

Télécom Paris

From Reconstruction to Reasoning

Structured 3D Representations

Bingchen Gong

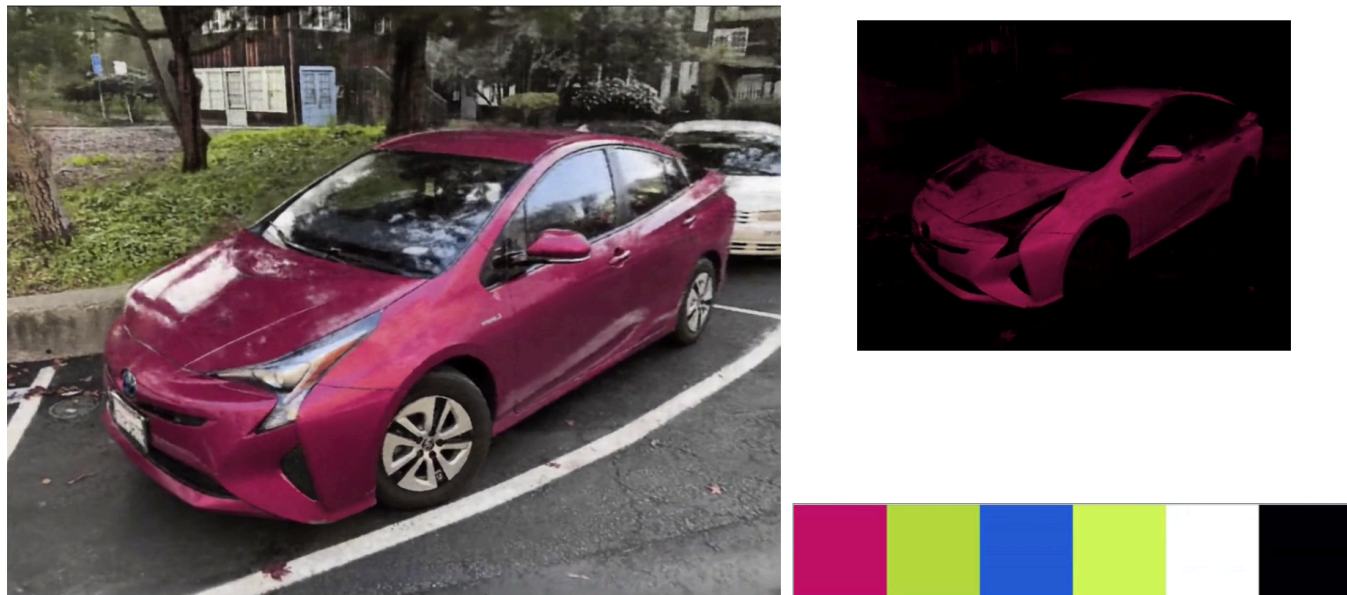
LIX, École Polytechnique

21 September 2026

Foundations for the research programme *From Video to Animatable and Efficient 3D Worlds*

Editing captured scenes

Color editing of the red car



RecolorNeRF: edit colors through a palette



SeamlessNeRF: compose parts and match appearance

Representations for useful visual intelligence

Different scenes can explain the same images,
yet behave differently when edited.

Which structure must be explicit to constrain reconstruction,
target an operation and verify its consequences?

Structure, meaning and feedback

01 What makes a reconstruction geometrically reliable?

Image agreement leaves ambiguity; control complexity and make surface constraints explicit.

FourieRF · MLo · Spokes

02 How precisely can meaning be grounded in geometry?

Connect concepts to points and parts; evaluate localization beyond recognition.

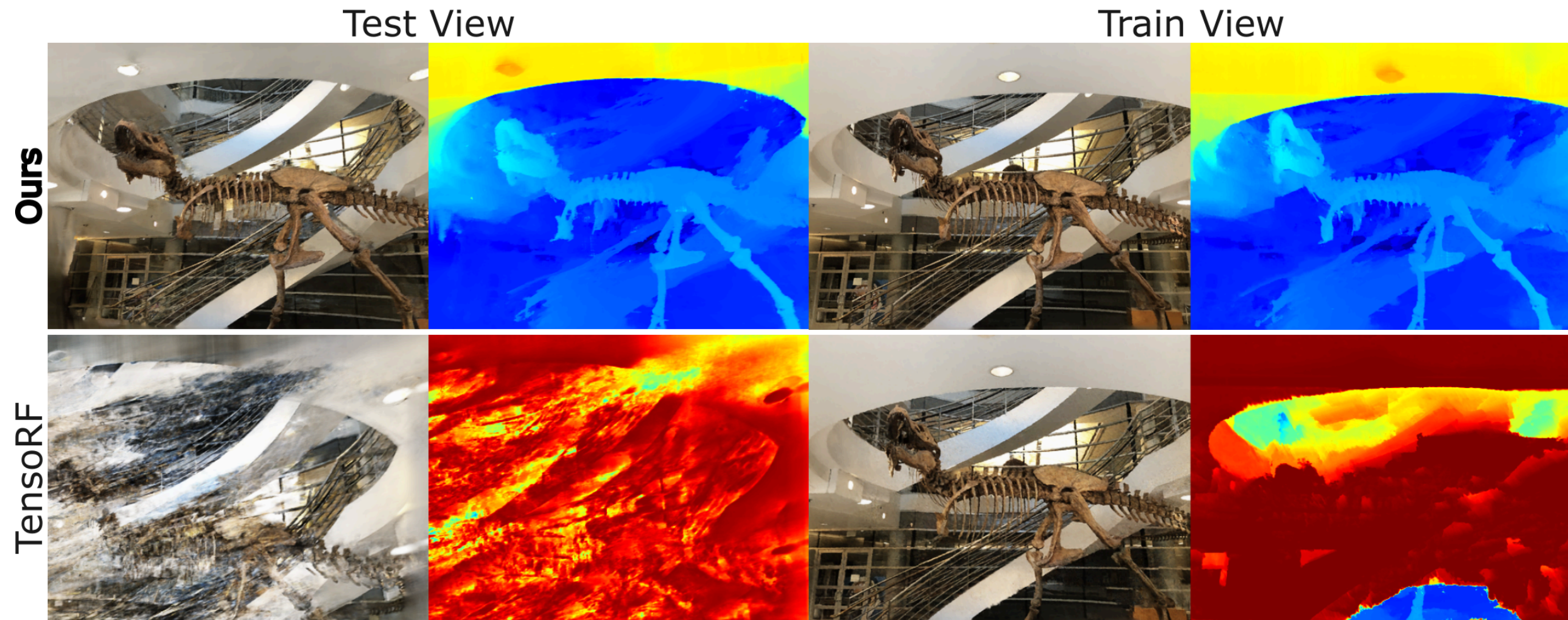
ZeroKey · PatchAlign3D

03 When does feedback improve behavior?

Judgments can train a predictor; richer evidence does not guarantee better construction.

Is It a Good Match? · 3DHarnessBench

FourieRF: controlling detail from sparse views

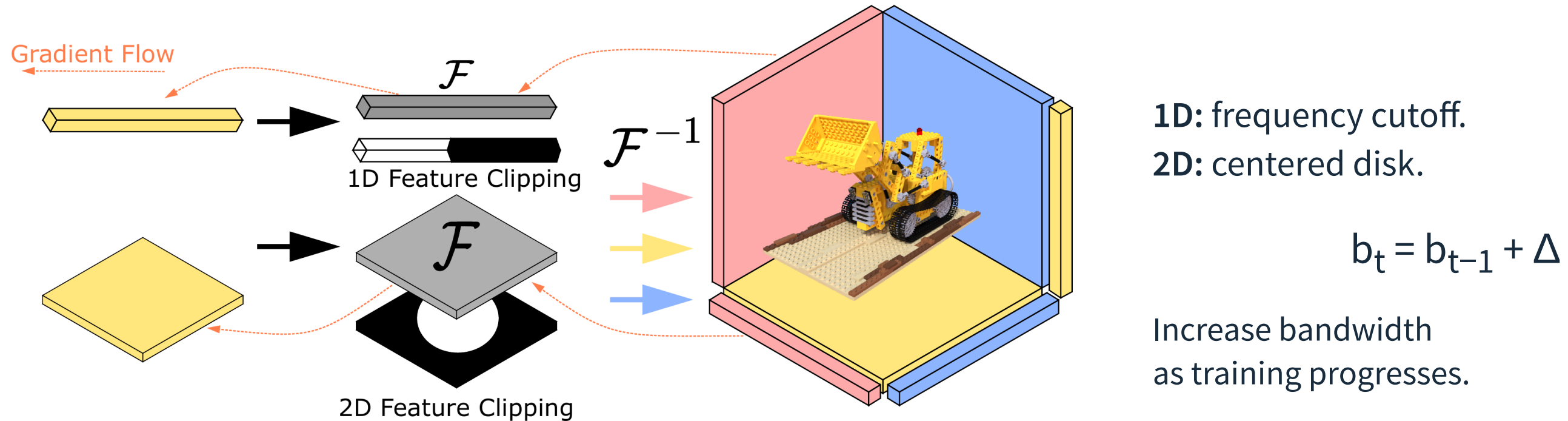


Held-out view on the left; training view on the right. Color and depth reveal different solutions.

Training views can fit well while unseen views fail.

Frequency control inside the representation

A compact 3D field sums learned line $\times$ plane factors.

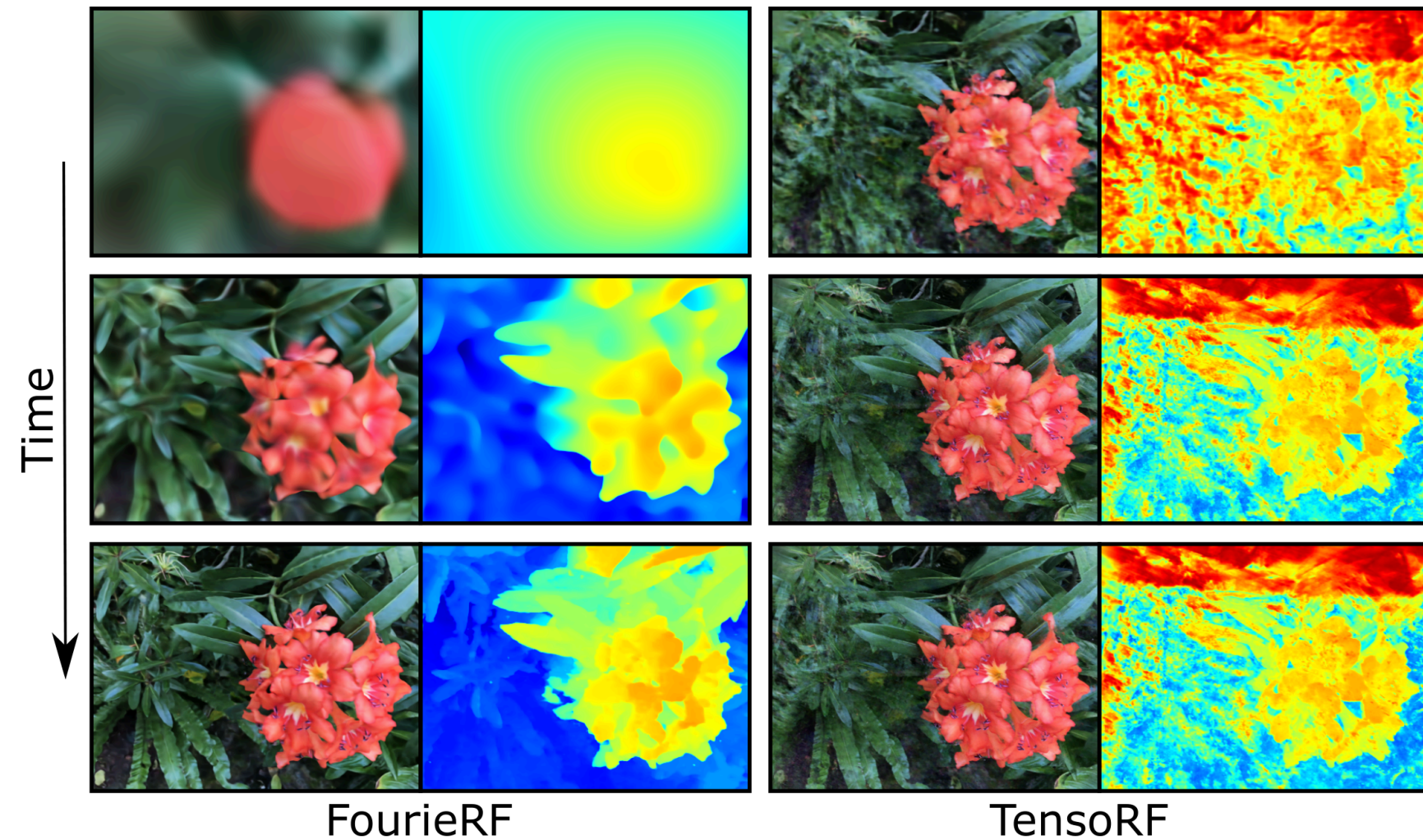


Original method illustration · FourierRF, Fig. 2

$$\tilde{f}_t = F^{-1}(M_{b_t} \odot F(f))$$

Render with the filtered factors; image error trains them through the mask.

Coarse structure before fine detail



Training proceeds downward. Left: FourieRF. Right: TensoRF.

Releasing detail gradually changes the geometry learned from the same images.

A measured speed–quality tradeoff

5.57× faster training than ZeroRF,
with 0.212 dB lower PSNR.

Method	Relative training time ↓	4-view Blender PSNR ↑
FourieRF-VM	0.930	21.728 dB
ZeroRF	5.181	21.940 dB

Reported Blender setup: 10,000 iterations, RTX 4090. Training time normalized to TensoRF = 1.

Frequency control regularizes the field. Geometric use also needs an explicit surface.

MILO and Spokes: surfaces from Gaussian splatting

How can Gaussians fitted to images describe a usable surface?

MILO • TOG / SIGGRAPH Asia 2025

From Blobs to Spokes • ECCV 2026

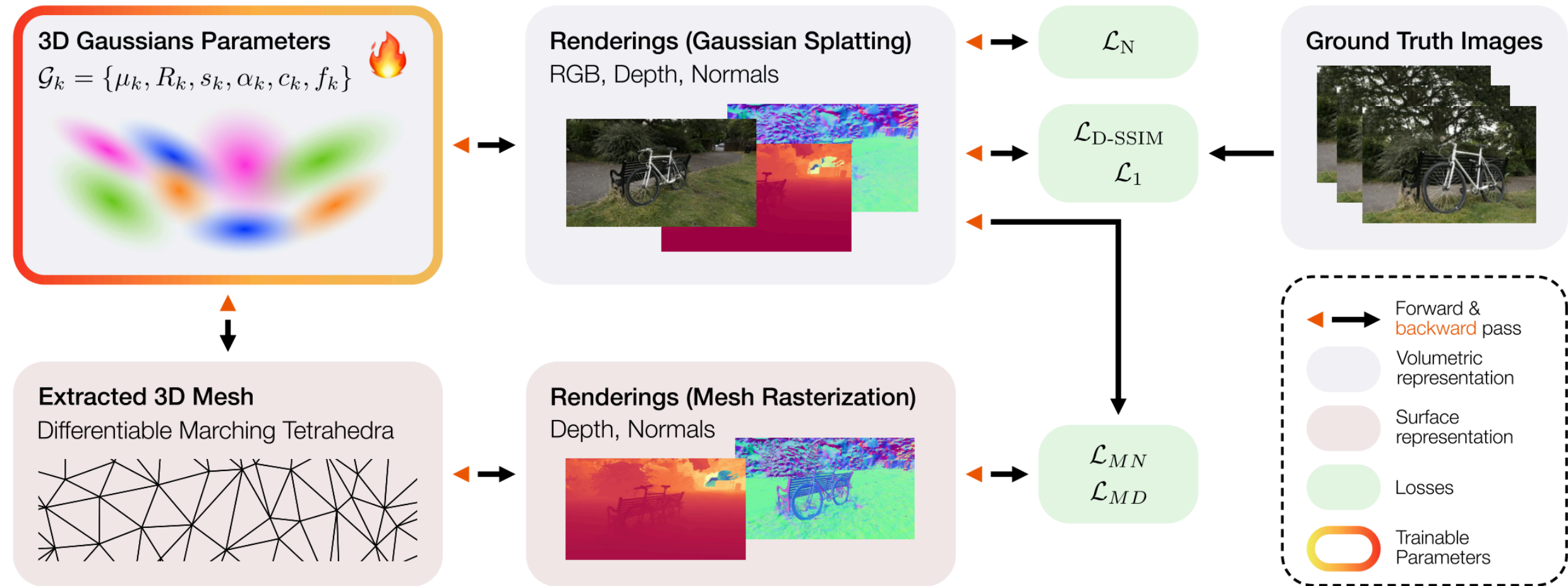


Let the extracted mesh
constrain training.



Give each Gaussian
an outward surface orientation.

MILo: the extracted mesh constrains training

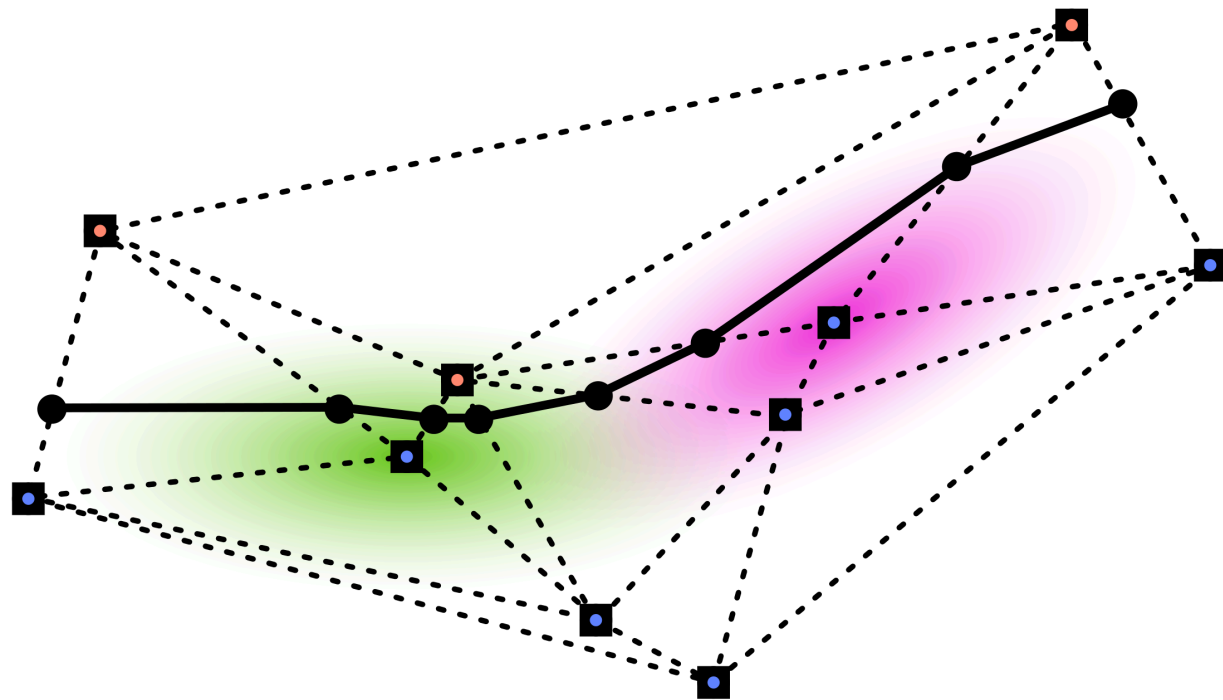


The same cameras render both representations. Mesh losses update the parameters that produce the surface.

$$\mathcal{L} = \mathcal{L}_{\text{vol}} + \mathcal{L}_{\text{mesh}} + \mathcal{L}_{\text{reg}}$$

Image fitting and Gaussian normals + mesh depth/normal agreement + surface regularization

MILo: surface vertices follow the Gaussians



Nine pivots in 3D: the Gaussian center and eight aligned box corners.

$$p_{k,i} = \mu_k + R_k(s_k \odot b_i)$$

Position μ , rotation R and scale s place each pivot p .
 b_i : unit-box center or corner.

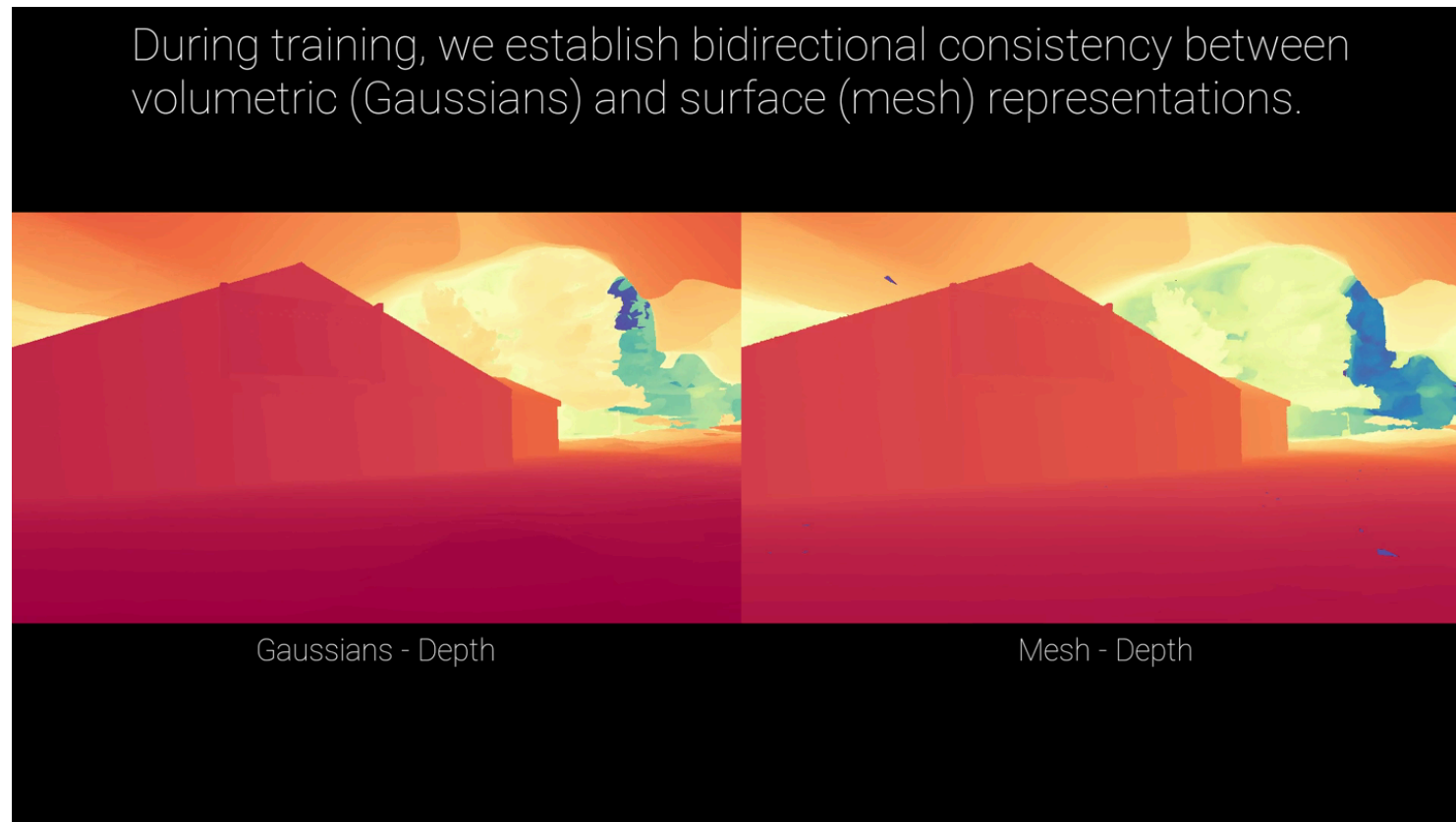
A learned scalar f marks each pivot as inside or outside.

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

Opposite signs locate the zero crossing: a mesh vertex v .

Mesh losses update both the signed values and the Gaussian geometry.

MILo: depth and normals align volume and surface



Gaussian and mesh renderings from the same camera

Depth locates the visible surface

$$\mathcal{L}_{MD} = \sum_i \log(1 + |D(i) - D_M(i)|)$$

D : Gaussian depth · D_M : mesh depth

Normals align its orientation

$$\mathcal{L}_{MN} = \sum_i [1 - \tilde{N}(i) \cdot N_M(i)]$$

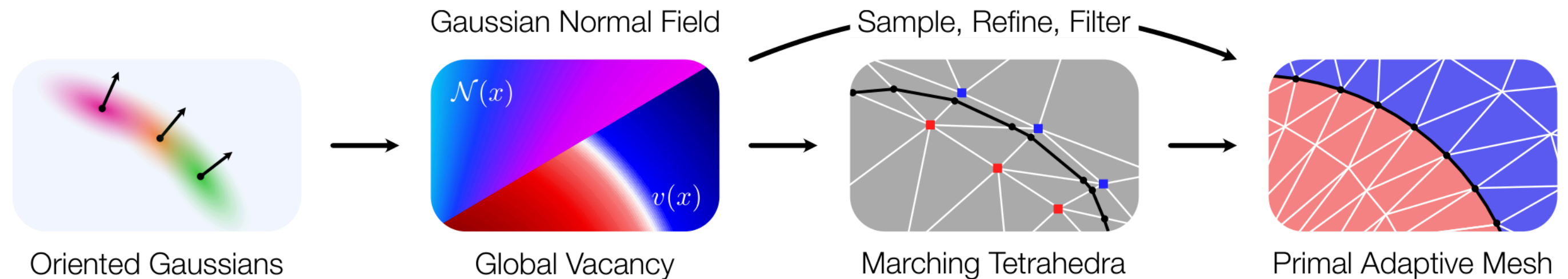
$\tilde{N}$: normal from Gaussian depth

N_M : rasterized mesh-face normal · i : pixel

$\mathcal{L}_{\text{mesh}} = \lambda_{MD} \mathcal{L}_{MD} + \lambda_{MN} \mathcal{L}_{MN}$. Both renderings should describe the same surface.

Spokes: oriented Gaussians define a surface boundary

Learn outward normals; encourage the Gaussians to wrap the surface.



Surface fields

Normal estimates describe orientation.

Vacancy estimates distinguish outside from inside.

Mesh extraction

Two pivots per Gaussian locate the boundary.

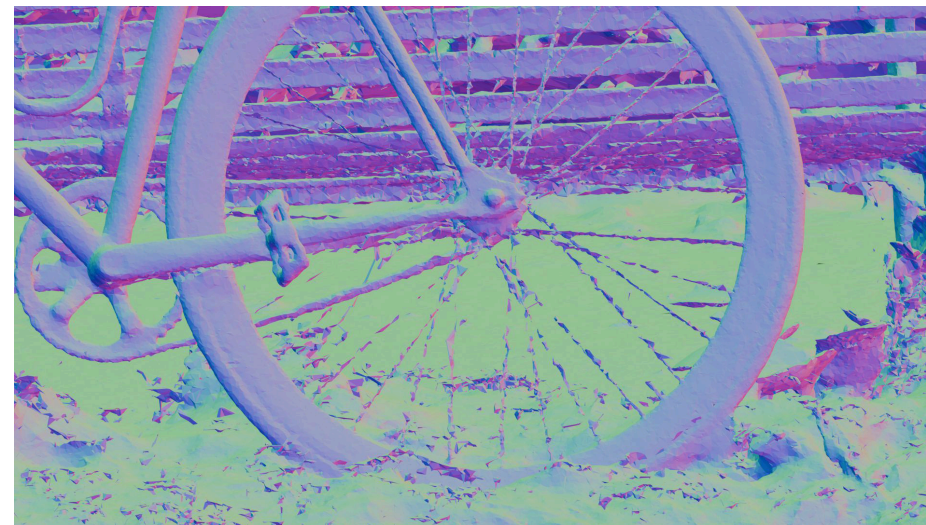
Optional PAM adapts mesh resolution.

MILo learns signed values on pivots. Spokes derives fields from oriented Gaussians.

Thin structures reveal the difference



Input photograph



MILo



Spokes / Gaussian Wrapping

Bicycle scene: input photograph and surface-normal visualizations.

Oriented Gaussian wrapping recovers more of the wheel's thin spokes in this example.

Mesh complexity and surface accuracy

Surface sampling evaluates geometry independently of vertex density.

Method	Mesh vertices ↓ Mesh-rendering evaluation	Uniform F1 ↑	Virtual-scan F1 ↑
GOF	11.63 M	0.31	0.38
RaDe-GS	10.06 M	0.35	0.39
MILo	4.36 M	0.34	0.42
Spokes	5.80 M	0.48	0.53

Tanks & Temples. Vertices: mesh-rendering evaluation (Table 2). F1: six-scene surface evaluation (Table 1); Spokes F1 uses two pivots, without PAM.

MILo has the fewest reported vertices; Spokes has the highest F1 among these methods.

Usable geometry still needs semantic handles

Our meshes are lightweight and have empty interior, making them well suited for downstream applications such as physics simulations.



MILo meshes used in downstream simulation.

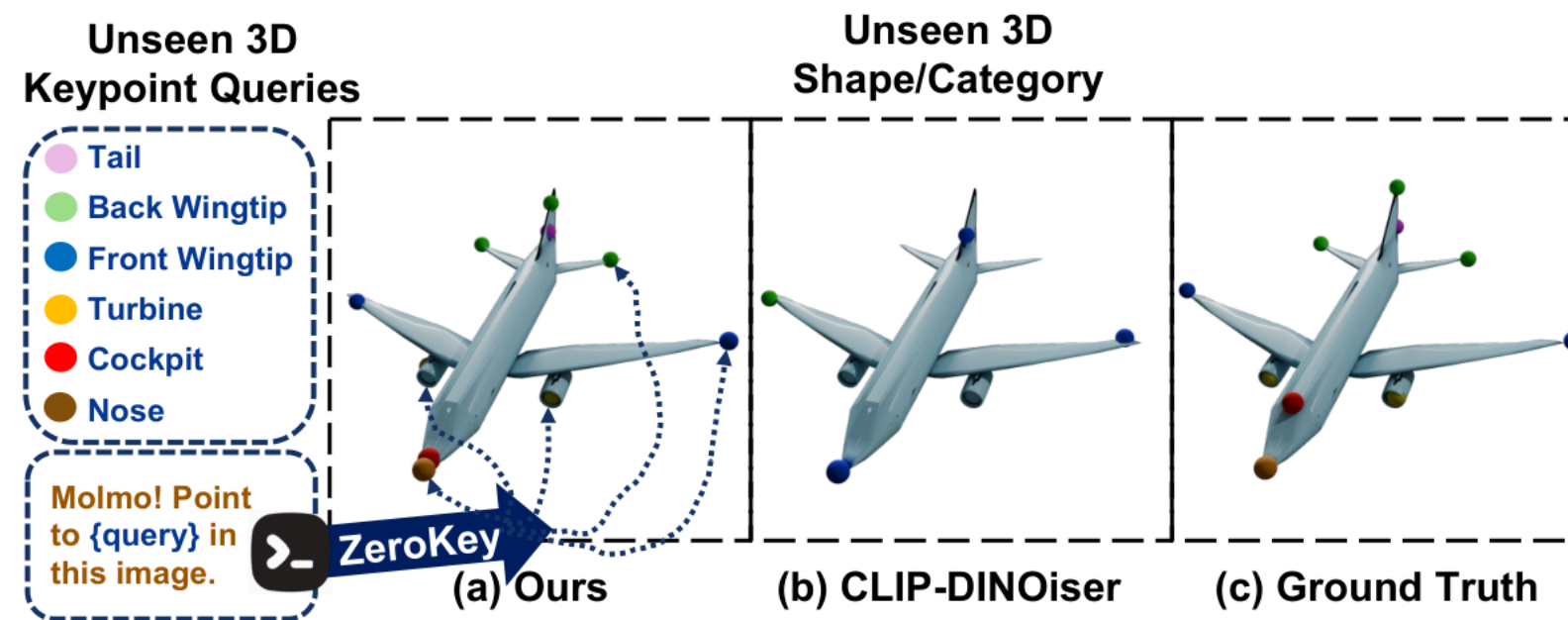
A surface can support interaction.

But geometry alone does not identify a hinge, a wingtip, or the part a user wants to edit.

The next question is how language can locate meaningful structure.

ZeroKey: connecting names to 3D points

Locate named 3D points through agreement across rendered views.

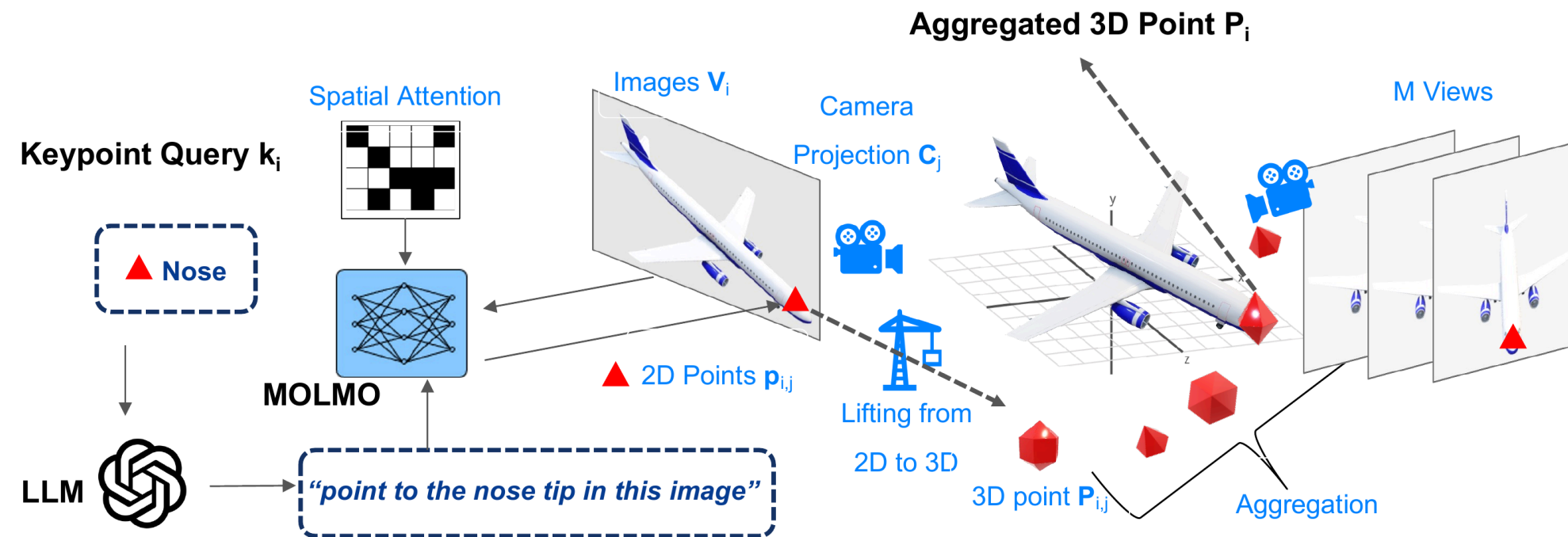


Query pretrained image models on a known mesh.

Zero-shot: no task-specific 3D keypoint training.

Bingchen Gong, Diego Gomez, Abdullah Hamdi, Abdelrahman Eldesokey,
Ahmed Abdelreheem, Peter Wonka and Maks Ovsjanikov

Language queries become 3D point hypotheses



Language → pixels

GPT-4o proposes names.

Molmo points in 26 rendered views.

Pixels → 3D hypotheses

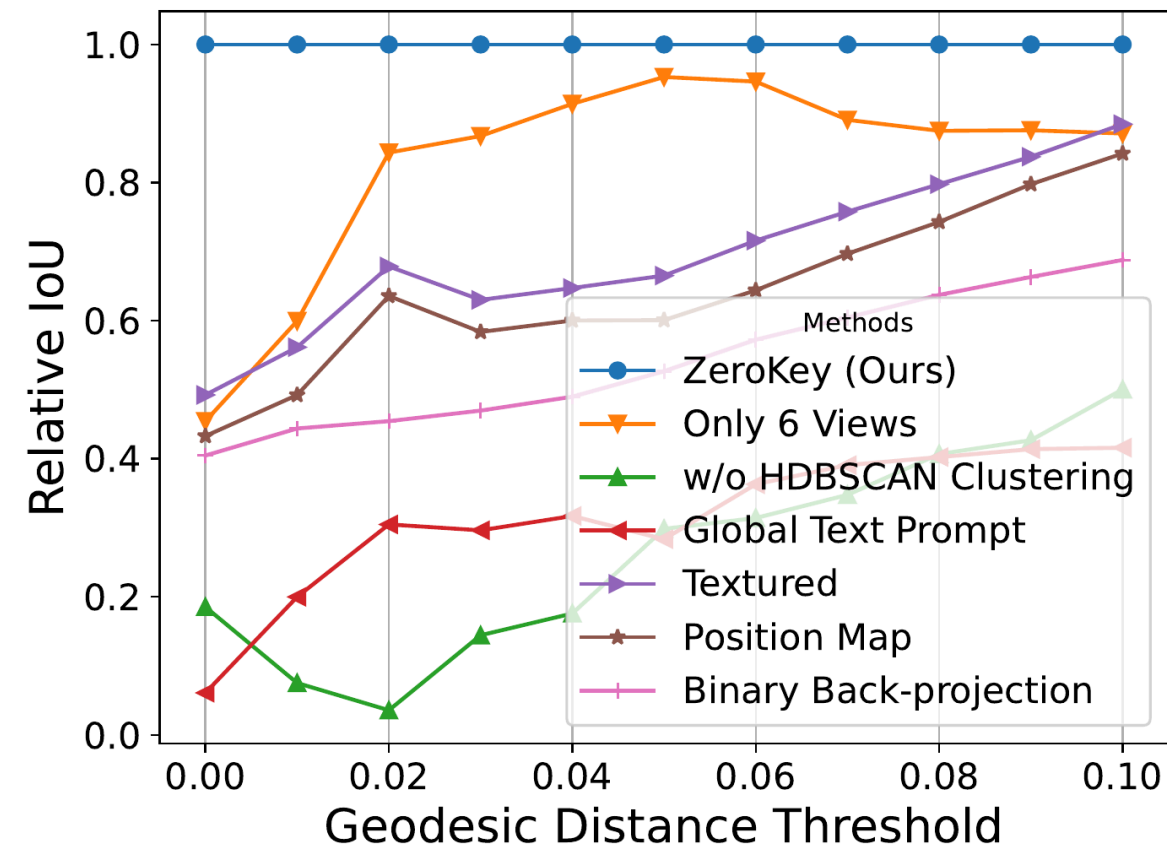
$$x_j = \pi_j^{-1}(u_j, D_j(u_j))$$

u_j : predicted pixel; x_j : 3D candidate.

Known camera π_j ; cached mesh depth D_j .

The mesh supplies geometry; the model supplies the point's meaning.

A name selects the target; 3D agreement stabilizes it



Relative IoU; full ZeroKey = 1 at each threshold.

Local uncertainty

Lift an $h \times h$ neighborhood; use Gaussian support weights.

Agreement in 3D

HDBSCAN separates consistent groups and rejects isolated detections.

One name can denote several corners. Preserve separate spatial hypotheses.

Global prompts, binary weights and direct averaging all reduce performance in this experiment.

Semantic grounding still leaves a precision gap

KeypointNet: airplane, chair and table. IoU (%) at geodesic distance thresholds.

Method	0.01	0.05	0.10
GPT-4o localizer	0.48	6.04	20.73
B2-3D · few-shot, 62 views	20.29	57.72	70.57
ZeroKey · 26 views	13.16	56.60	79.43

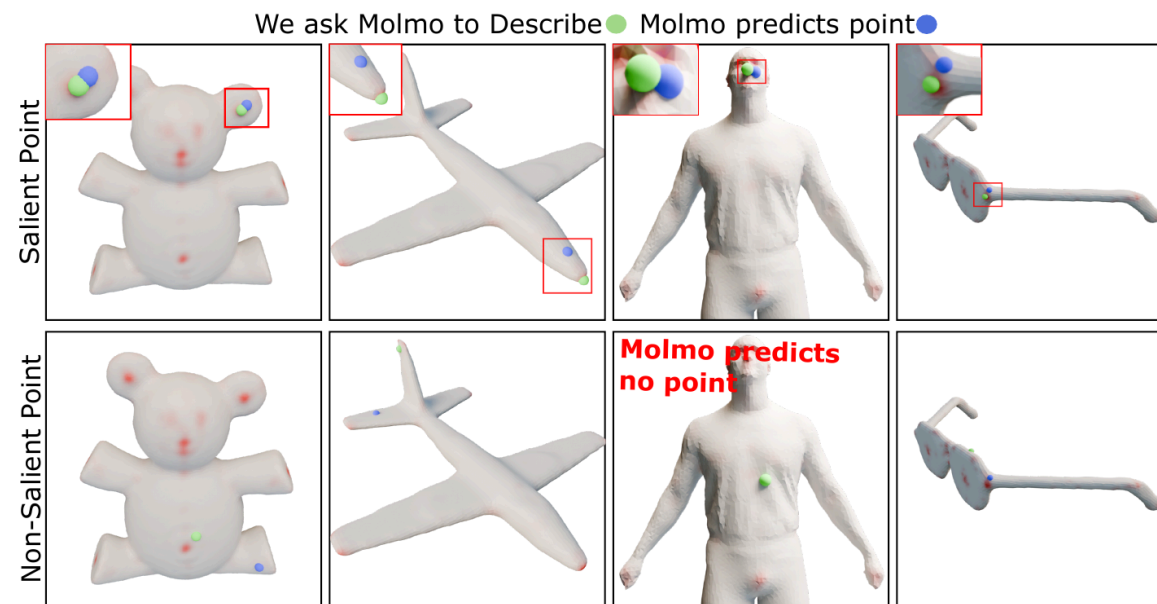
Benchmark queries are human-written descriptions of keypoint IDs.
ZeroKey requires no task-specific 3D keypoint training.

Names help identify the intended structure; tight tolerances expose localization errors.

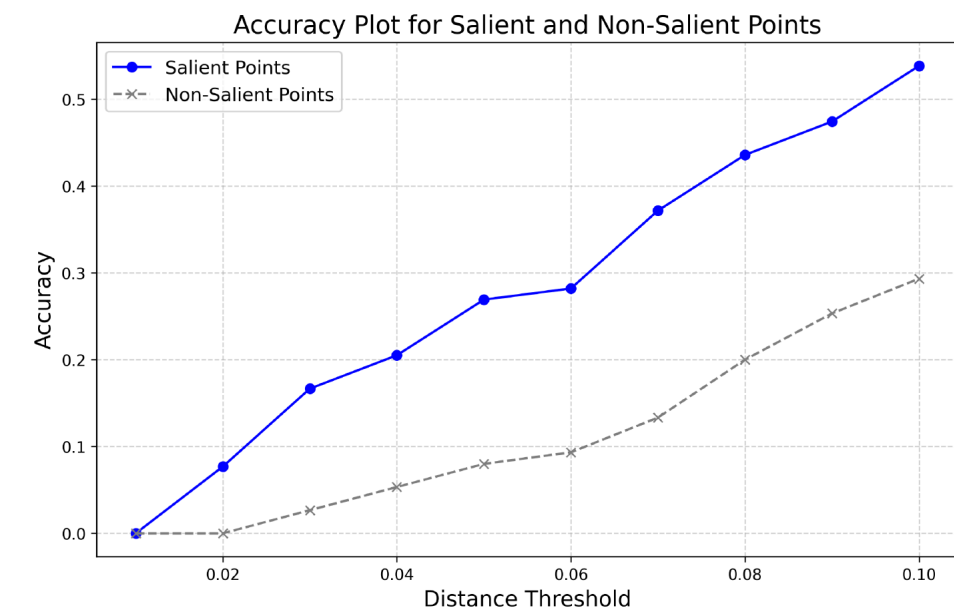
Can language recover the point it describes?

Nameability test: marked point $p \rightarrow$ short description $\ell \rightarrow$ recovered point $\hat{p}$

Keep the most repeated description; query new views with the marker removed.



Human saliency: Schelling-point selection density.



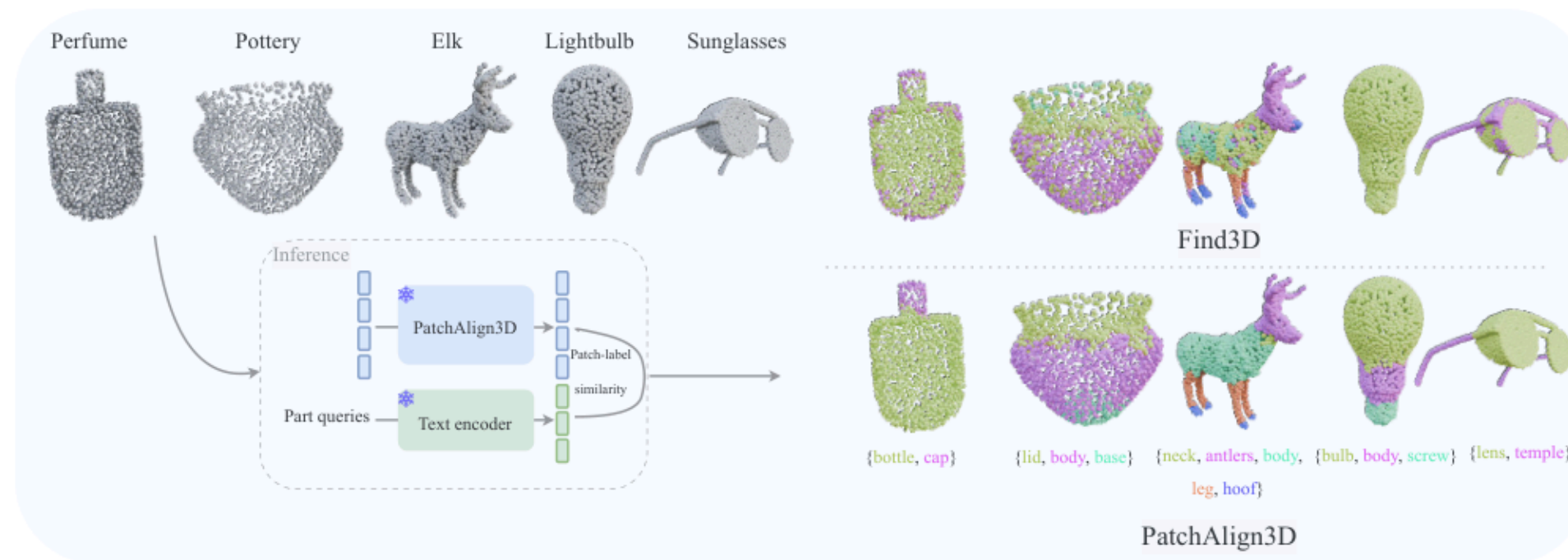
Retrieval stability: recovery within a 3D error tolerance.

Human-salient points are easier to recover from their descriptions.

Interpretation: distinctive names can give 3D points shared, repeatable identities.

PatchAlign3D: language-aligned local features

Encode the shape once; query regions with part names.

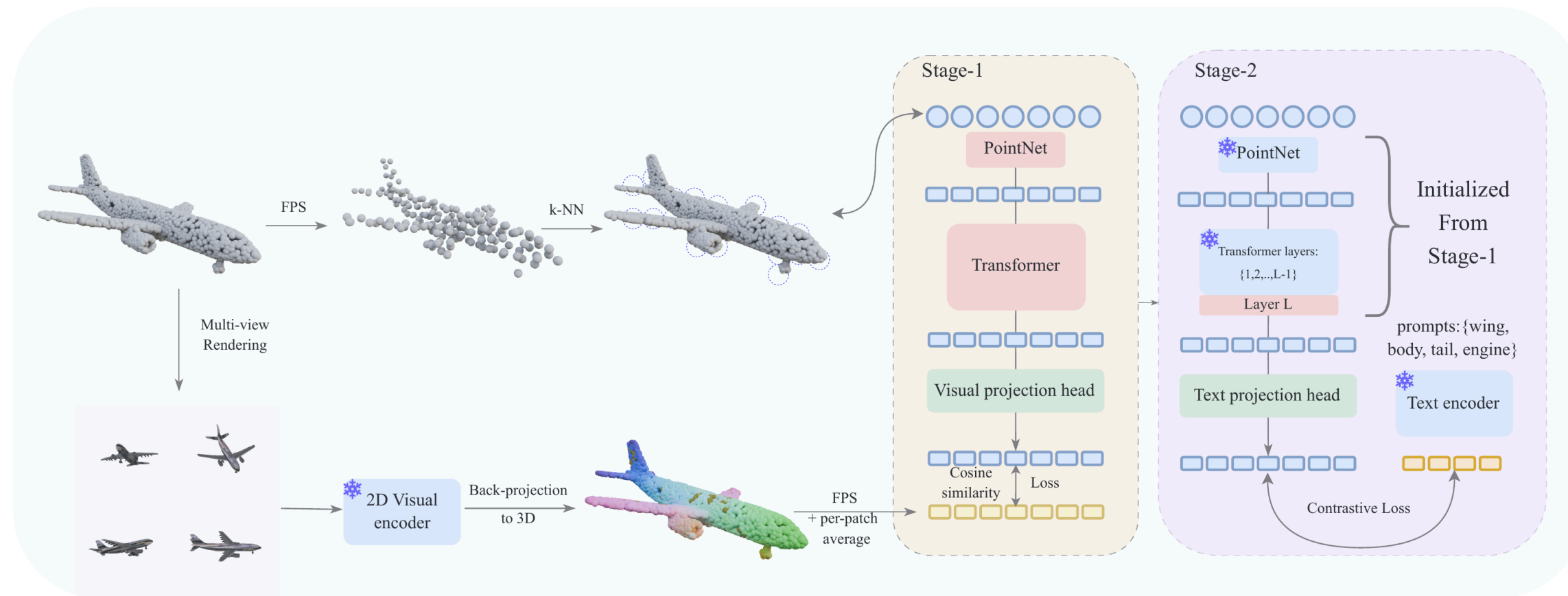


A trained 3D encoder matches local features to text embeddings.

Several part names can describe the same patch.

Souhail Hadgi, Bingchen Gong, Ramana Sundararaman, Emery Pierson,
Lei Li, Peter Wonka and Maks Ovsjanikov

Visual distillation, then local language alignment



2,048 XYZ points · 128 overlapping patches of 32 points · 12 transformer blocks

Stage 1 · visual structure

Match 3D patch features to frozen DINOv2 targets with a cosine loss.

Stage 2 · language alignment

Freeze the first 11 blocks; train the last block and projection into OpenCLIP's text space.

A patch can match several part descriptions

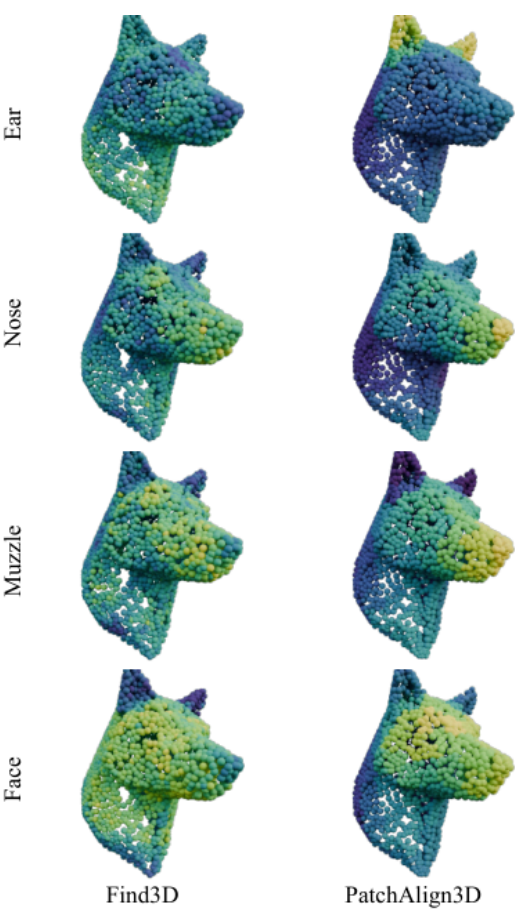
Training labels: 10 rendered views → SAM regions → Gemini part names → 3D pseudo-labels.

Fractional target y_{ij}
Share of patch i annotated with name j .

Name	Annotated points	y_{ij}
Nose	24 / 32	0.75
Muzzle	32 / 32	1.00
Ear	0 / 32	0.00

Illustrative patch with overlapping annotations.
The targets need not sum to one.

“Nose” and “muzzle” can both be valid positives.



Text-similarity maps; yellow means higher similarity.

Contrastive learning needs the right negatives

$$s_{ij} = \langle q_i, t_j \rangle / \tau + b, \quad p_{ij} = \text{sigmoid}(s_{ij})$$

$$L_{\text{text}} = -\sum_{i,j} [y_{ij} \log p_{ij} + (1 - y_{ij}) \log(1 - p_{ij})]$$

$q_i = h_{\text{text}}(z_i)$: projected patch feature; t_j : text embedding; τ : learned temperature; b : learned bias.

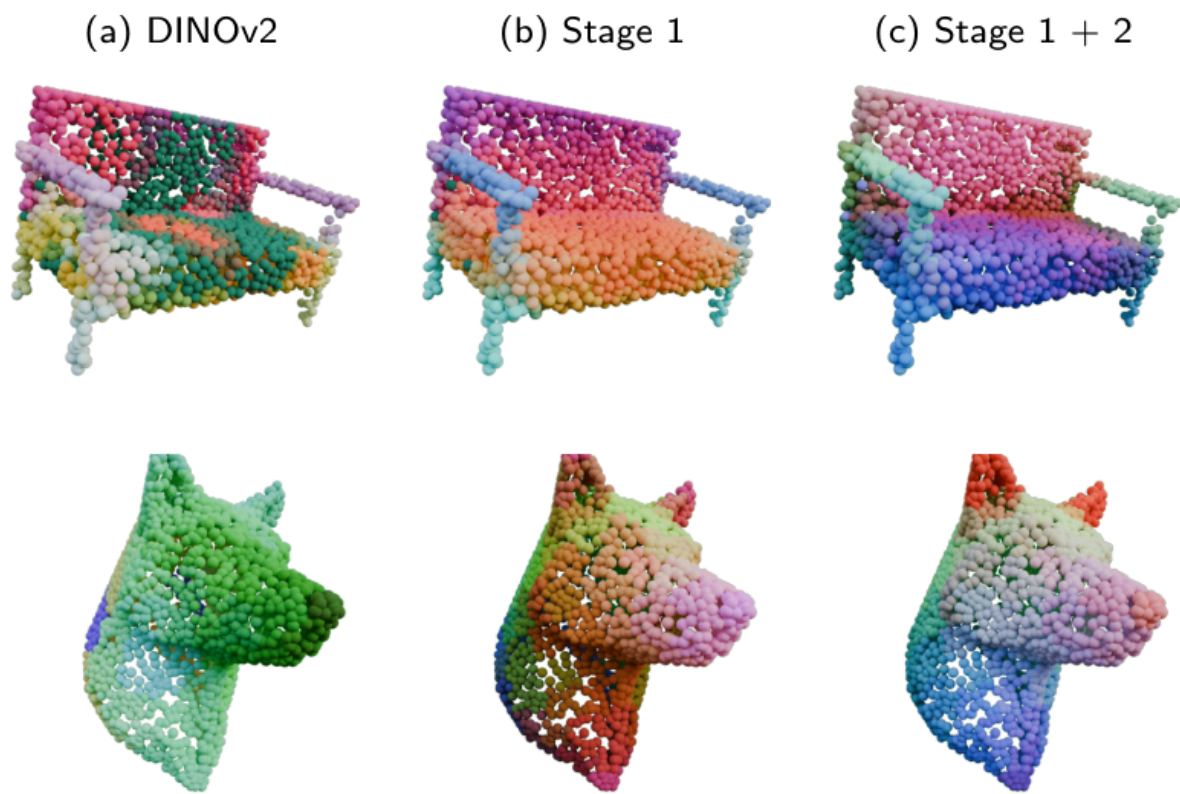
Independent patch–text pairs
Sigmoid allows several positive names.

Negatives within each shape
Use its valid part names with $y_{ij} = 0$.

ShapeNetPart · mIoU (%)	
Negative sampling	mIoU
Within shape	56.9
Also across shapes	45.3

A “leg” on another shape may be a false negative, not a different concept.

Visual structure first, language alignment second



Features visualized via PCA to RGB.

Independent PCA projections; colors do not match across columns.

Training order matters

ShapeNetPart · mIoU (%)

Training	mIoU
Language alignment only	50.5
Both losses jointly	50.2
Visual → language	56.9

Preserve local visual structure while learning what its regions are called.

Point-cloud baselines: COPS 25.6; Find3D 23.3 mIoU. Reported runtime: PatchAlign3D and Find3D both 0.4 s/shape.

Patch features can support ZeroKey’s point proposals

Semantic support for voting

Use patch–text similarity to modulate ZeroKey’s soft-voting weights.

Local features for clustering

Add learned feature cues when grouping the 3D point proposals.

Reported KeypointNet comparison · IoU (%) at geodesic distance thresholds

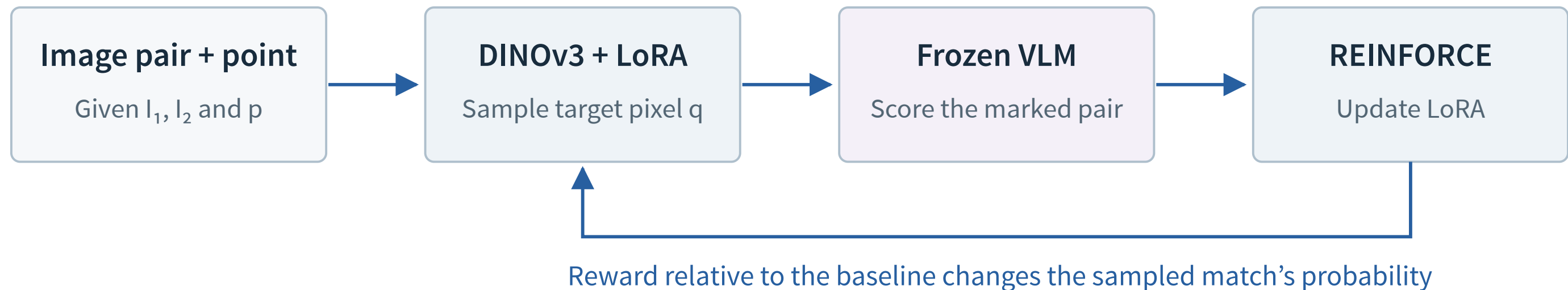
Method	0.01	0.05	0.10
ZeroKey	13.16	56.60	79.43
PatchAlign3D + ZeroKey	10.42	58.87	81.27

IoU improves at 0.05 and 0.10, but falls at 0.01.

The hybrid still uses ZeroKey’s multiview pipeline.

Is It a Good Match? A VLM as the training judge

Given two images and a source point, learn its target match.



ZeroKey

Query an image model for points during inference.

Good Match

Judge visual proposals during training; infer without a VLM.

Bingchen Gong · Viktoria Ehm · Emery Pierson · Maks Ovsjanikov

A one-step policy over target pixels

Context s

Images I_1, I_2 and source point p



Action q

One sampled target pixel



Reward $r \in [0, 1]$

“Same semantic part?”

A frozen VLM scores the marked pair.

Feature distance → policy

$$D(q) = \|f_1(p) - f_2(q)\|_2$$

$$\pi_\theta(q | s) = \exp(-D(q)/\tau) / \sum_{q'} \exp(-D(q')/\tau)$$

Shared encoder: $f_1 = g_\theta(I_1), f_2 = g_\theta(I_2)$.

$\tau = 0.1$ controls sampling diversity.

Adapt only the LoRA weights

$$W' = W + (\alpha / r_L) BA$$

W stays frozen. A and B learn.

Low-rank updates to ViT Q/K/V projections.

LoRA rank $r_L = 16$; scale $\alpha = 8$.

Train: sample $q \sim \pi_\theta$. Test: take $\operatorname{argmax}_q \pi_\theta(q | s)$, without a judge.

REINFORCE learns from a non-differentiable score

Reward above the recent baseline reinforces the sampled match.

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

Advantage: $r_t - b_t$

Frozen judge returns $r_t \in [0, 1]$.

Use b_t before adding the current reward.

$$b_{t+1} = 0.99 b_t + 0.01 r_t$$

Accumulate $B = 10$ samples per update.

Gradient through the policy

Differentiate $\log \pi$ through the adapter.

Hold reward and baseline fixed.

Positive advantage: increase probability.

Negative advantage: decrease probability.

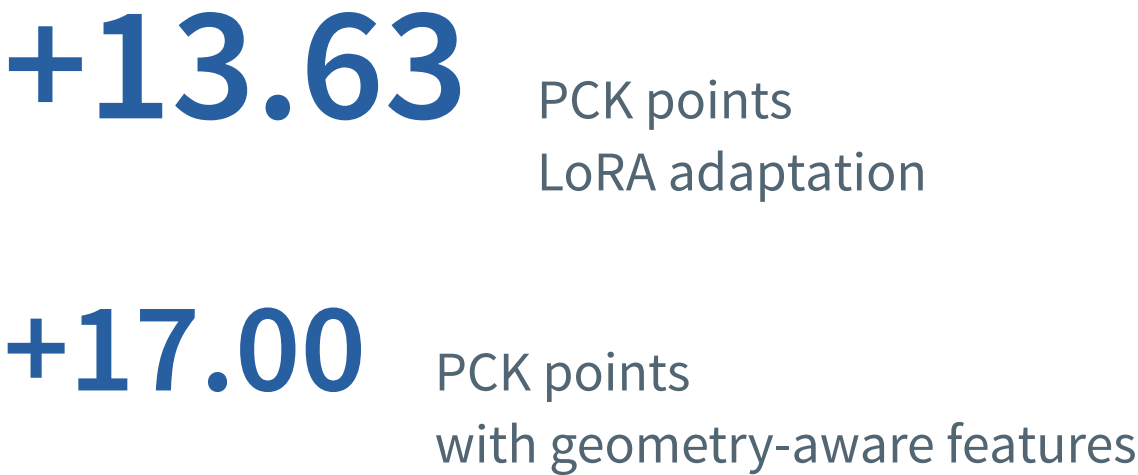
Keep valid zero scores; skip invalid responses.

No gradient through the VLM; no target-coordinate annotation in the reward.

VLM reward improves matching across 18 categories

SPair-71k, all 18 categories · Peak validation PCK@bbox 0.10 (%)

Method	PCK ↑
Frozen DINOv3	54.20
DINOv3 + LoRA	67.83
Geometry-aware + LoRA	71.20



Both trained models use REINFORCE with Gemini Pro reward.

Both gains are relative to frozen DINOv3.

Training and evaluation span the same 18 categories.

Source points are given. Target-coordinate labels do not enter the reward.

Reward precision and geometric feedback

Right part $\neq$ exact location

Qwen3-VL-32B-Thinking scores plateau within ≈ 20 px in a 126-keypoint cat probe.

Separate calibration probe with a different judge.

Separate cat-only transfer test

Cat-trained plain LoRA gains only +0.02 mean PCK across six other categories.

The 3D pilots do not establish successful transfer.

Feedback must distinguish the geometric errors the task needs to correct.

What changes under feedback?

Good Match · training

Propose a match $\rightarrow$ score $\rightarrow$ update visual weights.

3DHarness · inference

Execute a program $\rightarrow$ inspect renders or measurements $\rightarrow$ revise the program.

3DHarnessBench: a program as a 3D hypothesis

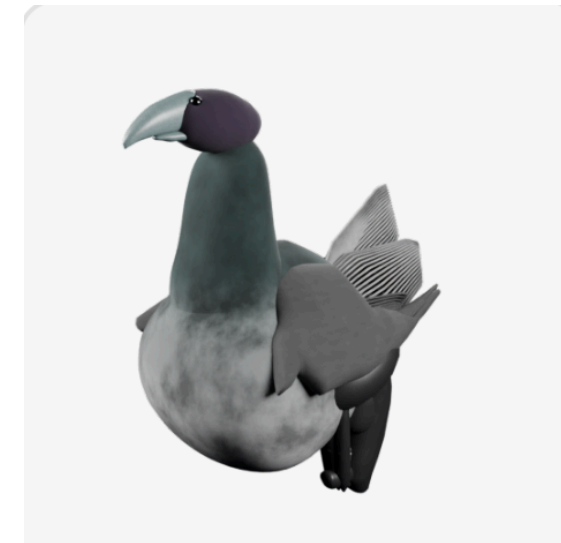
What does the agent believe
the object is made of?

- Parts and their dimensions
- Transforms, symmetry and repetition
- Geometry and material procedures

Ling Liu · Bingchen Gong
Amal Dev Parakkat · Maks Ovsjanikov

$$P \rightarrow E(P) = \hat{G}$$

Blender Python $\rightarrow$ Executed 3D object

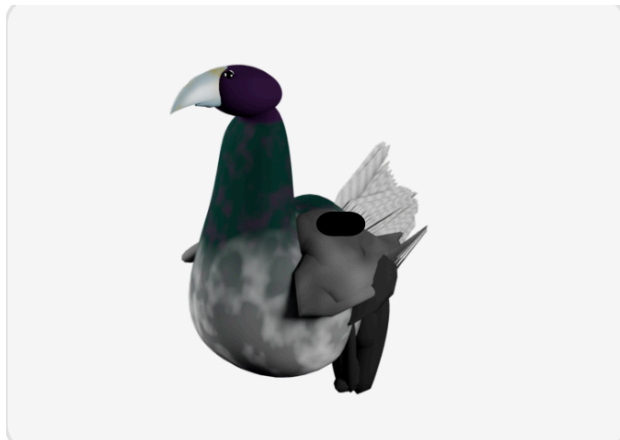


Execution makes the inferred structure observable and measurable.

100 targets from 3DCodeBench, one per category. Original source programs are withheld.

Code expresses repetition, symmetry and named parts

Target



Excerpts from the generated Blender Python

Repeat feathers along a row

```
for i in range(n):  
    s = i / float(n - 1)
```

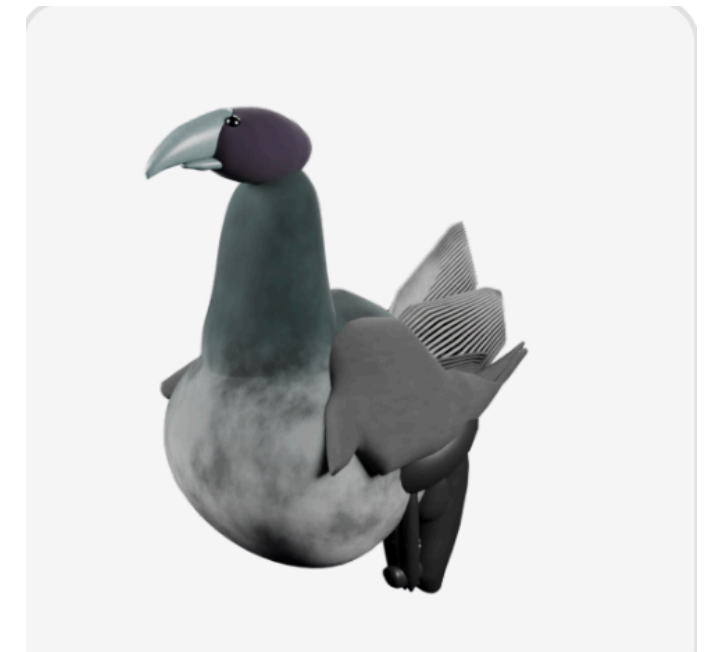
Mirror the wing geometry

```
wf.mirror_y()
```

Create a named component

```
wf.object(PFX + "wing_feathers")
```

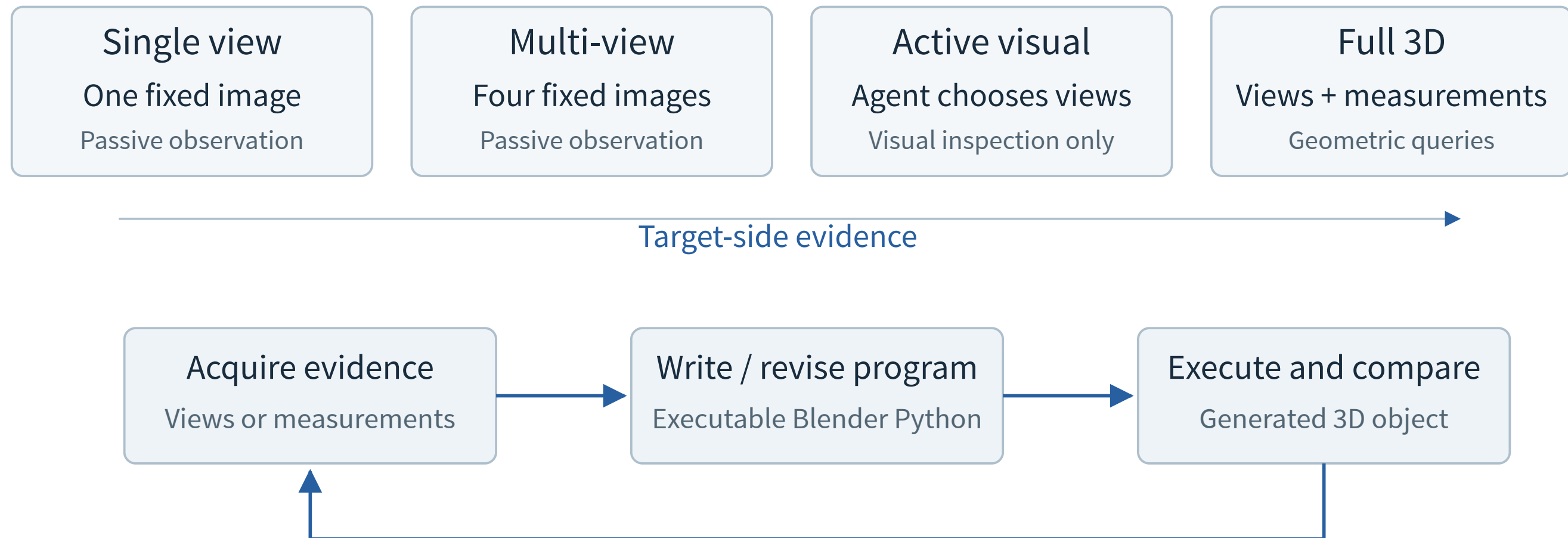
Result



Construction rules become explicit operations the agent can revise.

Opus 5, Full 3D Interaction. Placement calculations and coordinate tables omitted.

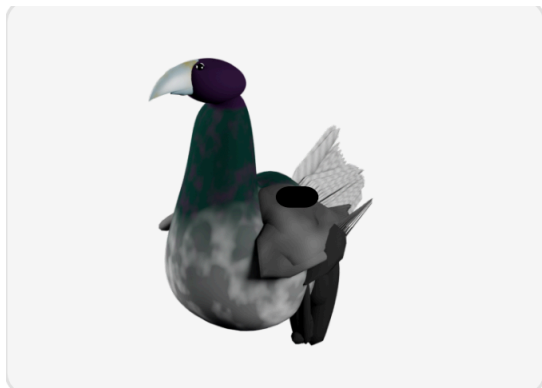
Observe, write, execute, inspect and revise



Revise P using evidence from the target and the executed object.
Fixed views: three refinement rounds. Interactive settings: agent-directed tool use.

Different evidence produces different executable structures

Target



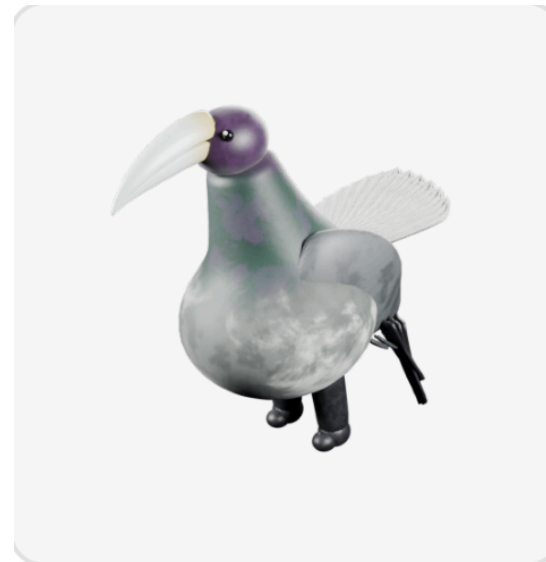
Reference

Single view



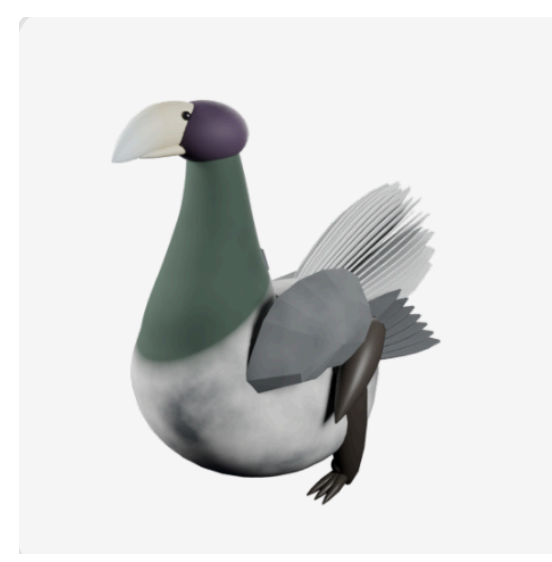
CD 0.017

Four fixed views



CD 0.014

Active visual



CD 0.005

Full 3D



CD 0.001

Four separate completed runs. Normalized, rotation-aligned Chamfer ↓ for this object.

Opus 5: richer evidence helps recover proportions, wings and legs.

Useful observations matter more than access alone

Across observation settings

Agent	4 views	Active	Full 3D
Opus 5	.748	.871	.927
Gemini 3.1 Pro	.753	.691	.783
GPT-6 Astra†	.842	.912	.956

Uni3D 3D–3D similarity ↑

Original agents: 100 targets.

† Preliminary update; coverage and protocol parity to verify.

Controlled viewport ablation

.3722 → .6148

MiniMax Uni3D after normalizing its viewport screenshots.

10 targets, agent still chooses its views.
Four fixed views on this subset: .6200.

Program revision depends on obtaining evidence that reveals what to change.

Measurements constrain the geometry expressed in code

Target evidence	Chamfer ↓	Uni3D ↑
Images	.0150	.8133
+ global	.0103	.8088
+ parts	.0099	.8557
+ global and parts	.0084	.8296

10 targets · Opus 5
Fixed multiview input plus measurements

44% lower normalized Chamfer distance

Global: center, bounds and dimensions.

Parts: anonymous centers, dimensions and repetition statistics.

Parts alone give the best Uni3D.
Combining cues does not improve every metric.

A program can express measured dimensions and relations as construction parameters.

Code provides a state that can be revised and re-executed

First program versus final program

Agent	Initial	Final
Opus 5	.914	.927
Qwen 3.8 Max	.796	.857
MiniMax M3	.566	.679
GPT-6 Astra†	unreported	.956

Full 3D · Uni3D ↑
Original repair study: 100 targets.
† Preliminary final aggregate; initial score unreported.

This benchmark evaluates reconstruction and interaction. Program quality and semantic editability remain open.

What the representation enables

- Inspect a construction choice.
 - Change a part, parameter or procedure.
 - Execute again and assess the result.
- Next: learn which observation and code revision will fix an error.
- Proposed follow-up: train on verified interaction traces and test semantic edits.

Five verified takeaways

01 Frequency control regularizes sparse-view reconstruction.

FourieRF · 5.57× faster than ZeroRF, -0.212 dB PSNR (4-view Blender).

02 Explicit surfaces improve geometric reconstruction.

MILO enforces depth/normal consistency; Spokes improves F1 under common surface sampling.

03 Semantics connects language and 3D geometry.

ZeroKey maps descriptions to 3D points; PatchAlign3D aligns local geometry with text.

04 VLM judgments can train better visual matchers.

Good Match · 54.20 → 67.83 peak-validation PCK with plain LoRA; no judge at inference.

05 Revising executable programs can improve reconstruction.

3DHarnessBench · Qwen 3.8 Max (Full 3D): 0.796 → 0.857 Uni3D over 100 targets.

Articulation exposes testable constraints

“Can the drawer open without hitting the chair?”



Concept illustration · motion axis and swept volume

An executable part description

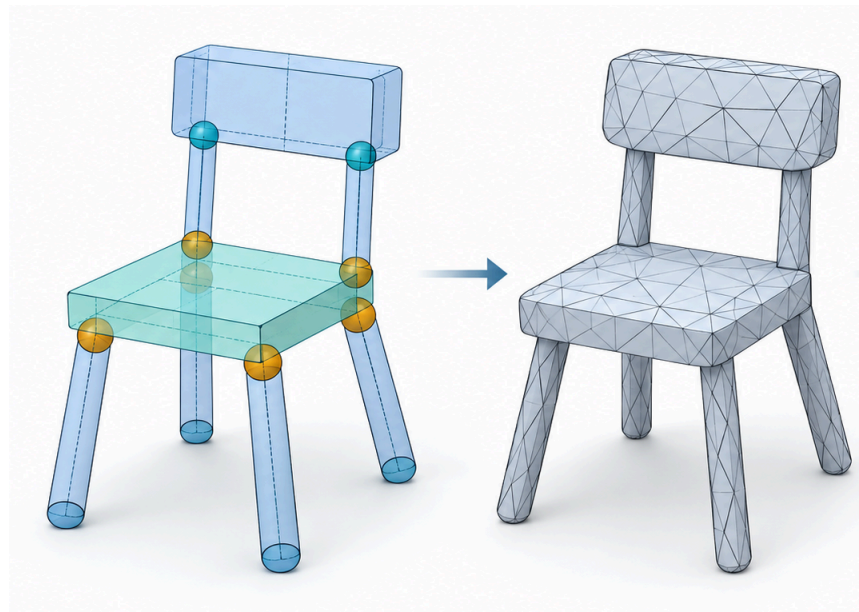
```
part: drawer  
parent: cabinet  
joint: prismatic  
axis: local x  
range: bounded  
constraint: no collision
```

Localize the part → execute the motion → check collision and appearance.

First test the operation with known geometry; then measure the errors introduced by reconstruction.

Explicit structure with neural appearance

Editable structure



Program + surface

Dimensions, parts, joints
and contact geometry

Appearance and light transport



Hair

Strands + Gaussian
appearance

Fog

Volumetric primitives +
scattering

Water

Surface + reflection /
refraction

Couple editable geometry with appearance models suited to each material.

Concept illustrations · Consistent relighting and optical edits need scattering, reflection and refraction models.

Can compact models learn from verified interaction?



Concept illustration · construction and repair traces

Astra as teacher, Qwen as learner

- **Collect:** Astra's tool calls, observations and Blender revisions.
- **Verify:** replay code and check geometry and task constraints.
- **Fine-tune:** predict the next tool call or code revision from observed results.

Astra traces

verified construction and repair



Qwen3.8-27B

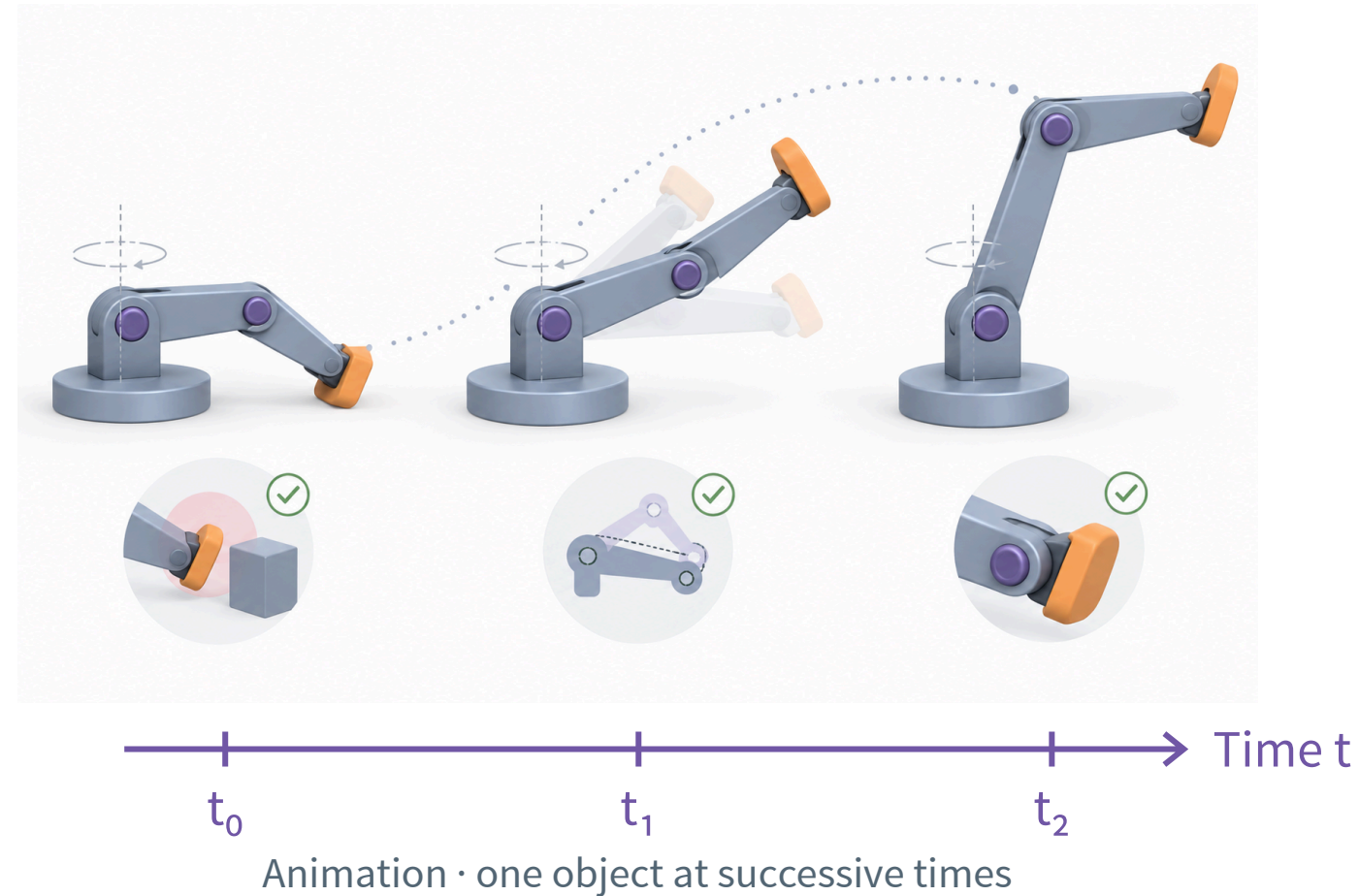
27B dense model · open weights



Transfer test

unseen objects and edits

4DHarnessBench would evaluate operations over time



Articulation

parts, axes and motion

Constrained edit

preserve unedited structure

Animation / video

temporal appearance and geometry

Spatial query

answer under executable tests

Compare representations and feedback with matched evidence and compute.

A first project with decisive tests

Verified Astra traces for reconstructing and editing articulated objects

Milestone	Deliverable	Decisive test
3D/4DHarnessBench	Static and articulated cases with exact checks	Known geometry, then reconstruction: isolate sources of error
Learn from interaction	Qwen3.8-27B trained on final code, full traces or verified traces	Transfer to unseen object and edit families at matched budgets
Demonstrate useful control	Constrained edits and spatial queries	Beat simpler representations while preserving unedited geometry

No transfer: revise supervision. No control gain: simplify the representation.

Complementary expertise for the research programme

Connection	Research opportunity	Relationship
Amal Dev Parakkat IMAGES team	Executable 3D construction and evaluation	Existing 3DHarnessBench collaboration
Kaouthar Messaoud	Geometry-grounded motion and video generation	Proposed collaboration
Stephan Alaniz · Jhony Giraldo	Fine-grained grounding and relational structure	Proposed collaboration
Enzo Tartaglione · Xiaoran Jiang	Compact models and Gaussian compression	Proposed collaboration
Jean Le Feuvre	GPAC streaming and end-to-end delivery cost	Proposed collaboration
Kiwon Um · S2A group IDS Department	Differentiable simulation and audio-driven animation	Proposed collaboration
Hi! PARIS	Seminars, summer school and internal calls	Chair attached to the post

Representations and feedback must support the task

Usable geometry exposes structure.

Semantic features identify what to change.

Execution and feedback test whether it worked.

The programme: *From Video to Animatable and Efficient 3D Worlds*.

The first project: verified trajectories for compact models that reconstruct and edit articulated objects.

The next step is to learn representations and operations together, with checks we can trust.

Bingchen Gong

bingchen.gong@lix.polytechnique.fr · gongbingchen@gmail.com