.40.

Mesh size and geometric accuracy are separate claims. Compare accuracy with a common surface-sampling protocol.

ZeroKey: localization thresholds and comparison scope

Method	0.01	0.05	0.10
B2-3D · few-shot, 62 views	20.29	57.72	70.57
B2-3D · few-shot, 26 views	14.39	44.11	56.90
GPT-4o localizer	0.48	6.04	20.73
CLIP-DINOiser	1.41	9.80	25.56
StablePoints	5.80	19.91	38.22
Binary back-projection	5.84	29.81	54.64
ZeroKey · 26 views	**13.16**	**56.60**	**79.43**

KeypointNet: airplane, chair and table. IoU (%) at geodesic thresholds.
StablePoints receives the ground-truth keypoint count; benchmark text descriptions are manually assigned.

ZeroKey: distinguish dense agreement from isolated votes

Ordinary distance treats an isolated pair like a well-supported cluster.

$$d_m(a,b) = \max\{\text{core}_k(a), \text{core}_k(b), \|a - b\|\}$$

$\text{core}_k(a)$

Distance from candidate a to its kth nearest neighbor.

Isolated candidates have large core distances, even if two happen to be close.

HDBSCAN on pooled 3D candidates

- Use the Gaussian soft-voting weights.
- Find stable dense clusters across views.
- Fix minPts k = 10 in the experiments.

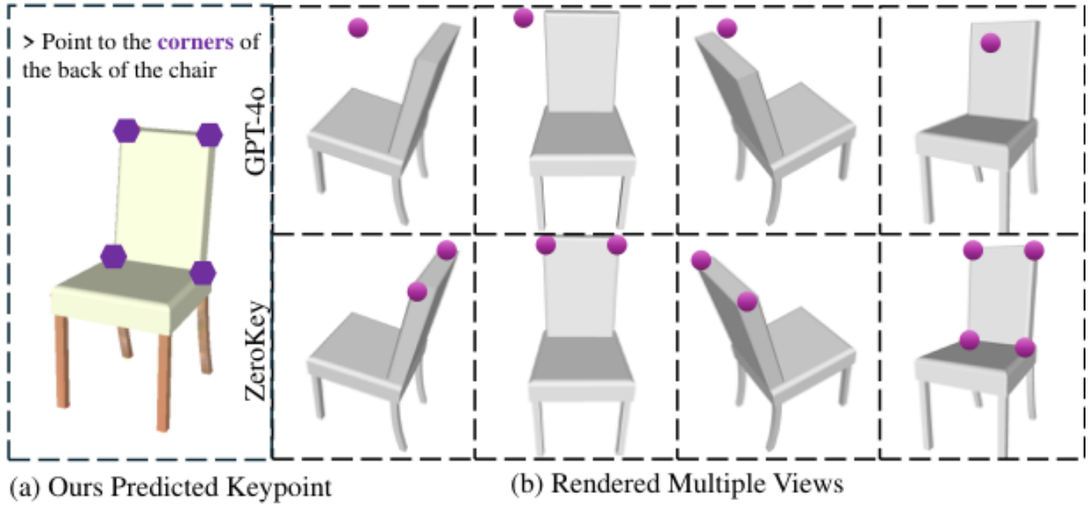
Spatial density separates repeated semantic instances from inconsistent predictions.

A named point is a localization problem

“Point to the corners of the chair back.”

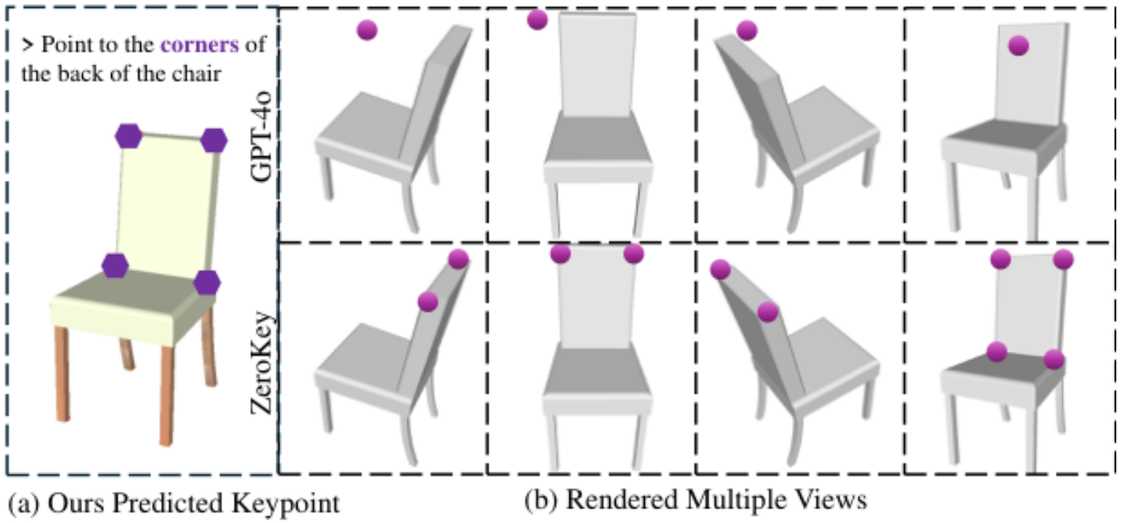
- Input: a known mesh and a text query
- Output: named locations on its surface
- No task-specific 3D keypoint training

Semantic recognition still needs precise spatial grounding.



The same chair query in multiple rendered views

Use known depth to lift an uncertain image point



A small image error can cross a silhouette or a depth discontinuity.

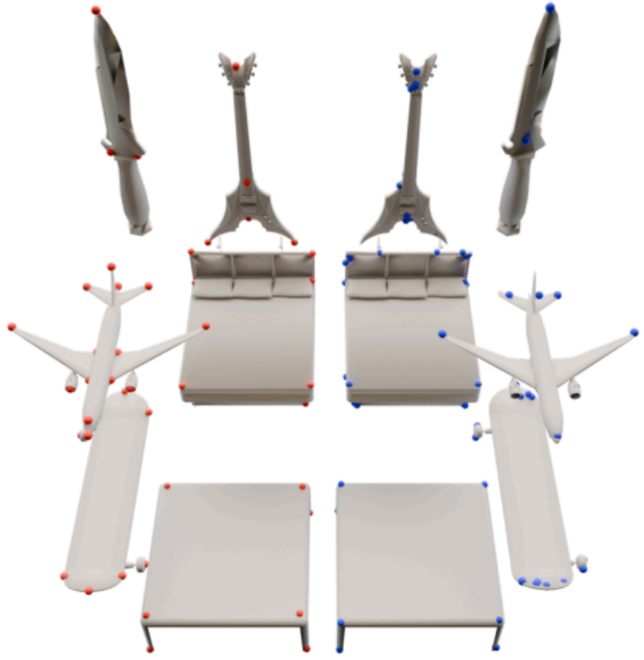
Cache the mesh depth while rendering.
Each valid pixel maps to a visible 3D surface point.

$$x_j(u) = R_j^T[D_j(u)\, K_j^{-1}\tilde{u} - t_j]$$

u: pixel; ũ: homogeneous pixel
D: camera-depth map; K, R, t: camera parameters

Lift an h × h neighborhood, reject background pixels, then average its valid 3D points.

Sparse semantic handles leave gaps



Selected named keypoints across different shapes

What represents the surface between the points?

- Repeated model calls make dense querying costly.
- Ambiguity and symmetry remain.
- Editing and segmentation need local features across the shape.

The next step is to learn language-aligned features directly in 3D.

PatchAlign3D: where the two training signals come from

Pseudo-part annotations

- 32,052 Objaverse shapes: 28,827 train / 3,225 validation.
- 10 views per shape → SAM masks → Gemini 1.5 part names.
- Back-project to 3D; retain overlapping labels.
- Over 2 million annotations, 761 categories.

Training and representation

- Stage 1: frozen DINOv2 visual targets.
- Stage 2: OpenCLIP ViT-bigG-14 text embeddings.
- 100 epochs per stage; batch size 32.
- Learn τ and b, initialized at 0.1 and −10.

Random rotations, translations, scaling and point jitter support generalization.

PatchAlign3D: build a stable visual target before training

Rendered feature maps

Bicubic upsampling to image resolution

→

Visible surface points

Average observations across views

→

Cached patch targets

Average features within each patch

$$d(x) = 1/|V(x)| \sum_{r \in V(x)} F_r(\pi_r(x))$$

$$d_i = 1/k \sum_{x \in P_i} d(x)$$

$V(x)$: views observing x; F_r: DINOv2 feature map; π_r: camera projection; k: points per patch.

$$L_{2D} = 1/G \sum_i [1 - \cos(h_{2D}(z_i), d_i)]$$

z_i: patch token; h_{2D}: visual projection; G: number of patches.

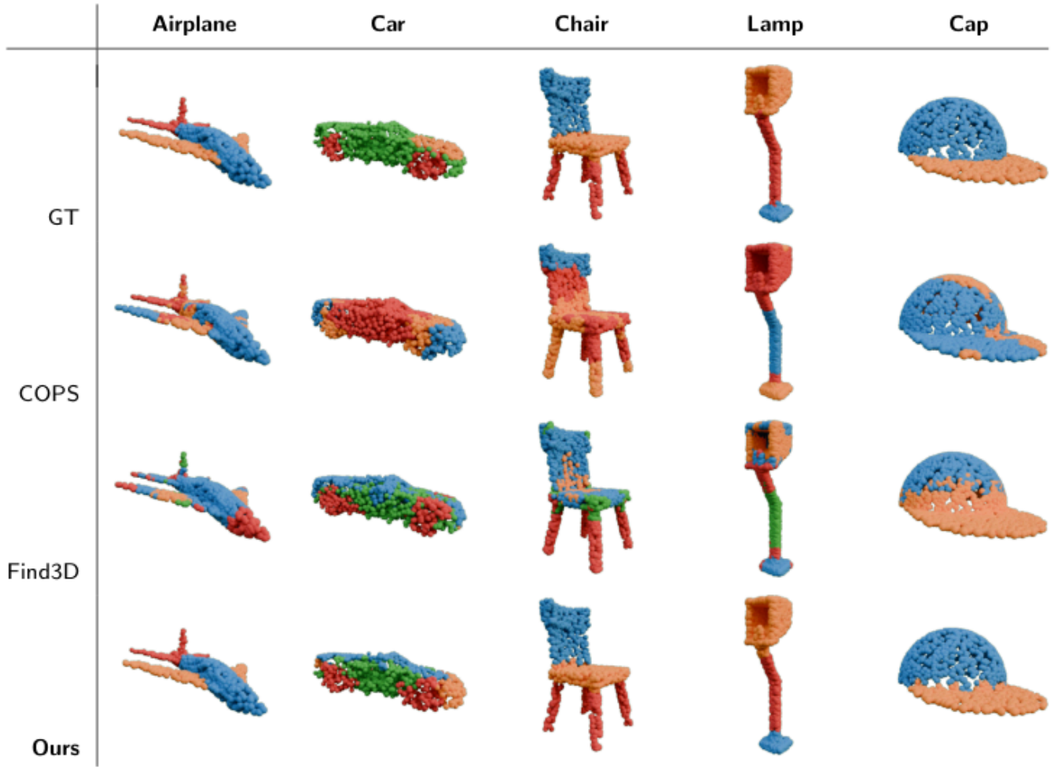
Match each online patch center to the nearest cached center before applying the cosine loss.

PatchAlign3D: benchmark scope and exceptions

Benchmark	PatchAlign3D mIoU	Comparator mIoU	Qualification
ShapeNetPart	56.9	COPS 25.6	Laptop: 61.1 vs COPS 63.3
FAUST	67.8	Find3D 63.2	Torso: 49.7 vs Find3D 52.0
PartNetE	41.4	COPS 27.0	PartSLIP reports 36.4 cIoU
ScanObjectNN	22.7	Find3D 18.8	Noise and clutter remain difficult
Objaverse unseen	35.61	Find3D 34.6	Author-defined 14-category split

Different datasets and metric definitions support different conclusions.

PatchAlign3D: selected segmentation examples



Original ShapeNetPart examples with all displayed baselines

Spatial coherence
Many predicted regions follow recognizable parts.

Boundary limitations
The car remains a useful counterexample.

Qualitative examples illustrate behavior. Aggregate tables determine the measured gain.

Stronger part segmentation at feed-forward speed

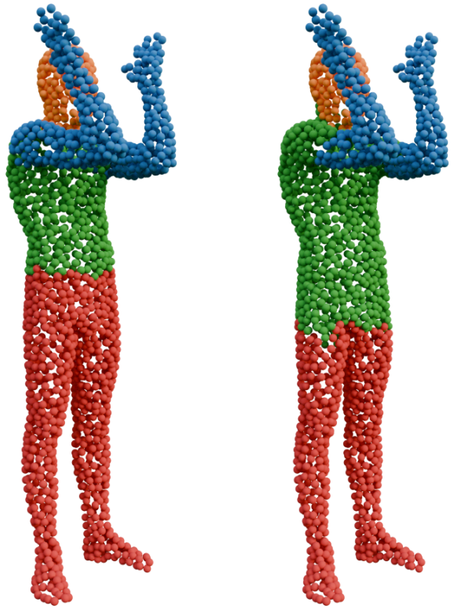
ShapeNetPart, point-cloud methods. Reported IoU (%) and inference time.

Method	mIoU	cIoU	Seconds / shape
COPS	25.6	32.2	1.38
Find3D	23.3	23.9	0.4
PatchAlign3D	**56.9**	**53.1**	**0.4**

+31.3 percentage points mIoU over COPS. The reported runtime matches Find3D.

mIoU averages instances. cIoU averages categories. Timings refer to the paper’s evaluation.

Transfer improves, but domain shift remains



Ground truth PatchAlign3D

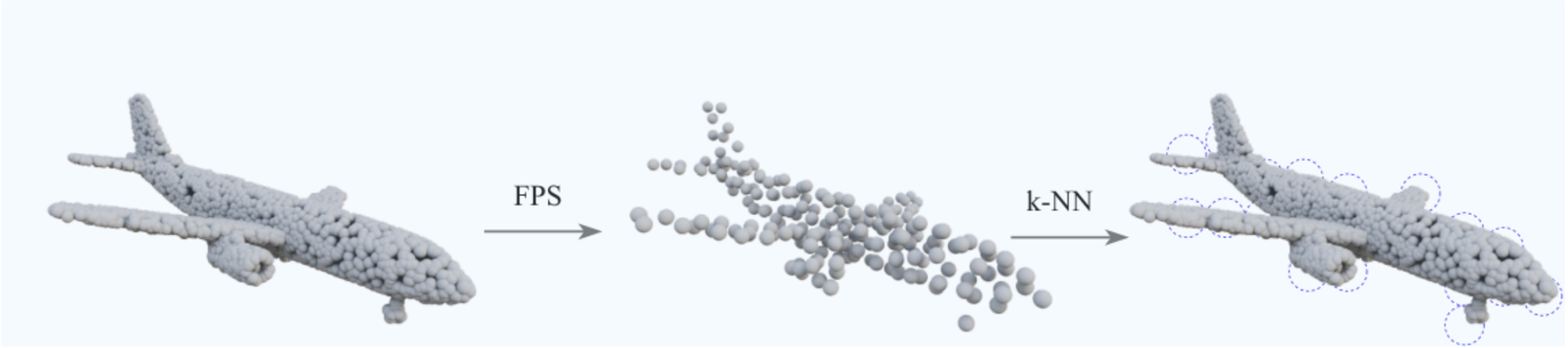
A FAUST pose, with visible boundary errors

FAUST	mIoU (%)
COPS	30.4
Find3D	63.2
PatchAlign3D	**67.8**

Real scanned objects are harder.
ScanObjectNN: 22.7 mIoU versus Find3D's 18.8.

Improvement does not remove the robustness gap.

PatchAlign3D: patch encoding and point-level inference



2,048 XYZ points → 128 FPS centers → 32 nearest neighbors per patch

Local encoding

PointNet(P_i) + MLP(c_i)

P_i: patch points; c_i: center

→

Context across patches

12 transformer layers

128 output patch tokens

→

Text projection

Compare each patch with part names

OpenCLIP ViT-bigG-14 text embeddings

$$\hat{y}(x) = \arg\max_j \langle z_{i(x)}, t_j \rangle, \quad i(x) = \arg\min_i \|x - c_i\|$$

z: text-aligned patch feature; t: part-name embedding. Each point inherits its nearest center's label.

What is trained, and what counts as supervision?

VLM-reward setting

- Source points are provided.
- DINOv3 is pretrained and frozen apart from LoRA.
- The VLM scores sampled target matches.
- Target annotations are used for evaluation, not the reward.

Separate diagnostic settings

- Ground-truth PCK reward is supervised.
- Geometry-aware fusion changes the feature stack.
- Cat-only and all-category training are different regimes.
- Peak validation is not held-out test performance.

Training details: stochastic updates and residual feature fusion

Policy update

$$\mathscr{L}_t = -(1/B)(r_t - b_t) \log \pi_\theta(q_t)$$

- Temperature $\tau = 0.1$; EMA rate 0.01.
- Accumulate $B = 10$ samples per Adam step.
- Clip gradient norm at 1.0.
- Skip invalid scores; retain valid zero scores.

Optional geometry-aware features

Cached SD v1.5 features: 640 + 1280 + 1280 channels.
Live DINO ViT-B features: 768 channels.

$$\hat{F} = \varphi(F_{DINO}) + a \cdot \mathrm{AggNet}([F_{SD}; F_{DINO}])$$

φ: DINO projection; a: learnable fusion gate.

Fuse at a shared 60 × 60 feature grid.
Train LoRA and the aggregation network.
The residual branch preserves the DINO signal.

Transfer is measurable, but not uniform

Test category	Frozen DINOv3	Geo-aware, 3 seeds	Δ PCK
Cat, trained category	73.27	77.74 ± 0.64	+4.47
Bird	80.45	83.04 ± 0.27	+2.59
Cow	72.06	73.31 ± 0.45	+1.25
Sheep	61.50	62.64 ± 0.42	+1.14
Dog	61.89	62.82 ± 0.09	+0.93
Horse	65.03	64.32 ± 0.17	−0.71
Person	65.62	64.50 ± 0.56	−1.12

SPair-71k test · cat-trained only · bbox PCK@0.10. Other-category mean gain: +0.68; plain LoRA: +0.02.

Alternative updates under the same judge budget

Method	PCK ↑
Frozen DINOv3	76.19
Top-K + LoRA	78.06 ± 0.04
DPO + LoRA	79.09
REINFORCE + LoRA	**78.94 ± 0.14**

SPair-71k cat validation · bbox PCK@0.10
Kimi reward · 3 seeds where uncertainty is shown.

+2.75 points
over the frozen backbone

The gain demonstrates useful feedback in this regime.

REINFORCE exceeds the listed Top-K setting; it does not establish superiority to DPO.

Semantic feedback still needs a geometry-aware representation

One SPair cat test pair · red: plain DINO pipeline · blue: geometry-aware variant · green: ground truth

Adding geometry-sensitive features can help resolve symmetry. This is an architectural comparison, not an isolated reward ablation.

Visual inference after reward training

Training

Visual proposal → VLM feedback
→ adapted visual features

Semantic supervision is expensive
but can be amortized.

Inference

Two images + source point
→ feature matching → target point

No VLM call.
26.3 FPS reported for the plain pipeline.

Reported timing: 512^2 input, RTX A6000. Geometry-aware timing uses cached SD features and is a different pipeline.

The trained adapter keeps inference visual; the VLM cost is paid during learning.

Uni3D similarity across all reported agents and settings

Agent	Single view	Four views	Active visual	Full 3D
Fable 5	.733	.764	.832	.884
Opus 5	.725	.748	.871	.927
GPT-5.6 Sol	.715	.745	.823	.879
GPT-6 Astra*	.794	.842	.912	.956
Kimi K3	.678	.713	.819	.848
Qwen 3.8 Max	.740	.768	.809	.857
Gemini 3.1 Pro	.728	.753	.691	.783
MiniMax M3	.495	.577	.482	.679

↑ Higher similarity. * Preliminary unpublished aggregate update; Astra run coverage and complete protocol parity remain unverified.

The harness changes the observation and interaction protocol

Setting	Target access	Target geometry	Refinement
Single view	1 fixed image	Hidden	3 rounds
Multi-view	4 fixed images	Hidden	3 rounds
Active Visual	Camera / viewport tools	Blocked by allowlist	Agent-directed
Full 3D	Unconstrained Blender tools	Available to query	Agent-directed

100 sampled objects from 3DCodeBench.
Original procedural programs are withheld.

Active Visual separates target and reconstruction processes.
Full 3D prohibits copying the target.

Evaluation conventions determine what a score can mean

Geometric alignment

Sample 8,192 surface points per mesh.
Center each cloud and scale to the unit sphere.

$$CD = \min_{R \in R_{24}} d_{sym}(RX, Y)$$

d_{sym}: mean squared nearest-neighbor distances in both directions.
R_{24}: the 24 right-handed axis-aligned rotations.

Translation, scale and these rotations are discounted.

Complementary checks

Appearance: four 512^2 renders; match views by the best one-to-one assignment.

$$e_{topo} = \Sigma_{i=0..2} |\beta_i^{GT} - \beta_i^{rec}| / \Sigma_{i=0..2} \beta_i^{GT}$$

Topology: median across objects.
Other quality metrics: mean across objects.

No loadable mesh: zero similarity; Chamfer penalty of 1.5 × the worst valid value in that run.

Ablations separate observation quality from evidence quantity

Agent / evidence setting	Chamfer ↓	Uni3D ↑
Gemini · four fixed views	.0165	.7609
Gemini · Active Visual	.0234	.6776
Gemini · normalized viewport	.0186	.7321
Gemini · Fable inspection views	.0191	.7605
MiniMax · Active Visual	.0798	.3722
MiniMax · normalized viewport	.0240	.6148

Ten-target ablation: better observations recover much of the lost performance, with metric-dependent outcomes.

The agent chooses both an observation and a program update

$$a_t \sim \pi_\theta(\cdot \mid H_t)$$ H_t: instructions, observations and previous actions

Acquire evidence

New view or geometric query

o_t: image; plus measurements in Full 3D

→

Revise the hypothesis

Blender program P_t

Parts, dimensions, transforms and materials

→

Execute and inspect

$\hat{G}_t = E(P_t)$

E: Blender execution; compare with the target

Active Visual

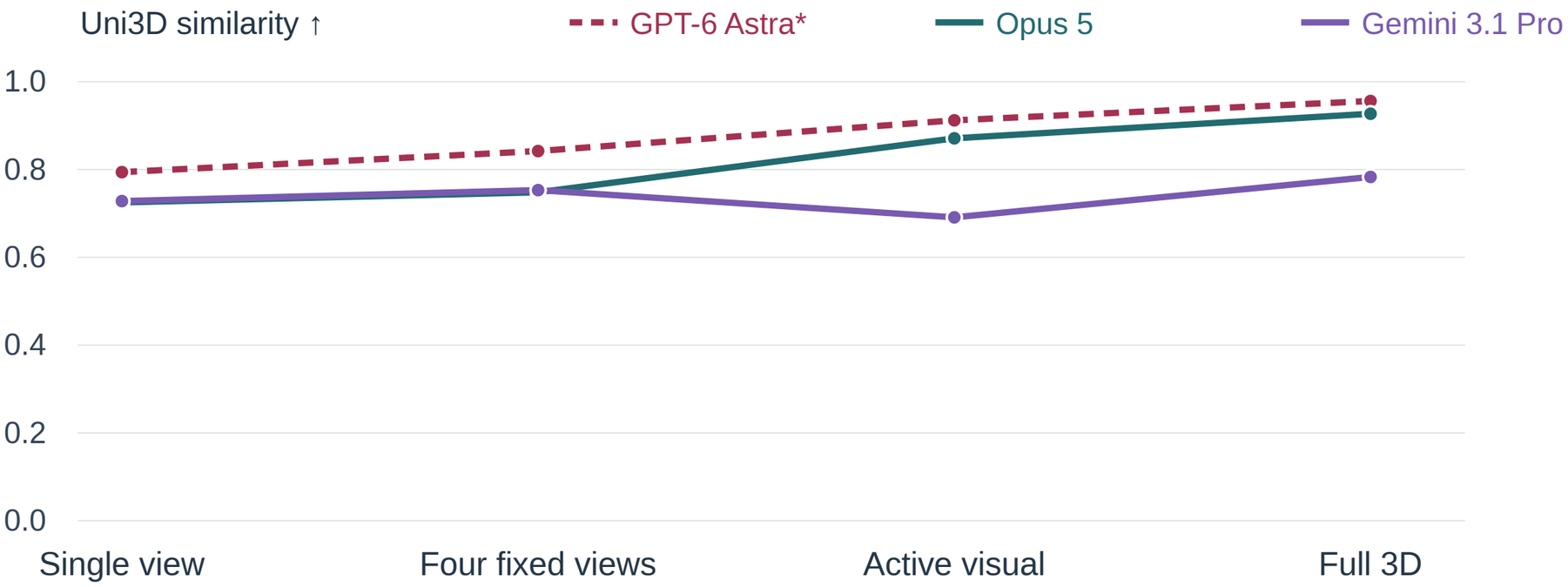
Separate target and reconstruction processes.
Target tools allow camera and viewport control.
Direct geometry access is blocked.

Full 3D Interaction

Query dimensions, transforms and distances.
The agent decides what to measure and when.
Copying the target is prohibited by instruction.

Evaluate the final executed object $\hat{G}_T$, plus the interaction needed to obtain it.

More access helps only when an agent can use it



* Astra: preliminary, unpublished aggregates; matched coverage and protocol not yet verified.

Selected agents · original seven-agent results plus preliminary Astra aggregates.

Active inspection can improve or degrade reconstruction. Full 3D access includes explicit geometric evidence.

Observation quality affects reconstruction

Gemini 3.1 Pro

Inspection count = 2

001 Active object · close | **002** Active object · distant | — No additional inspection | — No additional inspection | — No additional inspection

Two Gemini inspections on this target.

Opus 5

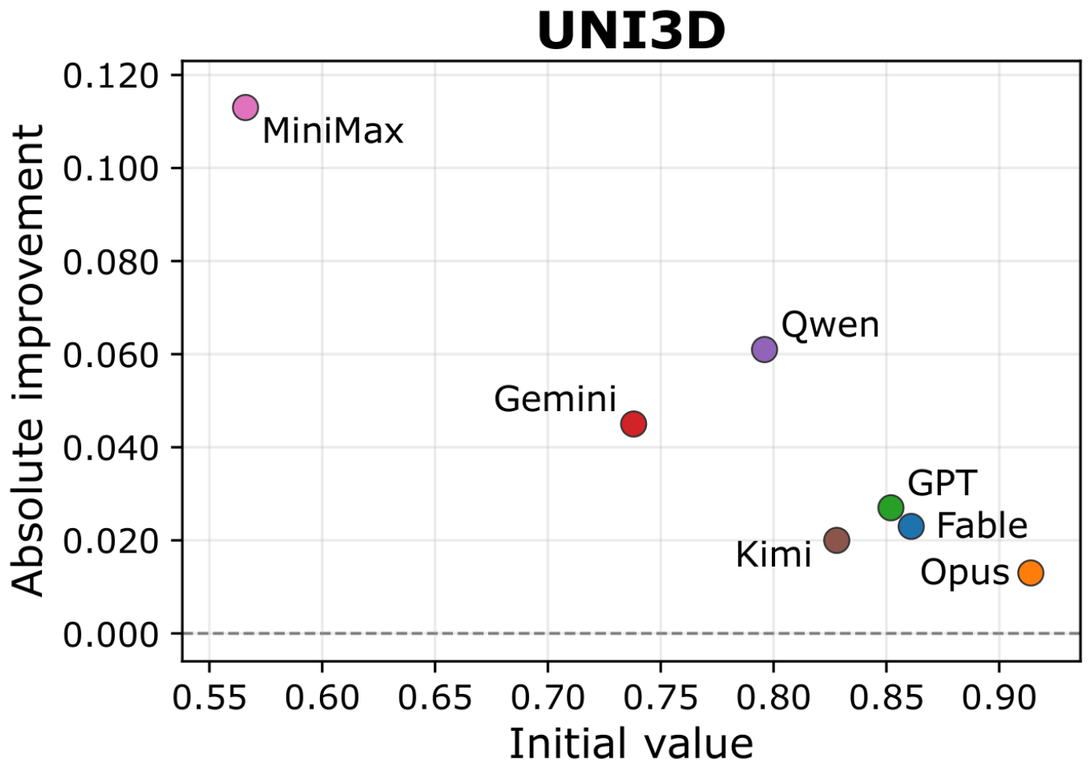
Inspection count = 22

001 Active object · close | **003** Solid shading · close | **006** Rendered shading · close-up | **013** Clipping | **022** Active object · distant

Selected views from 22 Opus inspections of the same target.

On 10 ablation targets, viewport normalization raises MiniMax Uni3D: 0.3722 → 0.6148.

A strong first reconstruction and a strong repair policy differ



Original seven-agent study · Full 3D interaction · no Astra repair data.

Measure both abilities

- Opus starts high and gains modestly.
- MiniMax improves substantially from a weak start.
- Qwen combines a stronger start with larger repair gains.

A larger gain does not imply a better final model.

Research ownership and collaboration

Project	Documented contribution
FourieRF / MILo / Spokes	Coauthorship, mentoring of Diego Gomez, MILo code contributions and LIX/GRAPHDECO collaboration
ZeroKey	Conceived, implemented and evaluated the first-author project
PatchAlign3D	Joint research with Souhail Hadgi and collaborators
Is It a Good Match?	Scientific lead since April 2026, experiments and manuscript
3DHarnessBench	Experiment guidance, project coordination, TaskSolver and restricted Blender interface

Teaching and mentoring

Experience

- Teaching assistant for eight HKU courses
- Certificate in Teaching and Learning in Higher Education
- Forthcoming MVA geometry teaching appointment
- Research mentoring across vision and graphics

Contribution at Télécom Paris

- Graphics, geometric learning and neural 3D representations
- Projects linking implementation to measurable evidence
- Teaching in English, with a commitment to French within two years

A staged research programme

Stage	Scientific focus	Development
First year	Controlled static/articulated tasks, verified trajectories, compact baselines	Initial student project and reproducible releases
Second year	Transfer to new operations and richer dynamics	Collaborative funding and industrial research links
Third year	Integrated reconstruction, animation and delivery evaluation	Broader partnerships and HDR preparation

Resources and extensions follow demonstrated progress on the core tasks.