COVERAGE OPTIMIZATION FOR CAMERA VIEW SELECTION

What makes a *good* viewpoint?

Timothy Chen* Adam Dai* Maximilian Adang Grace Gao Mac Schwager

STANFORD UNIVERSITY · * EQUAL CONTRIBUTION

chengine.github.io
/nbv_gym

§ 01 — THE SETUP

3D Reconstruction needs *informative* images of the scene.

Reconstruction quality is gated by the data — not the optimizer. Choose the next view well, and the reconstruction sharpens. Choose it poorly, and the model burns compute on redundant geometry and slows down downstream processes.

The active view selection problem: pick the camera pose that the model expects to learn most from, before even getting there.

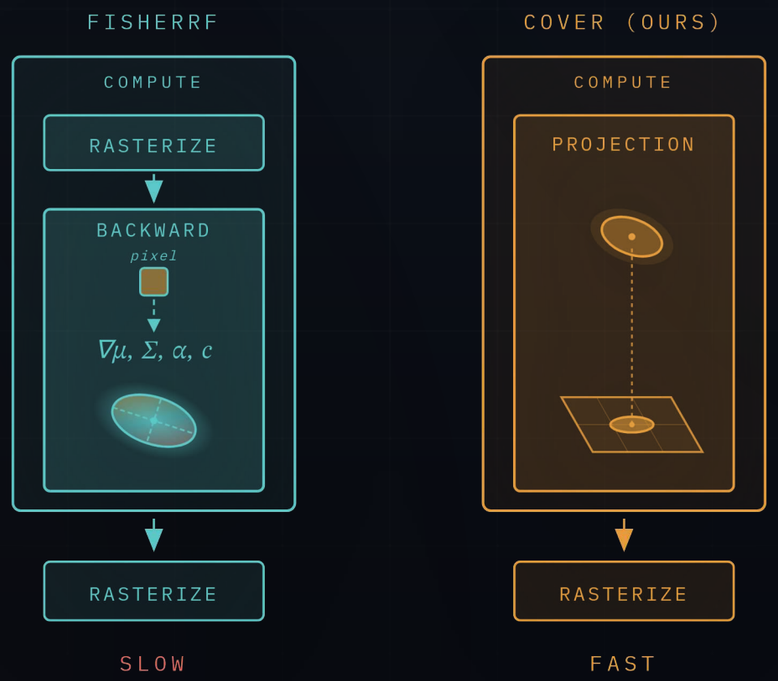
§ 02 — THE GAP

Information theory is the principled answer — but it's *expensive*.

FisherRF ranks views by Fisher Information Gain, but every candidate requires a **backward pass** through the renderer to compute gradients.

Random sampling is fast, but myopic and unprincipled — it leaves SSIM and PSNR on the table.

What we want: a forward-only metric — principled, fast, interpretable, non-myopic.



§ 03 — FISHER INFORMATION GAIN

Information theory, distilled into *geometry*.

FISHER INFORMATION GAIN

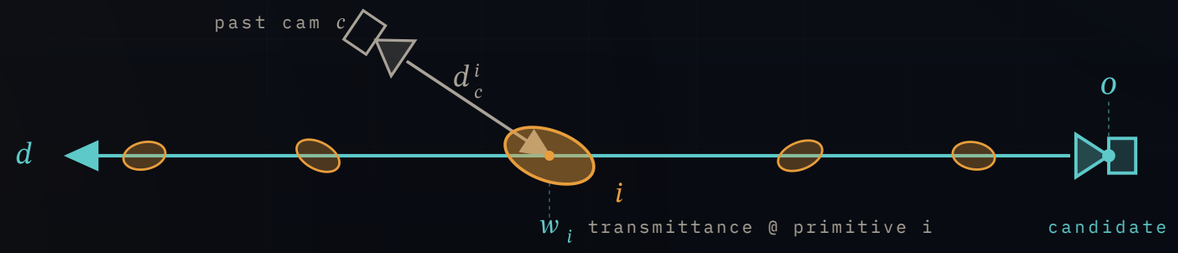
$$\mathrm{FIG}(w; W) \;=\; \log|G + ww^{\top}| - \log|G| \;=\; \log(1 + w^{\top} G^{-1} w)$$

$$G = W^{\top} W$$

FISHER MATRIX

COVERAGE METRIC

$$\mathrm{COVER}(o, \boldsymbol{d}) \;=\; \tfrac{1}{2} \sum_i w_i(o, \boldsymbol{d}) \; (1 + \max_c \boldsymbol{d}_c^{\,i} \cdot \boldsymbol{d})$$



“Take the picture that shows what your old pictures missed.”

§ 04 — WHAT COVERAGE ENCODES

Fill the *sphere of viewing directions*.

Every **Gaussian** carries a **sphere of viewing directions** — all the angles it could be observed from.

Each **TRAINING** camera lights a **patch** on that sphere — every direction the Gaussian has been seen from.

The best **TEST** camera that lands in the largest gap — the **most uncovered** direction — minimizes the coverage score.

The full metric, in one line.

$$\mathrm{COVER}(o, \boldsymbol{d}) \;=\; \tfrac{1}{2} \sum_i w_i(o, \boldsymbol{d}) \, \left(1 + \max_c \boldsymbol{d}_c^{\,i} \cdot \boldsymbol{d}\right)$$

Rendered like any other pixel: a transmittance-weighted sum over primitives i. Each primitive's score is the ***best dot product*** *between any past viewing direction $\mathbf{d}^i_c$ and the candidate $\mathbf{d}$ — clamped to [0, 1].*

1 **Projection.** No rasterization, no transmittance computation. Just project Gaussian centers.

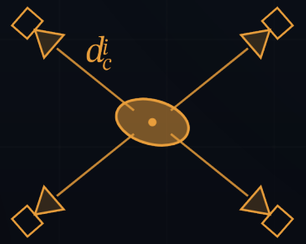
2 **Tracking.** For every primitive, accumulate incident viewing directions from past training cameras.

3 **Gaussian coverage attribute.** A boolean grid on the unit sphere per Gaussian — flip a bit each time a camera observes it.

4 **Rendering.** Rasterize the per-Gaussian coverage into a coverage image.

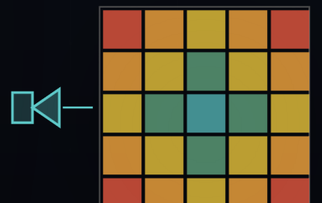
1 · PROJECTION

2 · TRACKING



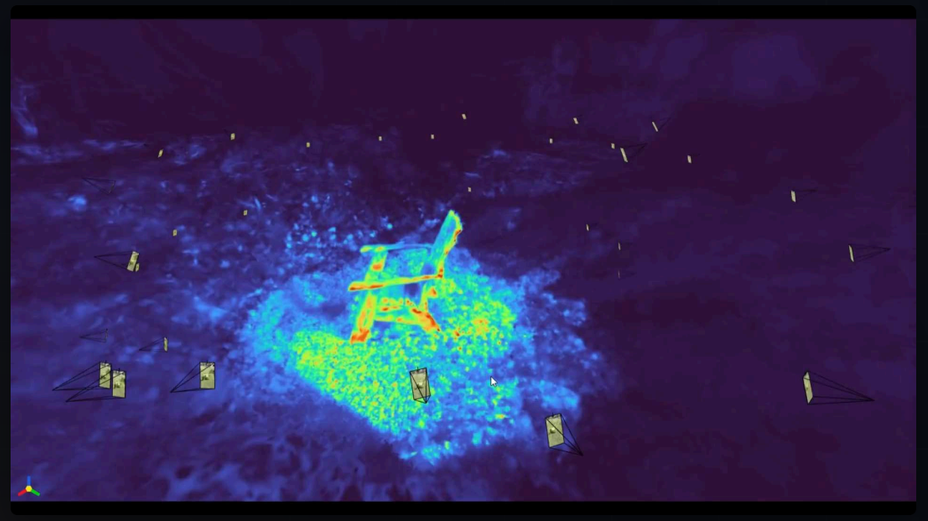
3 · STORAGE

4 · RENDERING

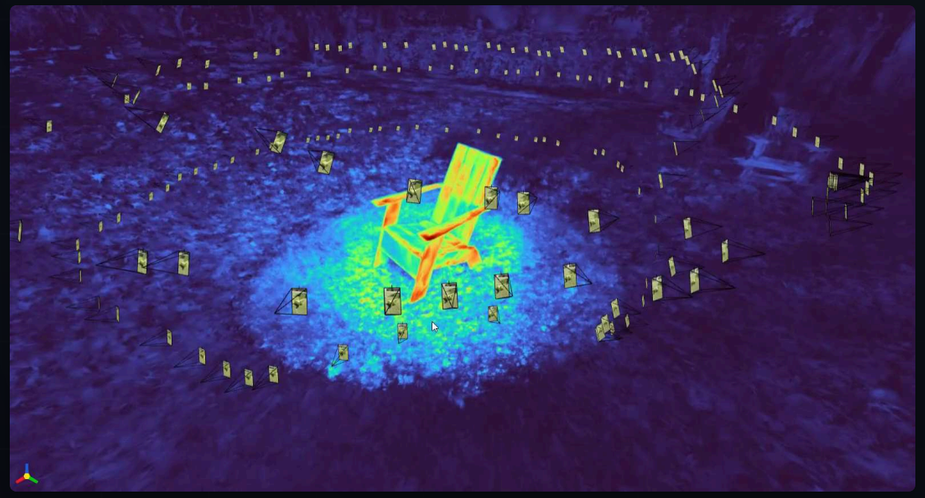


§ 06 — THE METRIC, MADE VISIBLE

Coverage renders as an *image*. See how it evolves and behaves.



TRAINING



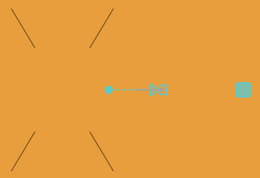
SPATIAL VARIATION

Cool blues *mark under-observed regions;* warm reds *mark well-covered ones. The next-best view is the one that paints the most* blue *into* red.

§ 07 — HOW COVER RUNS

Three lightweight stages, looped while training.

Coverage View Metric



Bin each training camera into a directional boolean grid on every Gaussian. Renderable at 85 FPS for live visualization.

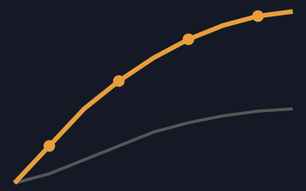
02 · SELECT

Next Best View

Rank candidates by their coverage score. Pick the argmin. Add it to the training set.

03 · TRAIN

Incremental Training



Train the 3DGS on the augmented dataset for several iterations. Then return to stage 1.

§ 08 — REAL-TIME, BY DESIGN

Fast enough to render *as an image*.

85

FRAMES PER SECOND · COLOR INCLUDED

COMPUTE METRIC

Forward only

RENDER METRIC

Standard rasterization

VISUALIZATION

Live in-viewer

Computing the coverage score for a candidate camera needs **only Gaussian projection** — no transmittance, no gradients. **Rendering** the score as a heatmap still uses the regular rasterizer (we just append one channel to the color pipeline), and it runs at 85 FPS.

§ 09 — 15 SCENES · 4 DIFFERENT METHODS · STATIC DATASET

COVER wins on *reconstruction fidelity*.

Tanks & Temples · MipNeRF360 · custom captures. Averaged across all scenes after 30K gradient steps.

PER-DATASET BREAKDOWN

Tanks & Temples

	PSNR ↑ HIGHER	SSIM ↑ HIGHER	LPIPS ↓ LOWER
COVER	**21.52**	**0.76**	**0.19**
RANDOM	21.12	0.75	0.20
FISHERRF	20.26	0.71	0.23
BAYES' RAYS	15.03	0.43	0.62

Mip-NeRF360

COVER	**25.78**	**0.78**	**0.17**
RANDOM	25.57	0.78	0.17
FISHERRF	24.92	0.78	0.19
BAYES' RAYS	18.81	0.45	0.63

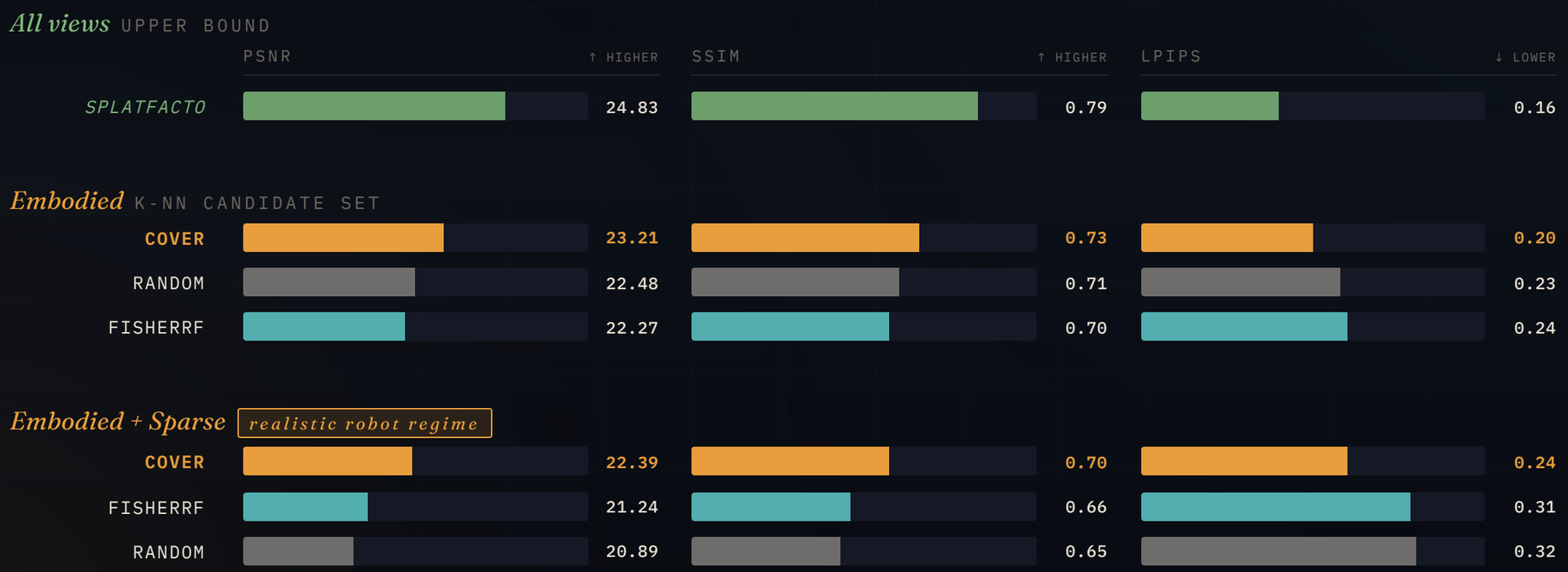
Captures — custom

COVER	**19.05**	**0.60**	**0.25**
RANDOM	18.71	0.58	0.27
FISHERRF	17.94	0.54	0.33
BAYES' RAYS	13.40	0.27	0.82

§ 11 — EMBODIED · THE ROBOT REGIME

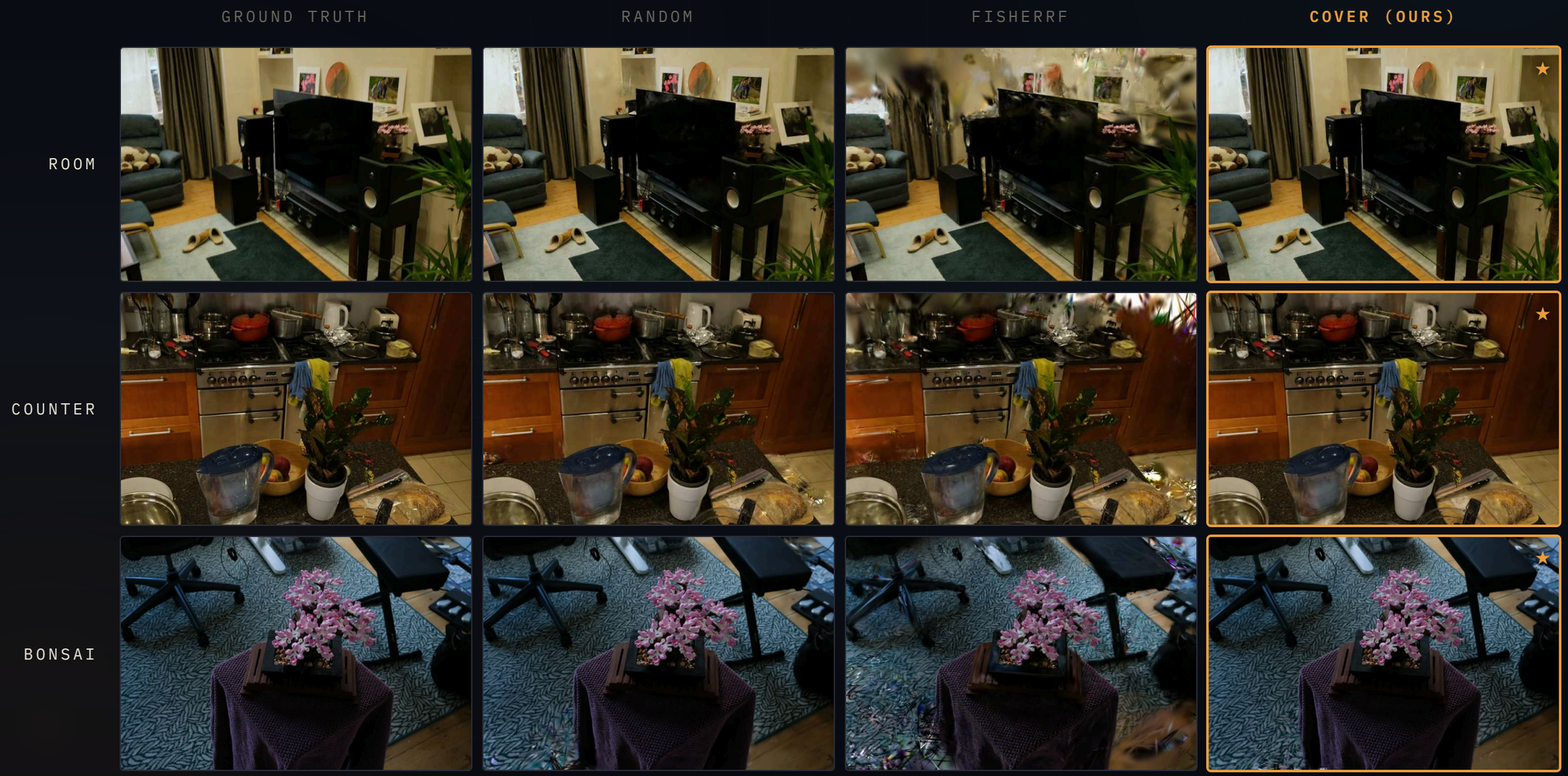
When the camera can't teleport, the gap *widens*.

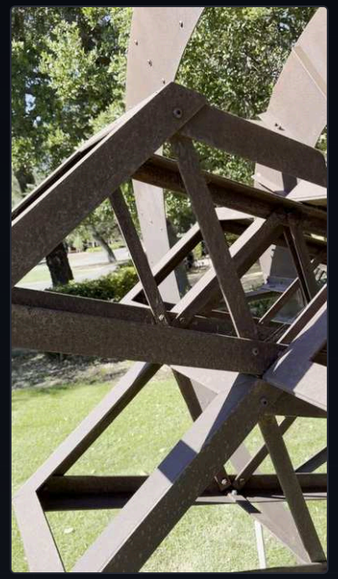
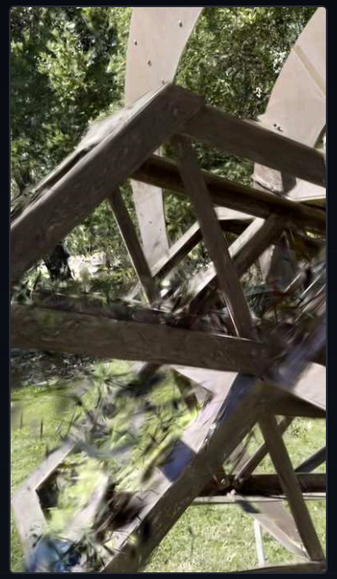
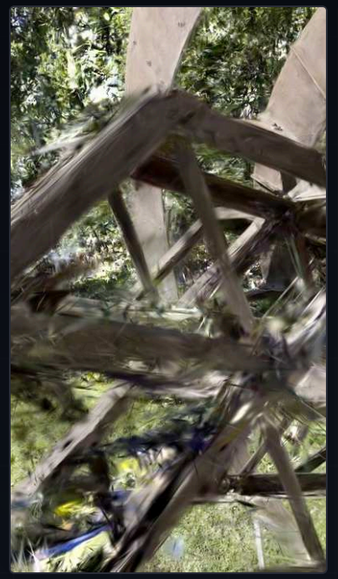
We simulate continuous deployment: at each step the selector can only choose among k-NN cameras near its current pose.



§ 10 — QUALITATIVE

A *visually attractive* view metric.



	GROUND TRUTH	RANDOM	FISHERRF	COVER (OURS)
SPACE				
CHAIR				

§ 14 — CLOSING

A simpler answer is also a *better* one.

01 · PRINCIPLED

Derived from FIG, reduces to coverage.

The information-theoretic objective can be simplified to storing a directional boolean grid per primitive.

02 · PRACTICAL

Projection-only · 85 FPS · live.

No transmittance, no rasterization. The metric is interpretable and renderable.

03 · PERFORMANT

Beats FisherRF + Random across 15 scenes.

Higher PSNR / SSIM, lower LPIPS, on both human-captured and embodied data acquisition settings.

COVER · CVPR 2026

What makes a good viewpoint? One that covers what you haven't seen.

Timothy Chen* Adam Dai* Maximilian Adang Grace Gao Mac Schwager

STANFORD UNIVERSITY

chengine.github.io/nbv_gym

chengine.github.io
/nbv_gym