Asking the right questions

3D and Code

Bingchen Gong · Group meeting · 5 November 2026

Useful edits · Reusable structure · Physical prediction

4DCodeBench: reconstructing a real event

Reference video

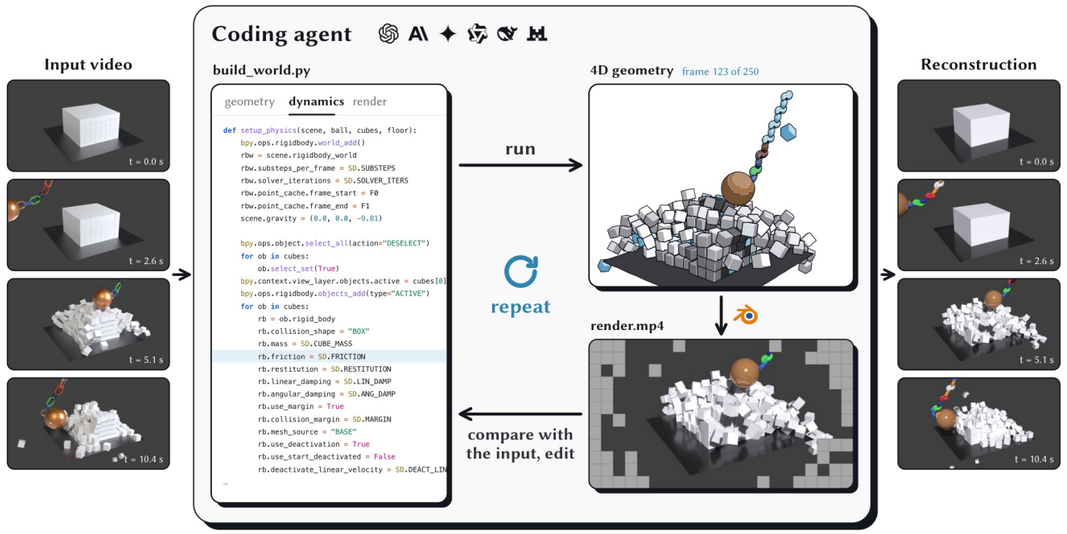
Agent reconstruction



The agent writes an executable scene from the video

4DCodeBench: task definition

A reference video → executable programs → a rendered reconstruction



Observe the video

$$x_{\text{ref}} = (x_0, \ldots, x_{T-1})$$

Generate executable programs

$$\Gamma = \mathcal{A}(x_{\text{ref}}) = (\gamma_1, \ldots, \gamma_m)$$

Execute geometry and render

$$\hat{x}_k = \mathcal{R}_\Gamma\left(s_\Gamma\left(\frac{k}{r}\right)\right)$$

$k = 0, \ldots, T-1$

k: frame index · T: total frame count

r: frames per second · k/r: time in seconds

Pasta reference video

219 frames · 24 fps · 9.125 seconds

Frame 24 is rendered at 1 second

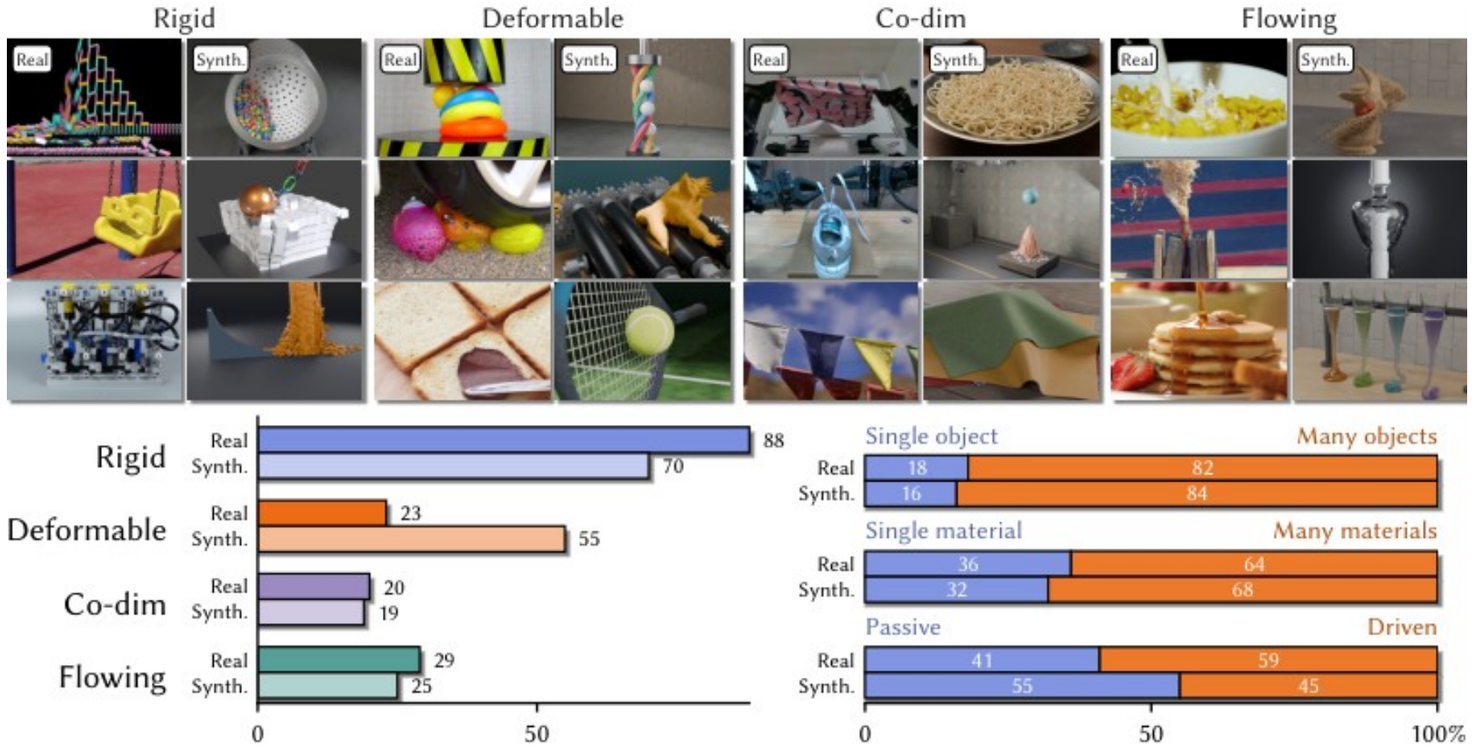
The generated code runs without the reference video

Match the reference resolution, frame count, and frame rate

4DCodeBench · §2.1, Eq. (1), and original Fig. A1

200 scenes, two sources of evidence

One reference video per task · up to 60 fps · real videos capped at 300 frames



100 real videos · 100 synthetic scenes

Real clips: 1.27–11.67 s · synthetic clips: 2.04–25.00 s

4DCodeBench · Fig. 3 · duration ranges computed from released metadata

What the evaluation observes

Metric family	Evidence
Perceptual	DINOv3 visual similarity
2D dynamics	Dynamic IoU, optical flow, point tracks
2.5D geometry	Depth error against estimated reference depth
3D geometry	Frame-0 surface Chamfer distance
3D dynamics	Trajectory DTW and step-displacement EMD

Complementary measurements; normalized scores are not scene success rates

Frame sampling for evaluation

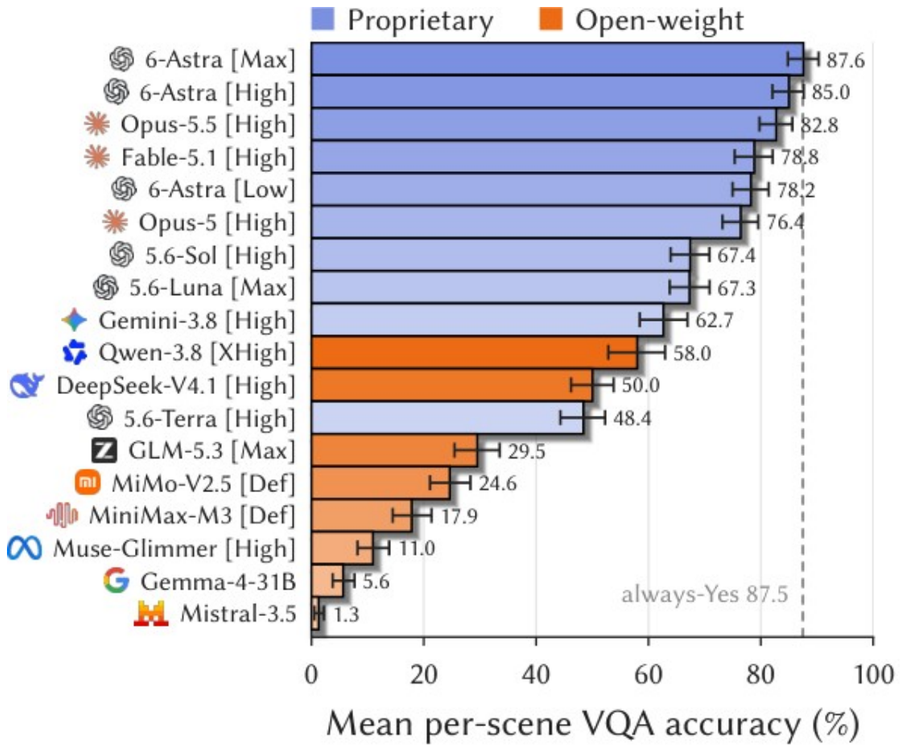
The output keeps every frame; each evaluator chooses its own sample

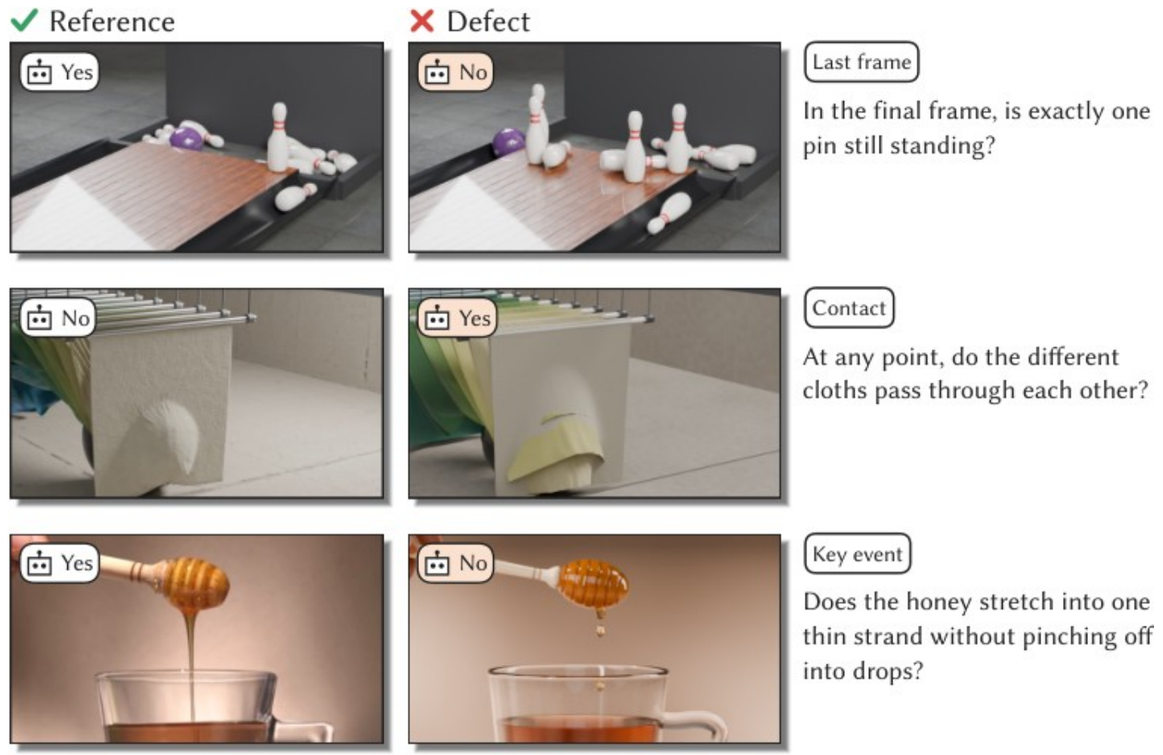
Stage	Frames used
Required rendered output	All frames at the reference frame rate and resolution
Sampled metric timeline	≤300 frames; integer stride for longer videos
VQA judge	16 reconstruction frames: first + last + 14 interior
Pairwise judge	3 fps from the reference and both reconstructions

Evaluator sampling does not restrict how the agent inspects its input

Questions probe events and contacts

Judge: Gemini 3.8 Flash · 16 reconstruction frames · no reference video

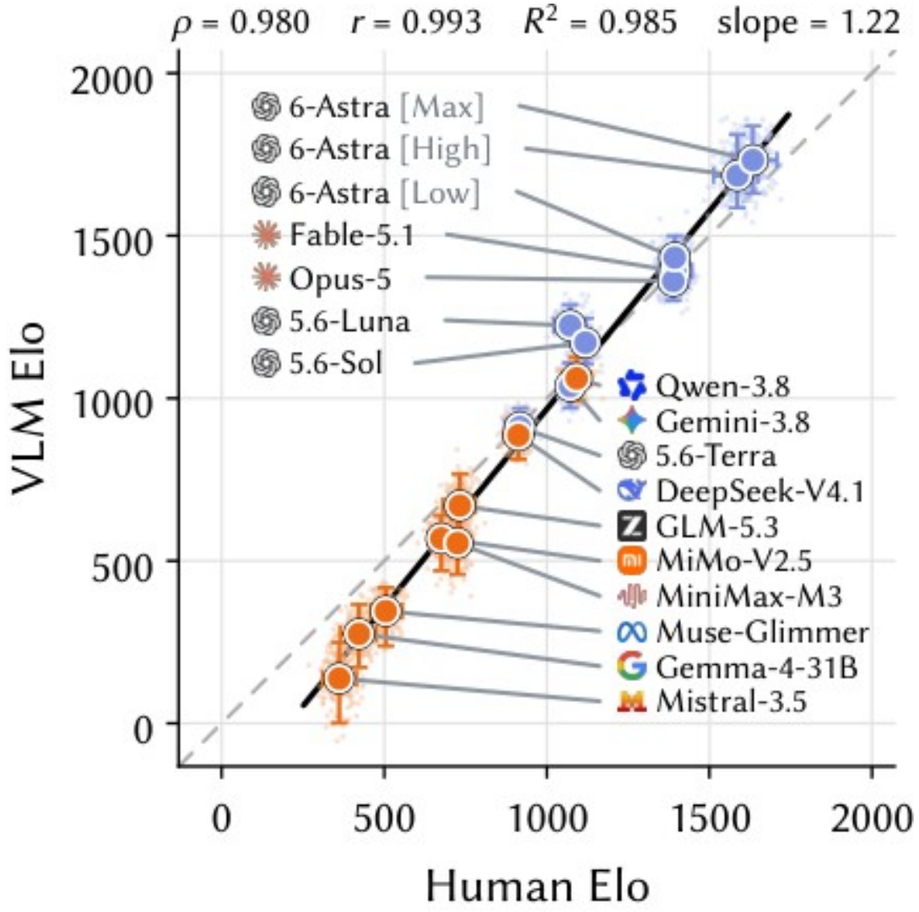




Complementary evaluation

VQA is reported separately from the five-family Overall score

Human agreement is strongest at model level



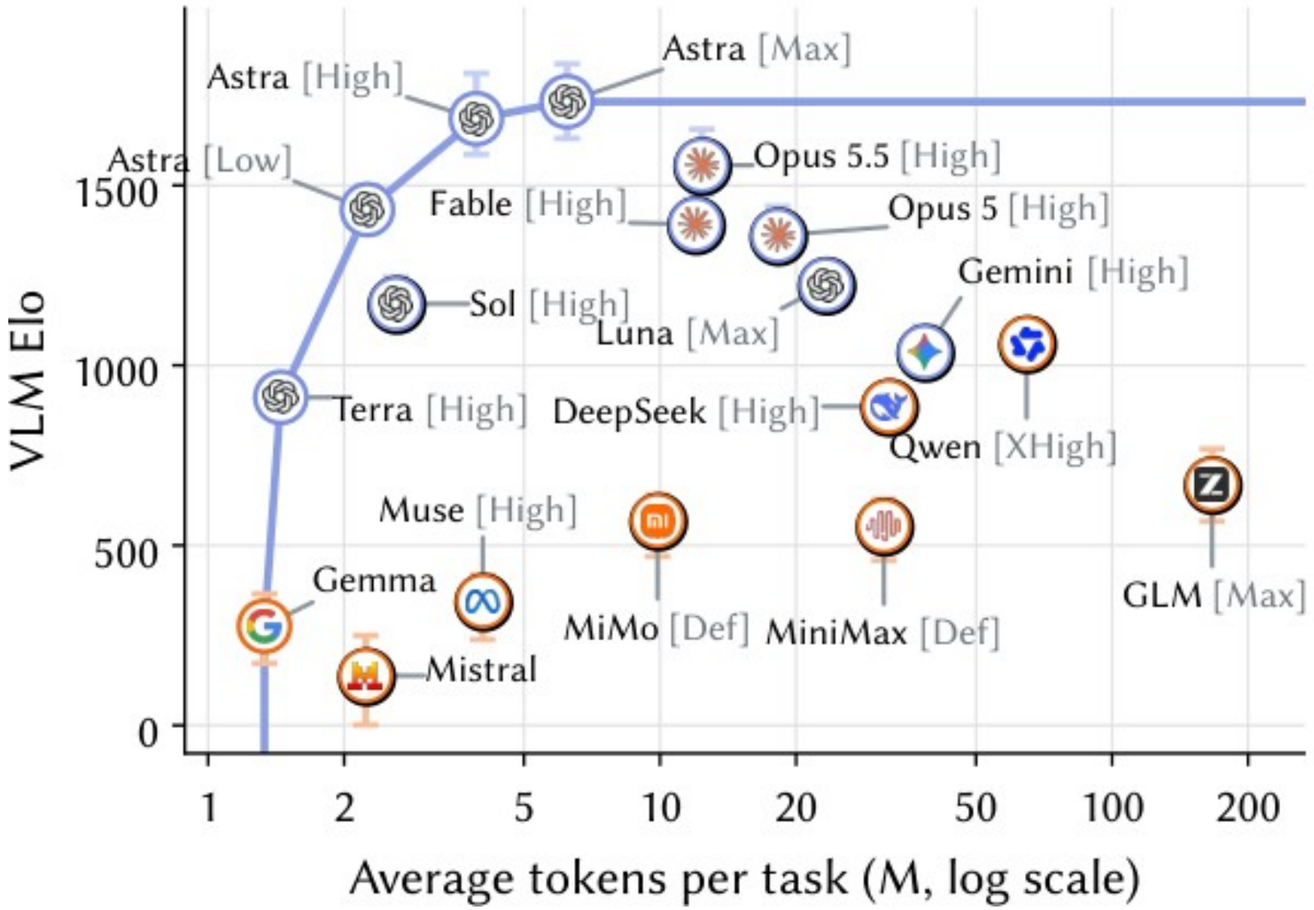
Preference rankings

Compare the reference with two reconstructions.

Strong aggregate agreement does not make every close judgment reliable

4DCodeBench · Figure 2, right panel · panel crop

More tokens do not guarantee a better reconstruction



Across models

Token use and output quality are different measurements.

Within-model reasoning effects require a separate comparison

4DCodeBench · Figure 2, left panel · panel crop

How agents express motion

Table A4: How models construct motion. Each solution either uses Blender physics, writes a custom simulation, computes motion analytically, or keyframes poses. Results cover the solutions of all 18 models; a few use other approaches, so some rows do not sum to 100%. The final row pools all solutions.

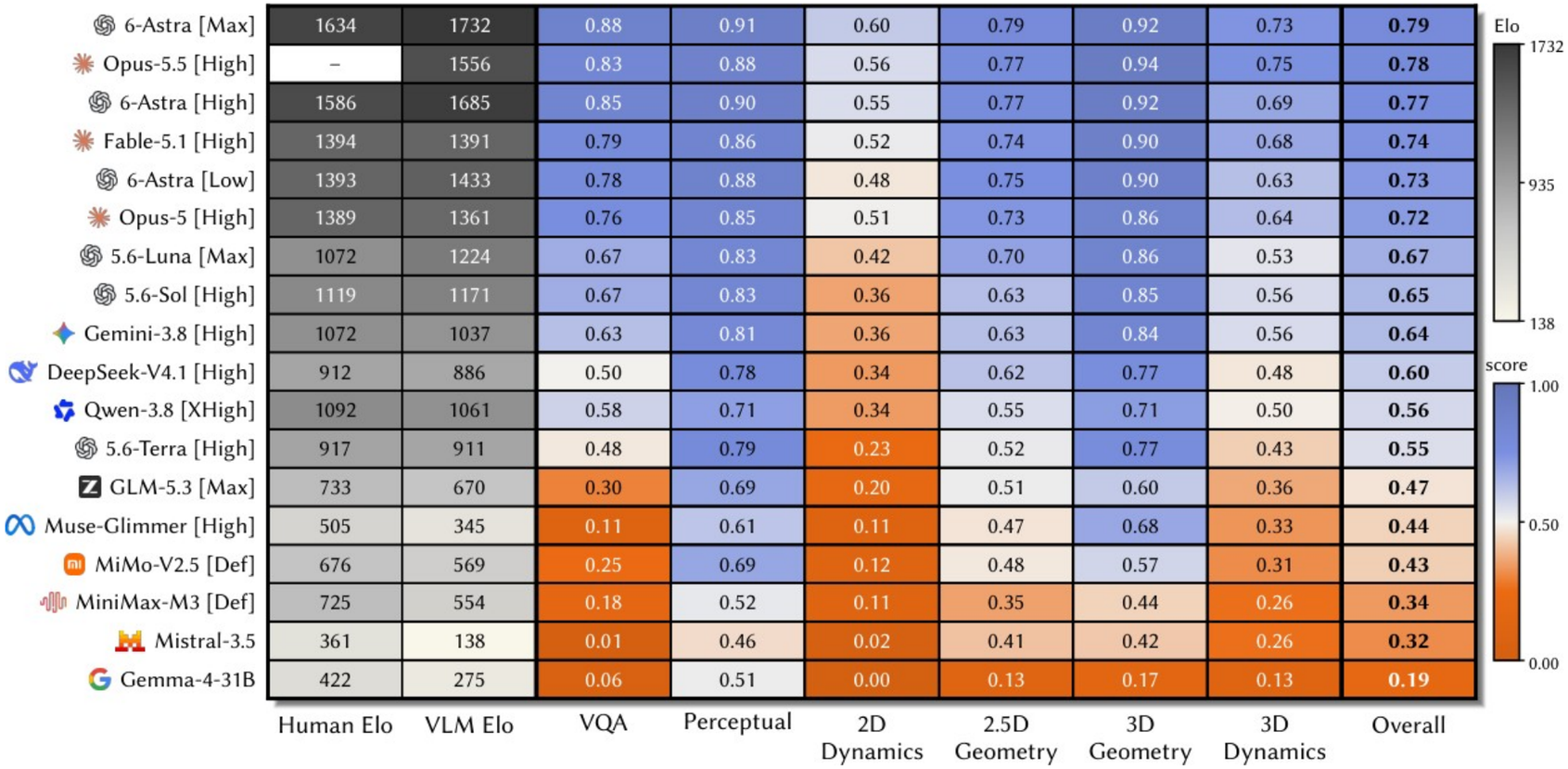
Model [Reasoning effort]	Blender	Custom sim.	Analytic	Keyframed
Claude Opus 5.5 [High]	15%	**61%**	22%	2%
Claude Opus 5 [High]	16%	51%	25%	7%
Qwen3.8 Flash [XHigh]	6%	45%	45%	4%
Claude Fable 5.1 [High]	22%	44%	27%	7%
DeepSeek v4.1 Flash [High]	10%	25%	56%	7%
GPT-6 Astra [Max]	8%	**24%**	**63%**	5%
GLM-5.3 Flash [Max]	14%	19%	56%	4%
GPT-6 Astra [High]	8%	14%	73%	5%
MiMo v2.5 [Def]	13%	14%	70%	3%
Gemini 3.8 Flash [High]	28%	12%	49%	11%
Mistral Medium 3.5	8%	9%	83%	0%
MiniMax M3 [Def]	15%	8%	74%	1%
GPT-6 Astra [Low]	7%	6%	85%	3%
GPT-5.6 Sol [High]	5%	5%	88%	1%
GPT-5.6 Luna [Max]	0%	5%	93%	2%
Gemma-4 31B	4%	2%	94%	0%
Muse Glimmer [High]	5%	1%	92%	1%
GPT-5.6 Terra [High]	0%	**0%**	**100%**	0%
Pooled	10%	19%	67%	3%

Pooled categories

67% analytic · 19% custom simulation · 10% Blender physics · 3% keyframing

A simulator call does not establish physical correctness

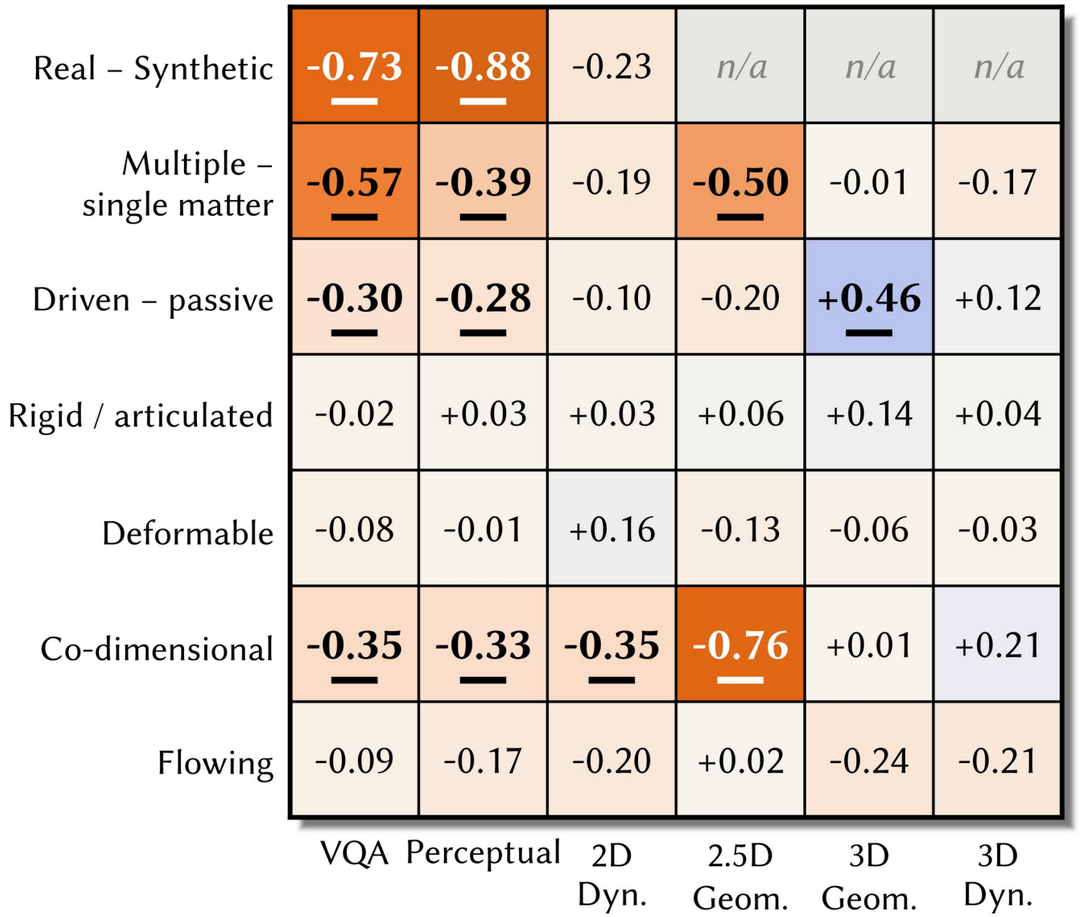
Read the leaderboard by capability



18 models · 200 scenes

Compare the individual families before interpreting the aggregate

Scene categories reveal different weaknesses



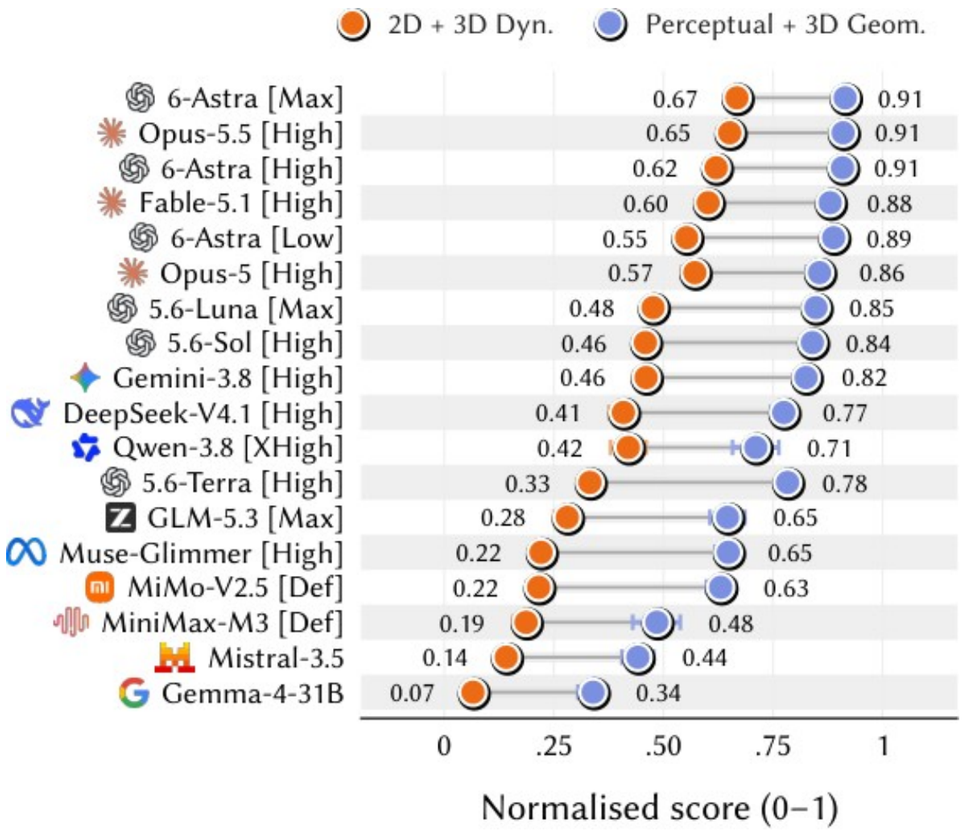
Category effects

Real versus synthetic; material families; interaction structure.

Category effects describe this evaluation, not universal physical difficulty

Static–dynamic gap

Static: appearance + 3D shape · Dynamic: 2D + 3D motion



Static score

Mean of Perceptual and 3D Geometry

Dynamic score

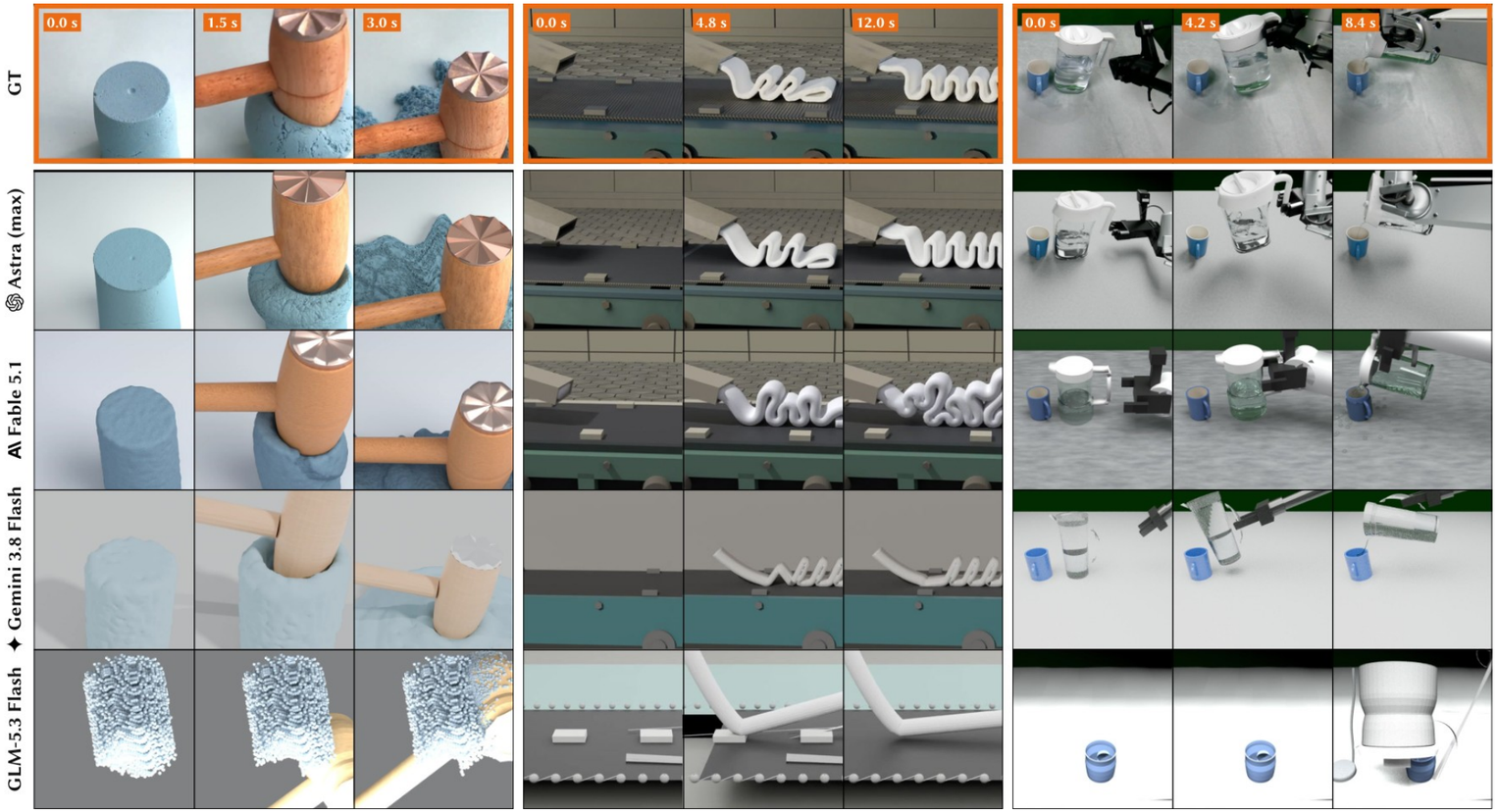
Mean of 2D Dynamics and 3D Dynamics

Difference

0.24 for Astra [Max]

Astra [Max]: 0.91 static · 0.67 dynamic (normalized scores)

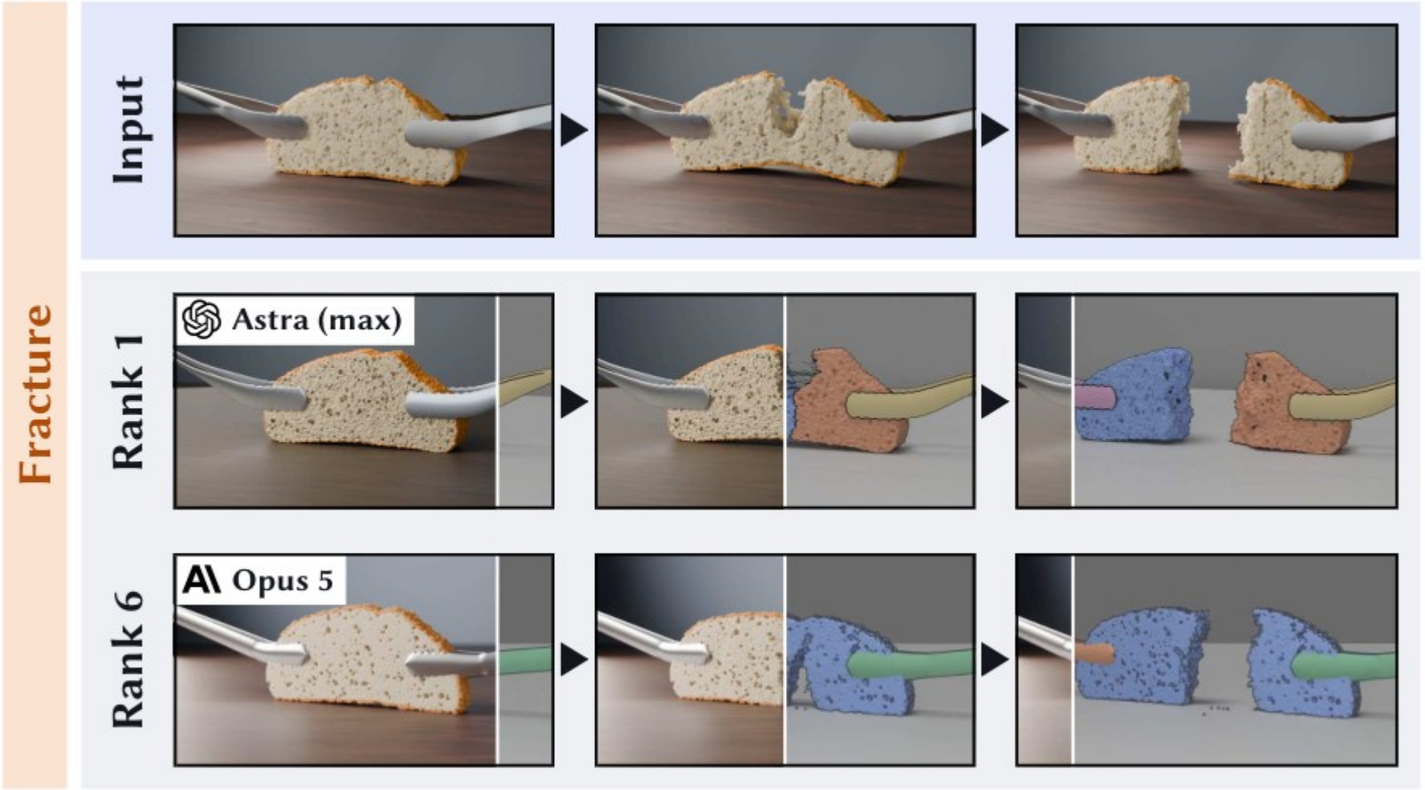
Inspect reconstructed motion frame by frame



Read the sequence

Plausible individual frames can hide errors in temporal evolution

Fracture tests changing geometry and topology

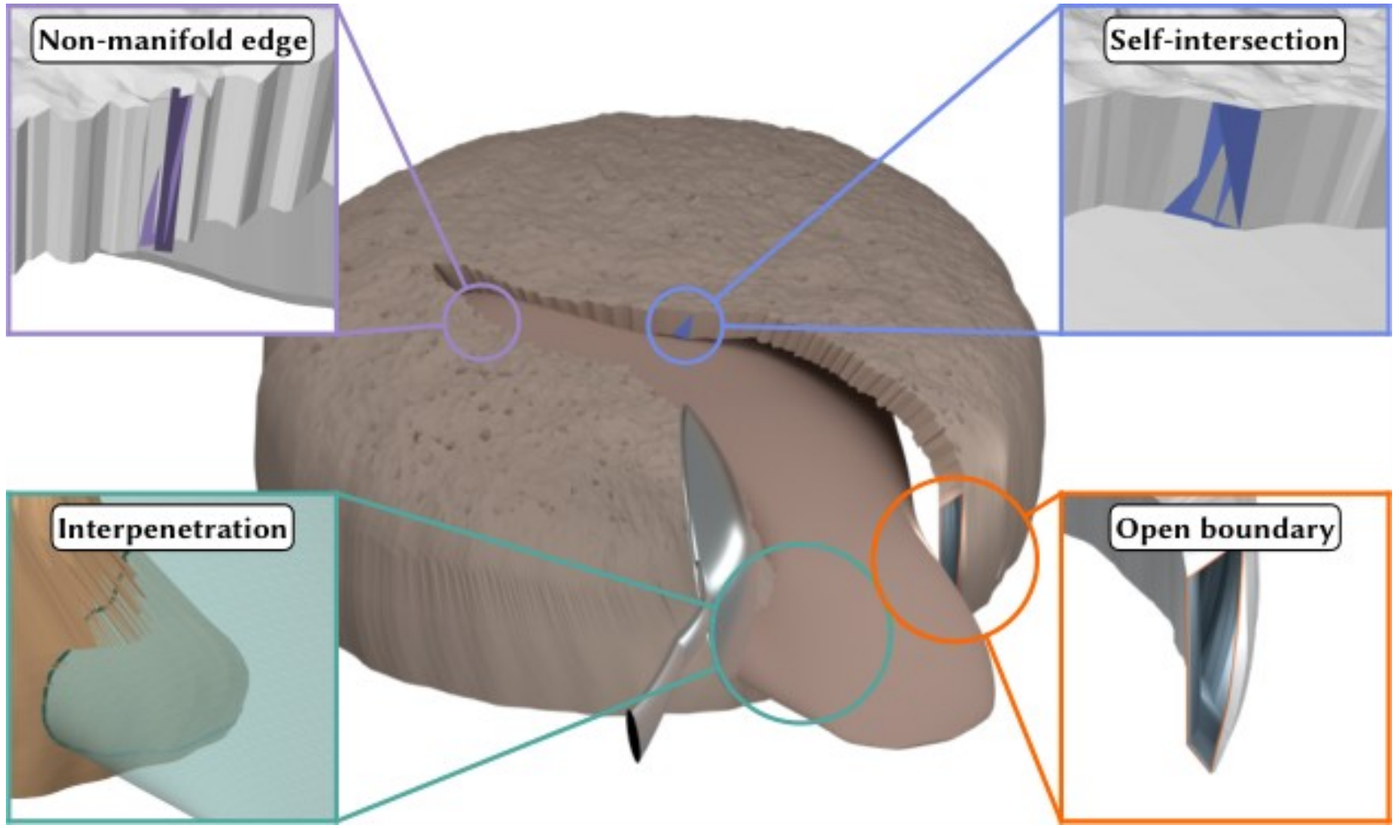


A selected paper example

A convincing image is only part of the reconstruction requirement

4DCodeBench · Figure 1, bottom-right fracture panel · panel crop

Mesh validity and reconstruction fidelity differ



Geometry diagnostics

Validity and fidelity answer different questions

4DCodeBench: limitations

Physical mechanisms	Prescribed trajectories can score well; the generating physics remains ambiguous.
Transfer to new conditions	Transfer to changed forces, initial conditions, and longer horizons remains untested.
Agent iteration	Evaluation focuses on final submissions; gains from each feedback round remain unclear.
Real-world evidence	No 4D ground truth: depth, flow, and tracks rely on estimates from vision models.
Dataset coverage	Stationary cameras and limited occlusion are favored; humans and animals are excluded.

What evidence would establish usefulness beyond the observed clip?

4DCodeBench · §2.2, §2.4, §6 · Details in speaker notes

When several programs generate the same shape, what makes one recovered program better?

The table example in Document review · Both programs start at W = 2



Observed: W = 2 **Program A: W = 3** **Program B: W = 3**

A · Fixed leg positions

```
tabletop(width=W)
for x in (-0.8, 0.8):
    for y in (-0.3, 0.3):
        leg(x=x, y=y)
```

W = 3: legs stay at x = ±0.8

B · Fixed edge inset

```
inset = 0.2
tabletop(width=W)
for x in (-W/2+inset, W/2-inset):
    for y in (-0.3, 0.3):
        leg(x=x, y=y)
```

W = 3: legs move to x = ±1.3

“Better” depends on the intended edit; the original shape cannot choose the rule.

Pseudocode adapted from Document review, p. 27 · Concept illustration: AI-generated

Where should symbolic structure end and learned or splatted detail begin?

Terrace count and attachments in code; irregular surfaces in a learned residual



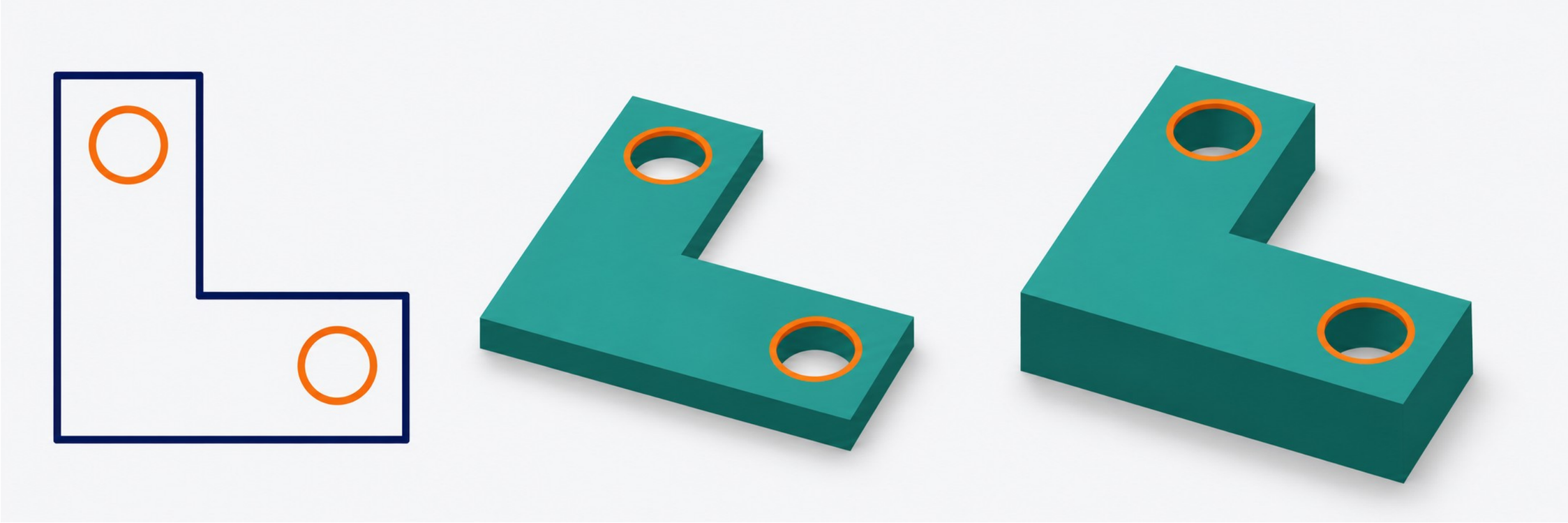
Symbolic structure | **Learned / splatted detail** | **Combined reconstruction**

Can flexible detail hide a wrong structural program?

Concept illustration · AI-generated · Proposed experiment

When does a 2D sketch or vector representation make 3D program inference easier?

Hypothesis: construction sketches simplify compatible object families

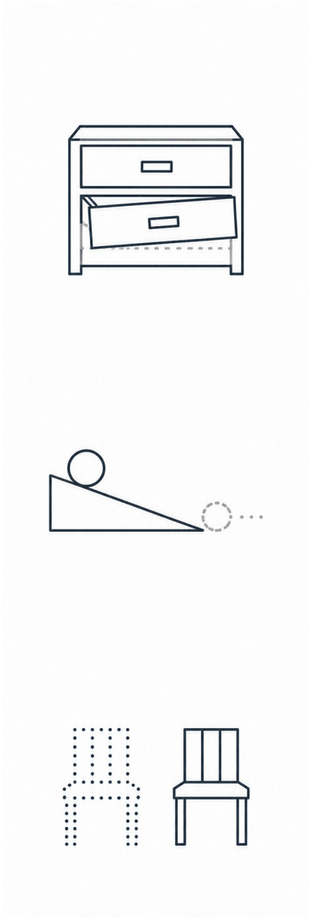


Closed 2D profile **Extrude through depth** **Change the thickness**

Oracle sketches separate the value of the intermediate from the difficulty of inferring it.

Concept illustration · AI-generated · Proposed experiment

Three further research questions



Which feedback best repairs each kind of spatial error?

When has a program recovered a physical mechanism rather than reproduced one observed motion?

Can a Gaussian scene be converted into a program that recovers reusable construction rules?

Open questions from Document review · Concept illustration: AI-generated

Try the multi-representation demo

Open it on your device and compare what each representation exposes



Mesh

Isolate a part;
explode the assembly.

Implicit field

Change turret count;
inspect a cross-section.

Spectral coefficients

Change M and N;
inspect the decoded height.

Try it yourself: 3dharnessbench.com/wtf

Presentation edition shown · Technical definitions and source details in speaker notes

Geometric representation and programming style

What does each representation make explicit, and what does it assume?

Explicit mesh

$$\mathcal{M} = (V, F), \quad V \in \mathbb{R}^{n\times 3}, \quad F \in \{1, \ldots, n\}^{m\times 3}$$

Vertex positions V and triangle indices F; a program supplies their construction.

Implicit field

$$f_\theta : \mathbb{R}^3 \to \mathbb{R}, \qquad \Omega_\theta = \{\mathbf{p} : f_\theta(\mathbf{p}) \le 0\}$$

A function encodes solid membership; its sign defines inside and outside.

Spectral coefficients

$$h(\mathbf{u}) = \sum_{j=1}^{K} c_j \phi_j(\mathbf{u}), \qquad \mathbf{u} \in D$$

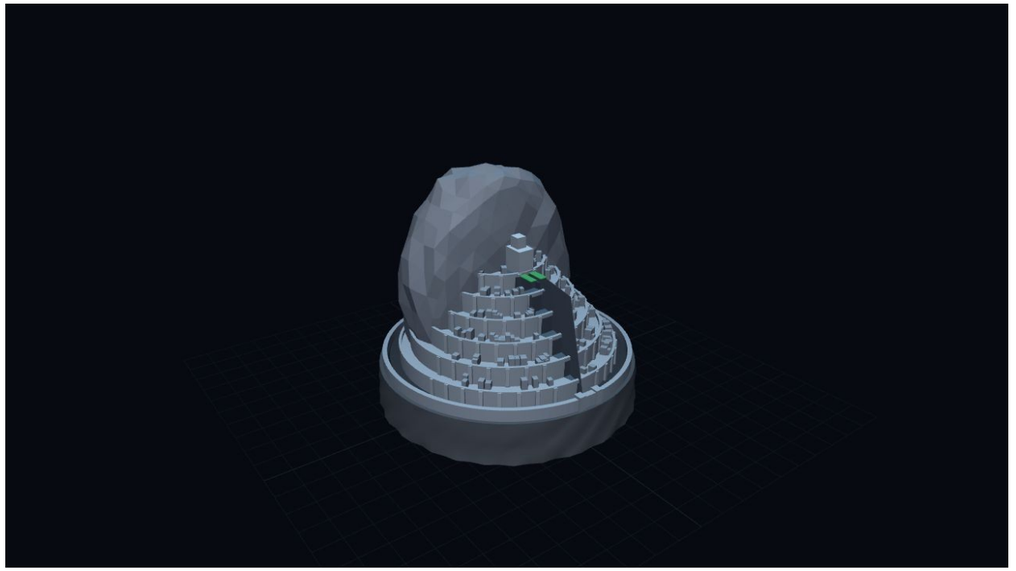
Coefficients c require a chosen domain D, basis functions, and a decoder.

Imperative and declarative describe how a program is expressed; either can construct these geometries.

Minas Tirith: explicit procedural geometry

One reference photograph; a program that exposes parts and repeated construction





User-provided reference

```
for (let i = 0; i < 6; i++) {
  const r = 4.2 - 0.48*i;
  const top = 1.6 + 0.62*(i+1);
  const g = new THREE.CylinderGeometry(r,r,0.62,64);
  // mesh center: (0, top - 0.31, 0.12*i)
}
```

Hand-authored Three.js model

$$R_i = 4.2 - 0.48i, \qquad i = 0, \ldots, 5$$

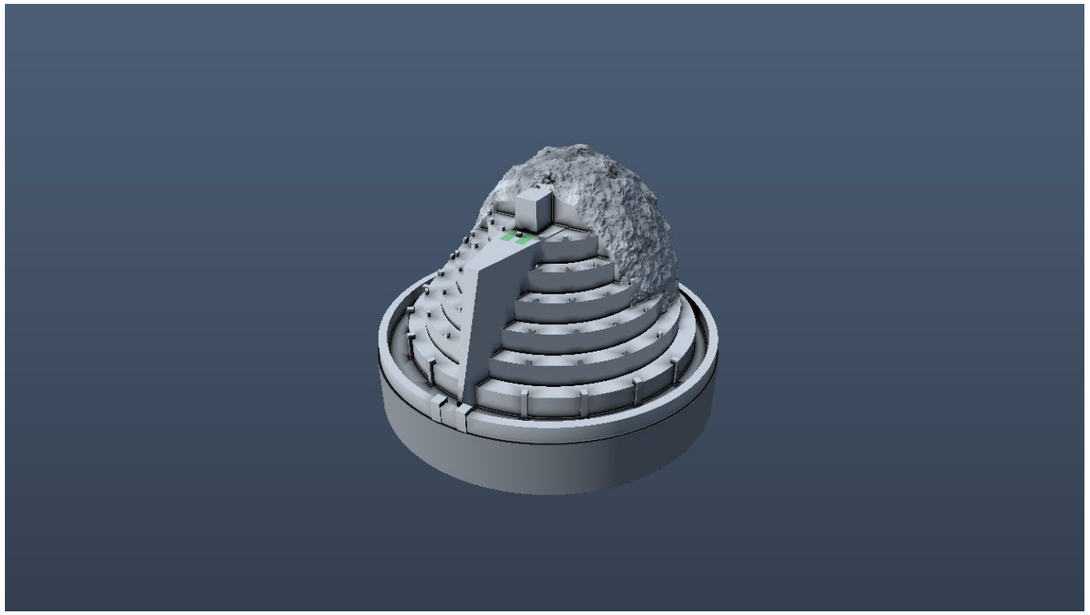
$$Y_i = 1.6 + 0.62(i + 1)$$

Named parts and transforms expose the construction rules.

Supplied renderer · Mathematical definitions and implementation details in speaker notes

Implicit geometry: a field and set operations

The sign describes the solid; distance guarantees require additional conditions



$$\Omega_f = \{\mathbf{p} : f(\mathbf{p}) \leq 0\}$$

$$f_{A \cup B} = \min(f_A, f_B)$$

$$f_{A \cap B} = \max(f_A, f_B)$$

$$f_{A \setminus \mathrm{int}(B)} = \max(f_A, -f_B)$$

```
float r = length(p.xz);
float f = max(r-5.0, abs(p.y-0.8)-0.8);
```

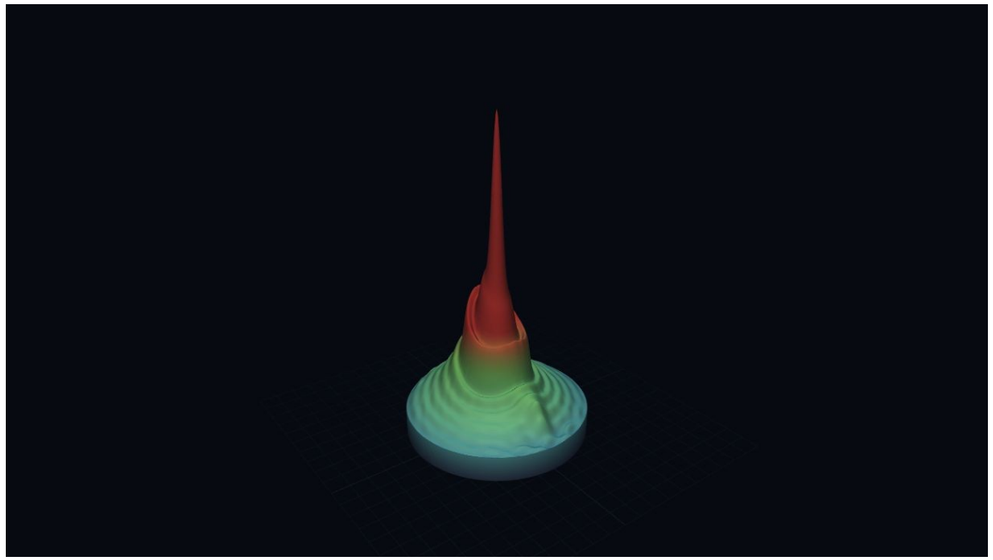
This cylinder field has the right sign; its value is not always Euclidean distance.

$$\mathbf{p} = (6, 2.6, 0): \quad f(\mathbf{p}) = 1, \quad d(\mathbf{p}, \partial\Omega) = \sqrt{2}$$

Supplied renderer · Mathematical definitions and implementation details in speaker notes

Spectral representation: fixed domain and decoder

The coefficients describe a height function on a known disk



$$\phi_{mn}(r,\theta) = J_m\left(j_{m,n}\frac{r}{R}\right)\cos(m\theta)$$

$$h_{M,N} = \sum_{m=0}^{M-1}\sum_{n=1}^{N} a_{mn}\phi_{mn}$$

$$y(r,\theta) = \max\{0,\ 1.4 + h_{M,N}(r,\theta)\}$$

J_m: Bessel function · $j_{m,n}$: its nth positive zero

Disk radius R = 5 · M ≤ 32 · N ≤ 18
576 coefficient slots at full settings

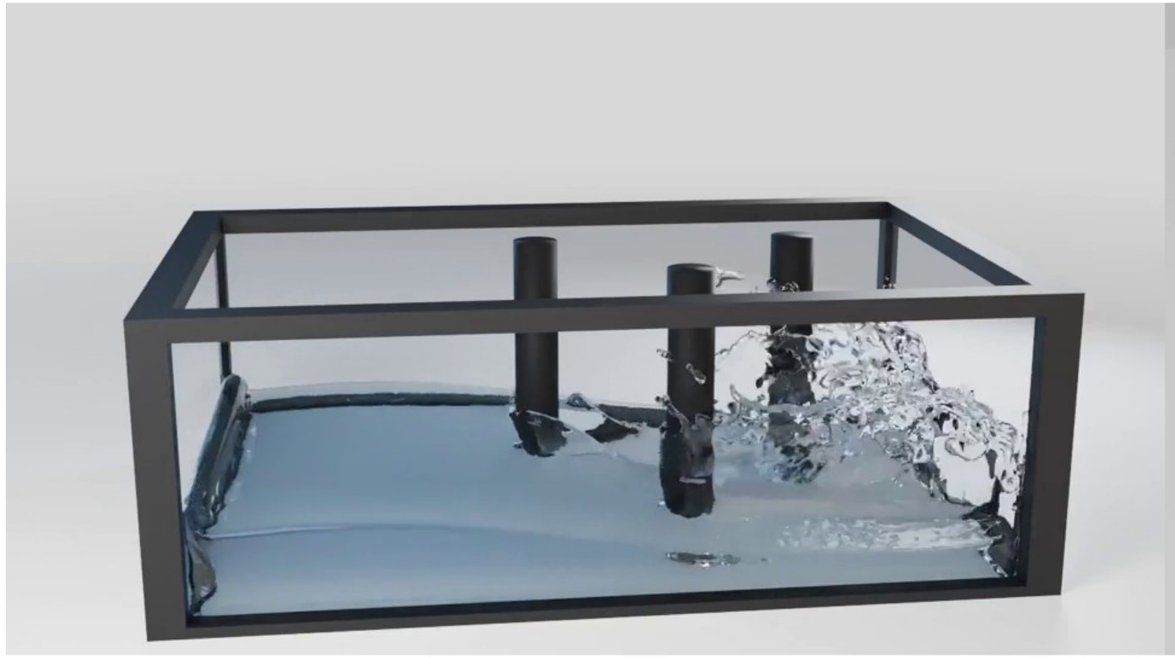
```
h += C_mr[m * SPEC_NR + ir] * Math.cos(m * theta);
const yHeight = Math.max(0.0, h + 1.4);
```

Hand-authored, cosine-only height field: reflection symmetry; no overhangs.

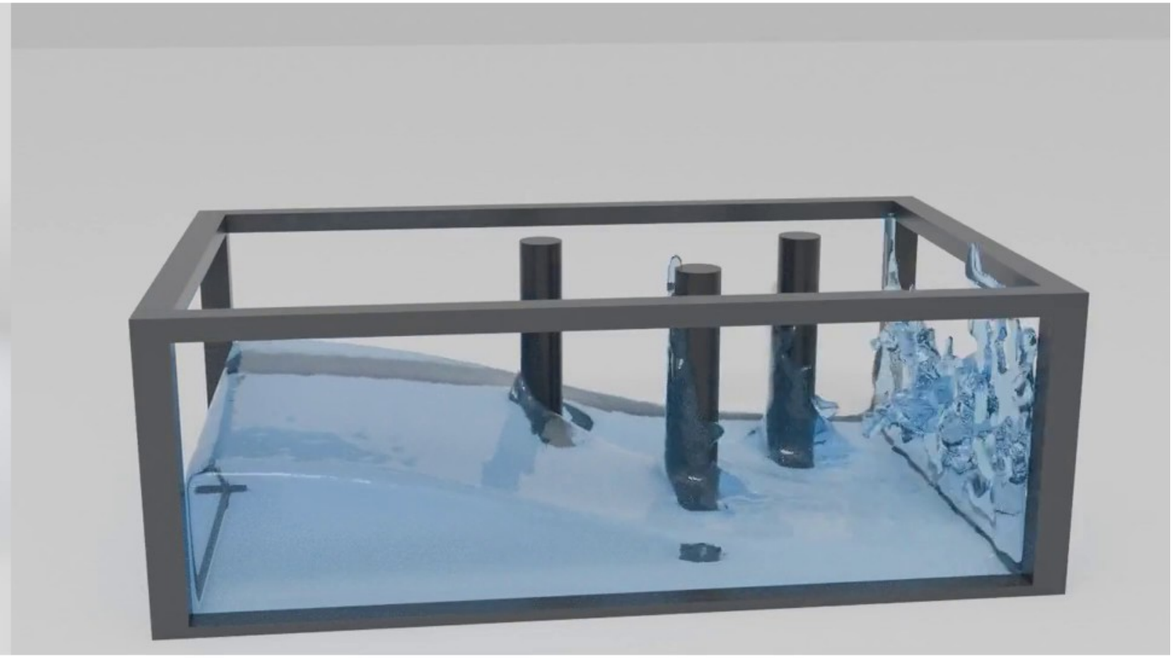
Supplied renderer · Mathematical definitions and implementation details in speaker notes

SceneAgent: reconstructing a dynamic scene

Reference video · B_09



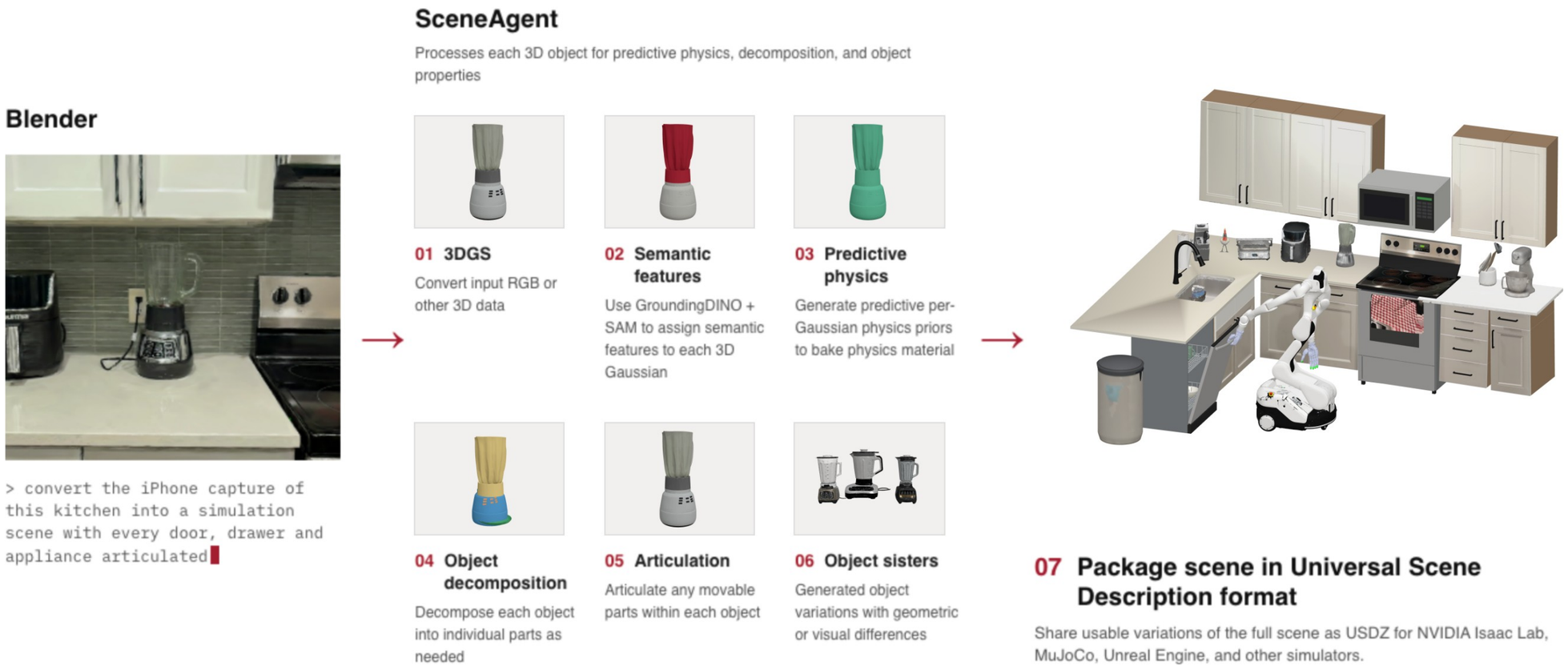
SceneAgent reconstruction



What do specialized tools add to a general coding agent?

SceneAgent authors · Official project videos · B_09 liquid dam break

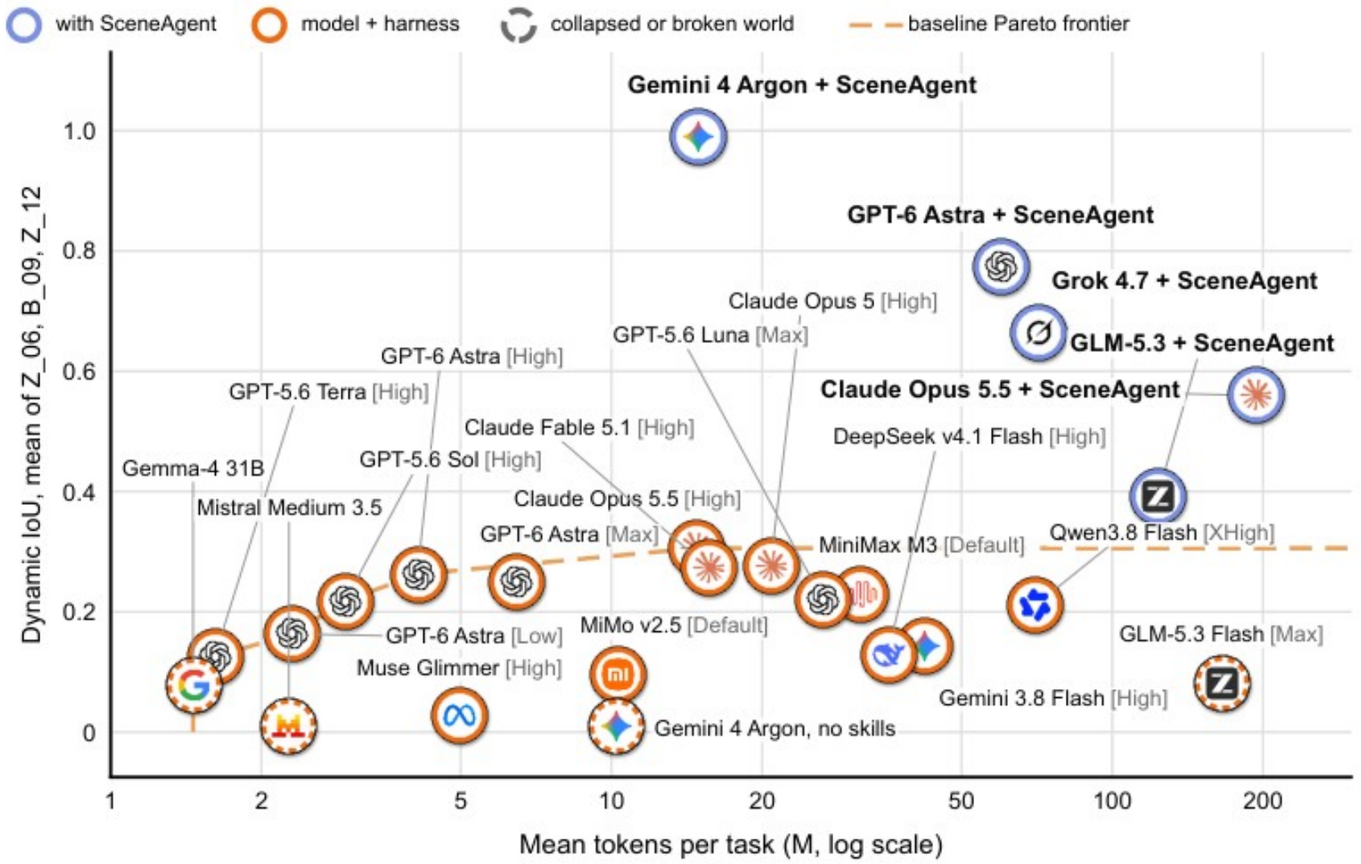
SceneAgent: inspect and repair the intermediate scene



Where can the agent observe a wrong scale, part, or articulation?

SceneAgent · Original Fig. 2 · Capture-to-simulation pipeline

SceneAgent: a large gain on three dynamic scenes



Gemini 4 Argon

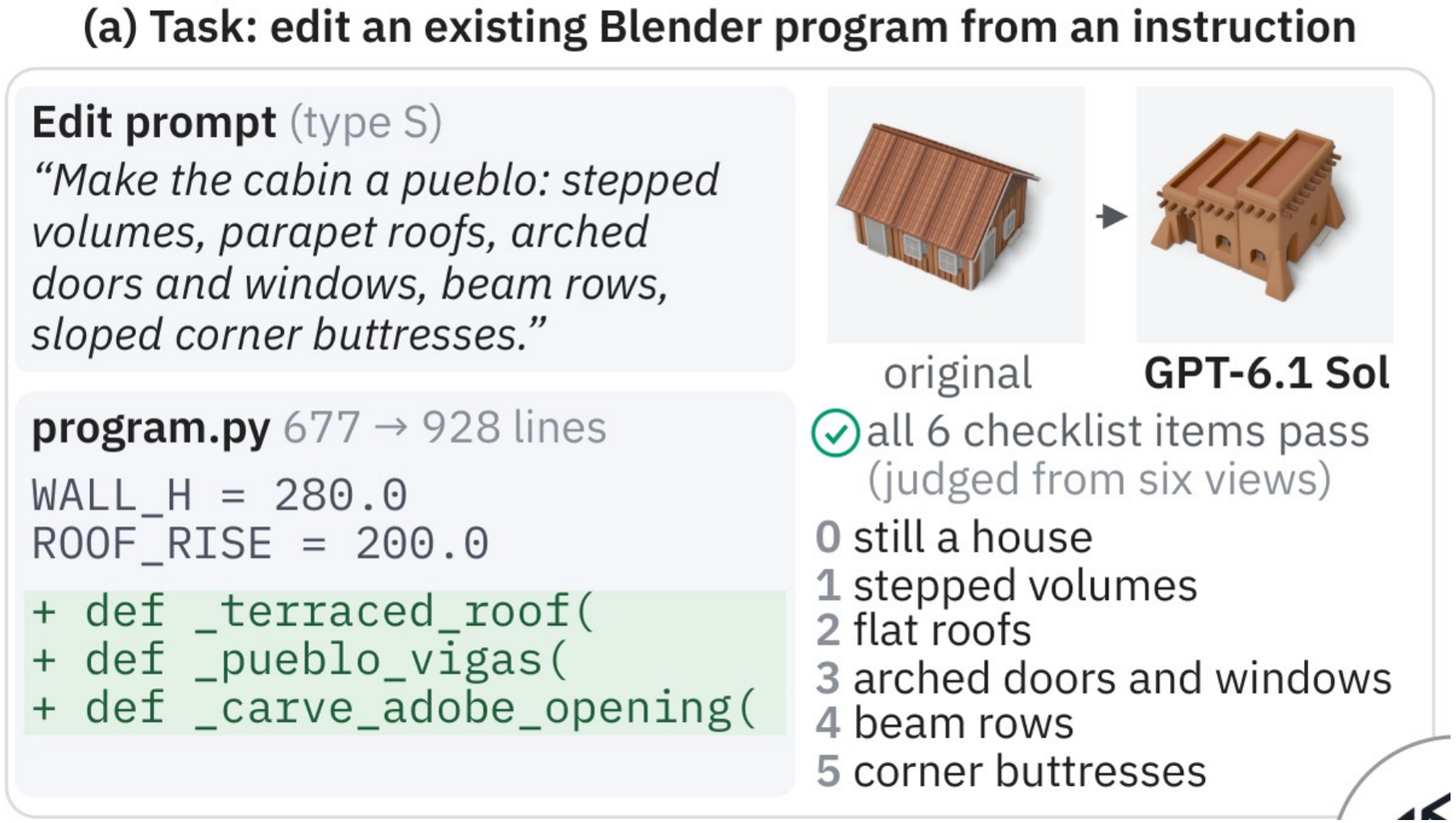
Dynamic mask IoU: 0.86% → 99.09%

Scope

Three synthetic scenes; one run per configuration and condition.

What evidence would distinguish reconstruction from mechanism recovery?

3DCodeEdit: edit the program, check the geometry



2,592 tasks · 108 programs

Six matched views

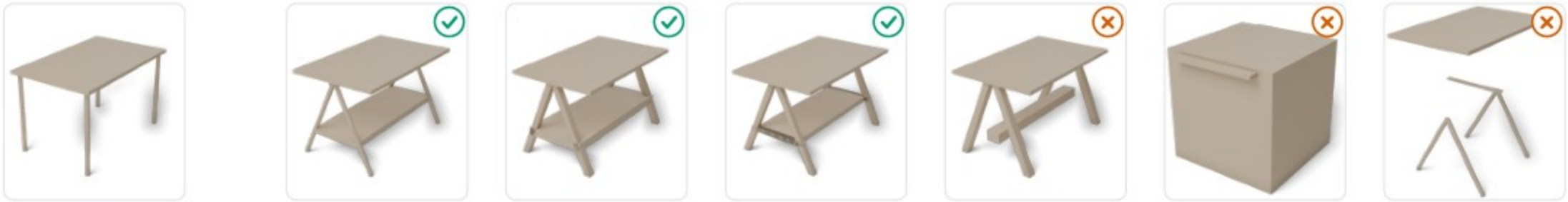
The target is an instruction and a checklist, not one reference mesh.

3DCodeEdit · Official website teaser, Fig. 1(a) · Task panel crop

3DCodeEdit: runnable code can build the wrong shape

Structural edit: two A-frame supports with a thick shelf between them

G4 Structural edit *"Rebuild the legs as two A-frames under the short ends that widen to far-apart feet; span a thick shelf between their lower halves, high above the floor, with a large triangular opening above it."*



	original	Claude Opus 5.5	GPT-6.1 Sol	Gemini 4 Argon	Gemini 3.1 Pro *thin rail, no shelf*	Claude Haiku 4.5 *built a solid box*	gpt-oss-120b *parts float apart*
0 still the same desk		✓	✓	✓	✓	✗	✗
1 two A-frame legs		✓	✓	✓	✓	✗	✗
2 A-frames widen to feet		✓	✓	✓	✓	✗	✓
3 thick shelf between legs		✓	✓	✓	✗	✗	✗
4 triangle above shelf		✓	✓	✓	✗	✗	✗
5 shelf well above floor		✓	✓	✓	✓	✗	✗

Which feedback would catch a missing shelf or detached support?

What does code add to a 3D representation?

Which spatial relationships should become explicit?

What can we reason about once geometry becomes executable?

Can a machine discover its own language of 3D?

What makes an abstraction useful beyond its original examples?

Can a representation evolve when the questions we ask change?

Discussion · Document review, Q3, Q10, Q16, Q40

Spectral representations need a decoding domain

Specify the decoder	Coefficients + basis + domain; count all dependencies.
State the invariance	Coordinate coefficients are not automatically rotation invariant.
Keep the roles distinct	Descriptors, functional maps, and generative shape codes differ.

A shared-domain deformation code is a precise starting point

Gaussian scenes to reusable construction rules

Proposed representation project



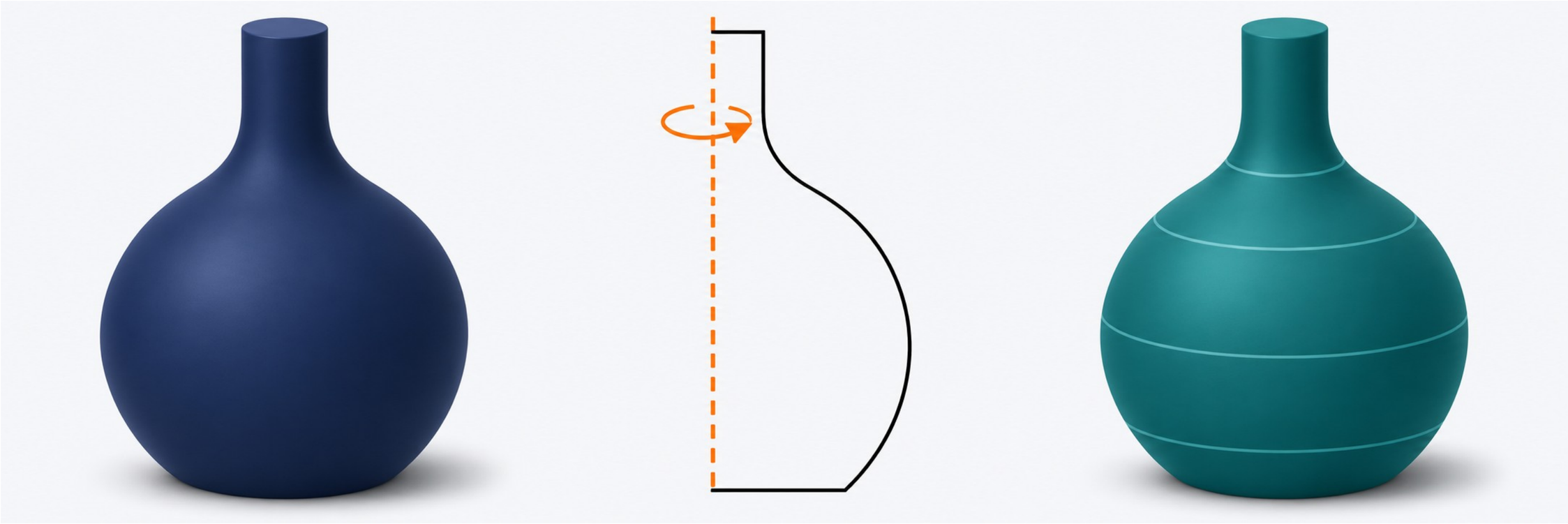
Captured evidence **Candidate rule** **New variant**

Measure stable behavior rather than identical source text

Concept illustration · AI-generated · Proposed experiment

A controlled test of 2D intermediates

Proposed diagnostic experiment



Observed view | **Canonical profile** | **Revolved model**

Hold observations fixed; use the oracle to diagnose the bottleneck

Nearby work defines the contribution

Prior work	Sharper question
Templates · BeCoS	Persistence under structural edits
ParSEL	Recover dependencies from incomplete evidence
ShapeCoder	Stable rules from noisy reconstructions
3DCodeEdit · 4DCodeBench	Executable checks · calibrated interventions

Read each result within its evaluation scope

4DCodeBench	Normalized metric families; geometry validity is a separate diagnostic.
SceneAgent	Dynamic image-mask IoU; three synthetic cases; one run per condition.
3DCodeVerse	Compilation and execution differ from geometric fidelity.
Controlled comparisons	Match observations, tools, task families, and cost accounting.

The 40-question research map

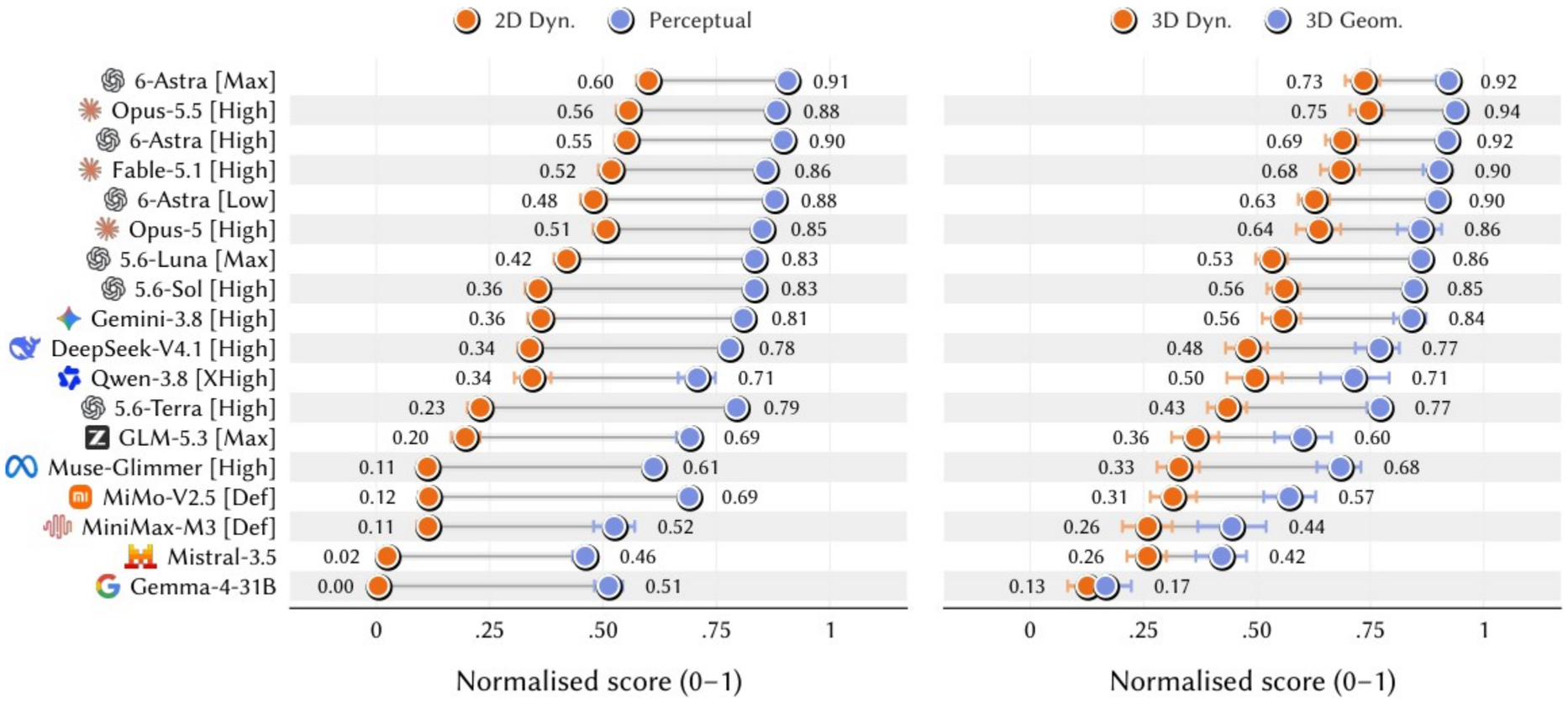
Q1–8	Meaning and abstraction · language and geometric grounding
Q9–16	Program interfaces · representations and intermediates
Q17–24	Hybrid detail · inference and active observation
Q25–32	Spectra and correspondence · Gaussian program recovery
Q33–40	Physical mechanisms · verification and learning

Numbers refer to the 40-question collection

Reading and discussion sources

Main paper	4DCodeBench · arXiv:2610.03715
Companions	SceneAgent · 2610.34745; 3DCodeEdit · 2610.38231
Broader landscape	3DCodeVerse · 2610.22341; InfiniDream · 2610.26853
Research directions	Document review; references in each slide’s notes

Static–dynamic gap: separate image and 3D comparisons



Separate comparisons

The finding is not confined to one aggregate comparison