GROW WHERE VIEWS AGREE: EFFICIENT SPARSE-VIEW 4D RECONSTRUCTION WITH GI-ACE

Anonymous authors
Paper under double-blind review

ABSTRACT

With only four cameras, a dynamic Gaussian-splatting reconstruction sees each surface from few directions, yet densification still adds primitives wherever the pooled image-space gradient is large. We argue that this signal conflates two quantities: how strongly a primitive asks to change, and how many independent viewpoints make the request. A residual that only one camera sees can be removed by fitting that camera, so it is weak evidence that new scene structure is needed. We introduce **GI-ACE** (Geometry-Informed Asynchronous Camera Evidence), an admission rule for 4D Gaussian densification that uses the per-camera gradient statistics ordinary training already produces. A primitive grows if its demand survives removing its strongest camera, or, for a large primitive, if two cameras jointly request a split from sufficiently different directions, measured at the times each camera observed it. Both tests reuse the native threshold and add no learned component, point budget, or rendering pass. On the six-scene Neural3DV four-camera benchmark, GI-ACE exceeds, to our knowledge, all published results, improving mean PSNR from 22.29 to 23.04 dB over the strongest baseline, and in our reproductions it uses 2.9–3.7$\times$ fewer Gaussians while training 1.6–2.0$\times$ faster than the same backbone under native densification. On six Panoptic Sports sequences it improves mean PSNR by 0.45 dB with an order of magnitude fewer Gaussians and 5.4$\times$ faster training. Controlled studies confirm the design: the population stays stable under biased camera sampling, temporally staggered capture further improves the same rule, and the geometric route activates precisely where complementary views exist.

1 INTRODUCTION

Four synchronized cameras are enough to capture a dynamic scene, and recent sparse-view methods have made such captures reconstructible (Zhou et al., 2026). The spatial evidence in four views is scarce and uneven, though: some surfaces are seen clearly by one camera only, many frames revisit the same directions, and motion changes which views constrain each scene element. Gaussian splatting exposes how a reconstruction spends this evidence, because the scene grows explicitly by cloning or splitting primitives whose projected position gradient exceeds a threshold (Kerbl et al., 2023). Every admitted primitive then takes part in all later rendering, backpropagation, and optimizer updates, so a growth decision costs computation for the rest of training.

Under sparse supervision this growth rule is poorly matched to the question it has to answer. The pooled gradient measures how strongly a primitive asks to change, but not how many independent viewpoints make the request. A residual concentrated in one camera, such as a view-dependent highlight or a surface seen at a grazing angle, can be reduced by fitting that camera with existing primitives, and repeated frames from the same camera keep the gradient high without adding a new direction. Native densification cannot tell such a request from one corroborated across views. The effect is large: on the Panoptic Sports Football sequence, the same backbone under native densification grows to 3.2M Gaussians, while a model with 10$\times$ fewer primitives scores 2 dB higher on held-out views (Section 4).

We therefore ask **which residuals deserve new capacity**, and answer with the camera structure of the evidence. Two principles guide the answer. *Robustness*: a growth request should survive losing its most supportive camera, much as a model-selection decision should survive leaving out one data

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

source. *Parallax*: support need not be broad to be genuine. Fine detail may be visible to only two cameras, and two views that see a point from well-separated directions constrain its 3D position, whereas two nearly parallel views add little beyond one. A dynamic scene adds a further difficulty. Cameras are sampled at different times, so their directions must be measured where the moving primitive was when each camera observed it.

GI-ACE turns these principles into an admission rule inside the native topology step (Figure 1). From the ordinary shuffled camera–time minibatches it accumulates, for every primitive and camera, the mean gradient demand and a gradient-weighted observation time. A *robust core* accepts a native proposal only if the mean demand of the remaining cameras still reaches the threshold after the strongest camera is removed. For large primitives the core rejects, a *geometric route* recovers a split when some camera pair has enough demand after weighting it by the sine of the angle between their rays, cast to the primitive's time-conditioned 4D centers. Both tests are expressed in the units of the native threshold, so the rule has no new hyperparameter, learned component, point budget, or extra rendering pass, and the final population follows from the evidence.

On the published four-camera Neural3DV benchmark, GI-ACE improves mean PSNR from 22.29 to 23.04 dB over the strongest prior method and lowers all reported structural metrics. The published benchmark reports quality but not model size or training time, so efficiency is measured against our reproductions of the same backbone under native densification: GI-ACE uses 2.9–3.7$\times$ fewer final Gaussians and trains 1.6–2.0$\times$ faster, at quality above the published result in every scene. The gains are not produced by simply growing less: at matched final populations, choosing recovery candidates randomly instead of by the geometric score loses up to 0.29 dB (Section 4). On six Panoptic Sports sequences, the same rule reduces the final model by an order of magnitude and the training time by 5.4$\times$, while raising mean PSNR and lowering every structural metric.

Our contributions are:

- **Densification as evidence-based admission.** We identify single-camera demand as a main source of wasted growth under sparse views and propose a strongest-camera-removal test that prevents any one camera from authorizing new primitives.
- **Time-conditioned parallax for local refinement.** We recover localized splits supported by two cameras using demand-weighted parallax, measured at the primitive's 4D positions at each camera's observation times, so the rule works with asynchronous sampling. The recovery provably preserves the core's decisions and needs no new parameter.
- **A better quality–efficiency frontier.** GI-ACE exceeds, to our knowledge, all published four-camera results on Neural3DV while using 2.9–3.7$\times$ fewer Gaussians and training 1.6–2.0$\times$ faster than our reproductions of the same backbone, and on Panoptic Sports it improves mean PSNR while using an order of magnitude fewer Gaussians and training 5.4$\times$ faster than the released code. Controlled studies identify when camera-aware admission helps and when it fails.

2 Related Work

Dynamic Gaussian scene representations. 3D Gaussian Splatting couples explicit anisotropic primitives with differentiable rasterization and adaptive density control (Kerbl et al., 2023). Dynamic extensions span deformation-field models such as 4DGaussians (Wu et al., 2024), spacetime feature splatting (ST-GS) (Li et al., 2024b), fully explicit keyframe interpolation in Ex4DGS (Lee et al., 2024), and native 4D Gaussian primitives with temporally conditioned geometry (Yang et al., 2024). These methods primarily design the temporal representation and motion model. GI-ACE instead operates on the discrete allocation of primitives during learning and retains the backbone's representation and renderer.

Sparse-view reconstruction. For static scenes, sparse-view Gaussian methods add geometric regularization such as depth priors or proximity-guided unpooling (Li et al., 2024a; Zhu et al., 2024). For dynamic scenes, 4C4D evaluates the above dynamic Gaussian families under a common four-camera protocol and improves their geometry through neural opacity decay and visibility-dependent handling (Zhou et al., 2026). We build on this backbone and address a different stage: opacity

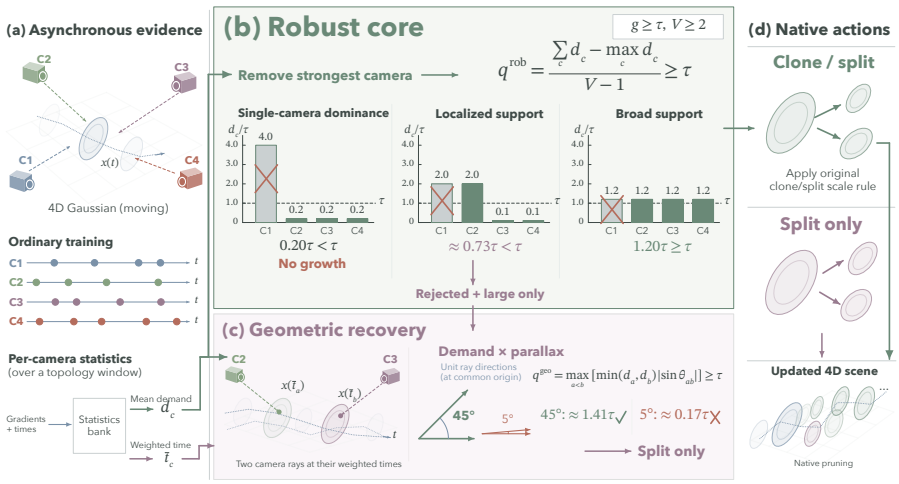

Figure 1: Overview of GI-ACE. (a) Camera-specific demand and gradient-weighted observation times are accumulated from ordinary, potentially asynchronous training batches. (b) The robust core removes the strongest camera contribution and accepts a native proposal only when the remaining-camera mean still reaches the native threshold τ. (c) For rejected large proposals, geometric recovery admits a split only when two cameras provide both sufficient demand and non-degenerate parallax at their representative observation times. (d) Accepted proposals use the backbone's native clone, split, pruning, and reset operations; GI-ACE introduces neither an extra rendering pass nor a point budget.

learning regulates existing primitives, while GI-ACE determines which proposed additions receive capacity.

Densification and compact Gaussian optimization. Several works improve the signal or operators used by density control. AbsGS addresses gradient collision with a homodirectional criterion (Ye et al., 2024), Pixel-GS weights gradient aggregation by projected pixel coverage (Zhang et al., 2024), and Deng et al. develop an edge-aware score, a long-axis split, and overfitting controls (Deng et al., 2025). Other methods treat population as an explicit resource: Revising Densification and 3DGS-MCMC reformulate growth with error-based or probabilistic relocation under a target count (Rota Bulò et al., 2024; Kheradmand et al., 2024), Taming 3DGS schedules growth toward a budget (Mallick et al., 2024), and LightGaussian prunes and compresses a trained model (Fan et al., 2024). These methods pool image-space statistics over all views or require a population target. GI-ACE uses the camera structure of sparse dynamic supervision instead, and its final population follows from the measured evidence.

Toward compact explicit dynamic scenes. Dynamic capture has long been a studio technology, with volumetric rigs relying on tens to hundreds of synchronized cameras (Joo et al., 2015; Guo et al., 2019); sparse-view reconstruction removes that barrier, and the explicit scene it produces is the state downstream systems store, stream, and simulate. Text-to-4D generation already produces dynamic scene assets (Bahmani et al., 2024), and recent generative approaches learn to simulate interactive environments and video dynamics (Bruce et al., 2024; Brooks et al., 2024); whether such models can be grounded in explicit, editable scene state is open, and a compactly reconstructed scene is a plausible interface. For these uses the population is part of the asset cost, so where capacity is spent matters as much as how much is available.

3 GI-ACE

3.1 Reconstruction setting and the growth decision

We observe a dynamic scene using calibrated cameras with centers o_c, where $c \in \{1, \dots, C\}$. Each training observation contains an image $I_{c,t}$ and its timestamp t. Our scene consists of N 4D Gaussian primitives. Primitive i has a spatial center x_i, temporal center μ_i, a spacetime covariance Σ_i, opacity, and appearance coefficients. Its spatial center conditioned on time is

$$x_i(t) = x_i + v_i(t - \mu_i), \qquad v_i = \frac{\Sigma_{i,xt}}{\Sigma_{i,tt}}. \tag{1}$$

Rendering produces $\widehat{I}_{c,t}$. We use the backbone's photometric objective,

$$\mathcal{L} = (1 - \lambda)\|\widehat{I}_{c,t} - I_{c,t}\|_1 + \lambda\big(1 - \mathrm{SSIM}(\widehat{I}_{c,t}, I_{c,t})\big), \tag{2}$$

together with its opacity learning and continuous parameter optimization. GI-ACE changes only the admission of topology proposals.

Let g_i denote the backbone's accumulated projected position-gradient score and let τ be its densification threshold. A native proposal satisfies $g_i \geq \tau$. The existing scale rule chooses whether that proposal clones a small primitive or splits a large one. We write $\ell_i = 1$ for the large-primitive case. GI-ACE evaluates evidence before these operations and passes the accepted proposals to the same operators.

3.2 Camera evidence from ordinary training

The evidence unit is a camera, with its observations collected over the current topology window. We preserve the original shuffled image minibatches; cameras can be sampled at different timestamps and need not appear together in a batch.

For a visible observation q of primitive i, let a_{iq} be the norm of its two-dimensional projected position gradient, as obtained from the usual backward pass. Within each camera, we accumulate gradient magnitude, observation count, and a gradient-weighted time moment. The camera demand and representative observation time are

$$d_{ic} = B\frac{\sum_{q \in W_{ic}} a_{iq}}{|W_{ic}|}, \qquad \bar{t}_{ic} = \frac{\sum_{q \in W_{ic}} a_{iq} t_q}{\sum_{q \in W_{ic}} a_{iq}}. \tag{3}$$

Here W_{ic} contains the visible observations from camera c in the window. The factor B is the minibatch size: per-image losses are divided by B before backpropagation, so this multiplication restores the native gradient scale. An unobserved camera has zero demand and is excluded from the observed-camera count V_i. When its accumulated gradient is zero, the representative time defaults to μ_i.

A camera with many observations contributes its mean demand rather than a vote for every frame. This separates the amount of sampling within a camera from the number of cameras supporting growth. The time moment records when that camera generated its demand, using the same observations and gradient weights. All statistics are refreshed through the native topology-window reset.

3.3 A robust core for representation growth

We first ask whether a proposal still has sufficient support after its strongest camera is removed. For $V_i \geq 2$, define

$$q_i^{\mathrm{rob}} = \frac{\sum_c d_{ic} - \max_c d_{ic}}{V_i - 1}. \tag{4}$$

This is the smallest remaining-camera mean obtained by removing any one observed camera, so the test has an immediate interpretation: a proposed addition must withstand losing its most supportive camera.

The robust core accepts a primitive when it is a native proposal, is observed by at least two cameras, and satisfies $q_i^{\mathrm{rob}} \geq \tau$. We denote this decision by A_i^{rob}. Each accepted primitive receives the native clone or split action according to its scale.

Consider demands of $(4.0, 0.2, 0.2, 0.2)\tau$. Their average is above the native threshold, yet removing the first camera leaves only 0.2τ, so the robust core rejects this pattern; it accepts $(1.2, 1.2, 1.2, 1.2)\tau$. The policy thus distinguishes concentrated demand from demand sustained across cameras at the same threshold scale.

3.4 Recovering local detail with geometric evidence

The robust core provides a selective default, but local refinement can be supported by a narrower subset. Suppose a large primitive has demands $(2.0, 2.0, 0.1, 0.1)\tau$. Removing the strongest camera and averaging the remaining three yields approximately 0.73τ. The core rejects it even though two cameras strongly request refinement.

The remaining question is whether those two cameras provide complementary directions. For each camera, we evaluate the primitive at the time represented by its evidence and construct the unit observation ray,

$$r_{ic} = \frac{x_i(\bar{t}_{ic}) - o_c}{\|x_i(\bar{t}_{ic}) - o_c\|_2}. \tag{5}$$

Using $x_i(\bar{t}_{ic})$ ties each direction to the portion of the primitive's motion that generated the camera's demand, which matters when cameras are sampled at different times. Because the conditional trajectory in Equation 1 is affine in time, this position is exactly the gradient-weighted mean conditional position under the current parameters.

We score every camera pair by its weaker demand and angular separation:

$$q_i^{\mathrm{geo}} = \max_{a<b} \left[\min(d_{ia}, d_{ib}) \, \|r_{ia} \times r_{ib}\|_2 \right]. \tag{6}$$

The minimum requires both cameras to contribute. The cross-product norm is the sine of their ray angle, discounting parallel and antiparallel directions and reaching one for orthogonal directions. The maximum selects the pair that best combines demand and directional complementarity; selecting the strongest demands first would miss a less demanding pair with substantially better geometry.

Equation 6 retains gradient units, so the same τ determines whether a geometric refinement is admitted, with no additional angle cutoff or weighting coefficient. For a fixed pair with angle θ, admission requires its weaker demand to reach $\tau/|\sin\theta|$: well-separated directions need little excess demand, while nearly collinear directions need much more. In the preceding example, a 45° supporting pair scores 1.41τ and recovers the split, whereas a 5° pair scores only 0.17τ.

We use this score to recover **large native proposals rejected by the robust core**. This targets the operation that increases local resolution, splitting a large parent into smaller children, and keeps localized pair evidence from becoming a general route for duplicating small primitives. The final policy is

$$A_i = A_i^{\mathrm{rob}} \ \vee \ \left[g_i \geq \tau \ \wedge \ \neg A_i^{\mathrm{rob}} \ \wedge \ \ell_i = 1 \ \wedge \ V_i \geq 2 \ \wedge \ q_i^{\mathrm{geo}} \geq \tau \right]. \tag{7}$$

The robust route retains every accepted core clone and split, and the geometric route adds only residual splits supported by a suitable pair. Opacity pruning then proceeds normally.

3.5 Properties

GI-ACE treats growth as a local model-selection decision with two forms of support: demand robust to removing a dominant camera, or localized split demand supported by complementary rays. A uniform increase of the gradient threshold cannot express this policy, because two primitives can have the same native score and different camera-demand distributions, and two residual proposals can have the same camera demands and different angular support.

Several properties follow from the construction (Appendix A). Every robust-core action is preserved on the current state. A single camera with nonzero demand cannot authorize growth through either

route, and collinear supporting rays cannot recover a rejected proposal. If $d_i^{(2)}$ is the second-largest camera demand,

$$0 \le q_i^{\mathrm{geo}} \le d_i^{(2)}, \tag{8}$$

so geometric recovery is always at least as selective as requiring two cameras above the threshold, and uses geometry to choose among such candidates. The score is independent of camera ordering and, with demand held fixed, invariant to a shared rigid transformation or uniform scaling of scene and cameras. When $V_i = 2$, $q_i^{\mathrm{rob}} = \min(d_{i1}, d_{i2}) \ge q_i^{\mathrm{geo}}$, so a proposal rejected by the core cannot be recovered; the recovery route requires at least three observing cameras. Directional complementarity serves as a selection heuristic; it does not guarantee that the geometry is uniquely identifiable.

3.6 Computation and integration

GI-ACE stores three scalar arrays of size $N \times C$ (gradient sums, observation counts, and weighted time sums) and the C camera centers. The arrays serve training only and are discarded with the topology window. The robust score costs $O(NC)$ per topology event, and pair scoring costs $O(MC^2)$ for the M large residual proposals reaching the geometric route; with four cameras there are six pairs. Evidence collection reuses gradients already produced by training, and the output at inference is an ordinary 4D Gaussian scene. Algorithm 1 in Appendix B summarizes the procedure.

The computational benefit accumulates over training: GI-ACE adds a small cost at each topology event and reduces the population that every intervening iteration renders, backpropagates, and updates. Because render cost also depends on projected footprint and visibility, we report measured training time alongside population size.

4 Experiments

4.1 Protocol

Datasets. We evaluate on six Neural3DV scenes (Li et al., 2022) and six Panoptic Sports sequences (Luiten et al., 2024; Joo et al., 2015). Neural3DV uses training cameras 01, 10, 13, and 20 at half resolution with 300 frames; every remaining camera and frame is held out (4,200–5,100 images per scene). Panoptic Sports provides 31 calibrated cameras at 640×360 over 150 frames; we train on cameras 04, 05, 14, and 29, selected by farthest-point sampling, and test on the remaining 4,050 images without using the released segmentations.

Baselines and configuration. Prior Neural3DV results are transcribed from Zhou et al. (2026). The published protocol specifies the camera split but neither populations nor runtimes, so efficiency on Neural3DV is compared against our reproductions of the same backbone under native densification with the GI-ACE configuration on the same GPU (Table 2); quality comparisons use the published values, which are the benchmark's public record. Where the literature reports no four-camera numbers (Panoptic Sports and the staggered-capture protocol), *4C4D* is trained with the released code and configurations. GI-ACE is implemented on the same backbone. All paired runs share initialization, training cameras, resolution, 30k iterations, and the full held-out evaluator.

Settings and metrics. GI-ACE uses batch size four, $\tau = 2 \times 10^{-4}$, topology events every 100 iterations from iteration 600 until the end of training, and no population budget. We report PSNR, LPIPS, DSSIM1/DSSIM2 ($(1-\mathrm{SSIM})/2$ at data ranges one and two, following Zhou et al. (2026)), final Gaussian count, and training time on a single GPU with 32 GB of memory and ≈52 TFLOPS peak FP32 throughput (excluding test rendering). Means are unweighted over scenes. Main results are single runs; on Cook Spinach, three seeds of a recovery control vary by 0.13 dB (s.d.), well below the dataset-level margins we report. Appendix D gives further evaluation details.

Table 1: Four-camera Neural3DV benchmark (six-scene means). Prior rows are transcribed from Table 1 of Zhou et al. (2026). GI-ACE uses the same public camera split.

Method	PSNR ↑	DSSIM1 ↓	DSSIM2 ↓	LPIPS ↓
ST-GS (Li et al., 2024b)	17.70	0.158	0.107	0.325
4DGaussians (Wu et al., 2024)	20.82	0.117	0.077	0.190
4DGS (Yang et al., 2024)	20.60	0.143	0.094	0.244
Ex4DGS (Lee et al., 2024)	19.33	0.149	0.099	0.239
4C4D (Zhou et al., 2026)	22.29	0.098	0.062	0.146
GI-ACE	**23.04**	**0.089**	**0.056**	**0.140**

Table 2: Neural3DV per-scene comparison. 4C4D quality metrics transcribe the per-scene results of Zhou et al. (2026); its Gaussian counts and training times are our reproductions of the same backbone under native densification with the GI-ACE configuration on the same GPU, as the published work reports neither. GI-ACE trains and evaluates under the same public four-camera split.

Scene	Method	PSNR ↑	LPIPS ↓	DSSIM1 ↓	DSSIM2 ↓	G (M) ↓	Time ↓
Cut Beef	4C4D	23.22	0.124	0.083	0.051	2.478	3:49:07
	GI-ACE	**24.101**	**0.1226**	**0.0766**	**0.0472**	**0.762**	**2:10:24**
Cook Spinach	4C4D	24.44	**0.111**	0.075	0.046	4.282	4:44:15
	GI-ACE	**24.646**	0.1172	**0.0739**	**0.0453**	**1.161**	**2:22:12**
Sear Steak	4C4D	23.66	0.118	0.078	0.048	1.735	3:13:50
	GI-ACE	**24.520**	**0.1166**	**0.0729**	**0.0451**	**0.568**	**1:58:27**
Coffee Martini	4C4D	19.57	0.203	0.140	0.093	2.273	3:48:10
	GI-ACE	**19.835**	**0.1931**	**0.1275**	**0.0844**	**0.739**	**2:01:08**
Flame Steak	4C4D	23.72	0.114	0.075	0.046	2.373	3:36:24
	GI-ACE	**24.685**	**0.1126**	**0.0682**	**0.0413**	**0.804**	**2:08:09**
Flame Salmon	4C4D	19.15	0.205	0.136	0.091	3.227	4:36:13
	GI-ACE	**20.434**	**0.1790**	**0.1163**	**0.0753**	**1.031**	**2:15:20**
Mean	4C4D	22.29	0.146	0.098	0.062	2.728	∼3:58:00
	GI-ACE	**23.037**	**0.1402**	**0.0892**	**0.0564**	**0.844**	∼2:09:00

4.2 Quality and efficiency

Neural3DV. GI-ACE exceeds every baseline in the published four-camera comparison (Table 1), including 4C4D (23.04 vs. 22.29 dB PSNR, 0.140 vs. 0.146 LPIPS, 0.089 vs. 0.098 DSSIM1). Per scene (Table 2), GI-ACE is higher than the published 4C4D result on all six scenes in PSNR, on five of six in LPIPS, and on all six in both DSSIM measures. Against our reproductions of the same backbone under native densification (same configuration), it uses 2.9–3.7× fewer final Gaussians and trains 1.6–2.0× faster, at quality above the published result in every scene.

Transfer. Table 3 evaluates the same rule, without retuning, on Panoptic Sports. GI-ACE improves mean PSNR by 0.45 dB, scores higher on four of six sequences, and wins every DSSIM cell; the baseline keeps the LPIPS mean and two sequences while using 11× the final Gaussians and 5.4× the training time. The efficiency gap is the largest in our study. With four wide-baseline views, native growth reaches 1.7–4.2M Gaussians without a structural-quality benefit, whereas GI-ACE finishes with 0.18–0.40M and trains in about 33 minutes instead of three hours. The scene-level gains are not carried by a few frames: GI-ACE renders the higher-PSNR image on 84% (Football) and 73% (Juggle) of held-out views (Wilcoxon signed-rank $p < 10^{-170}$).

Qualitative results. In Figure 2, the official model ghosts moving heads and arms and leaves floaters in static regions, consistent with growth driven by residuals that few views corroborate. The caption describes the crop-selection procedure; the tables report aggregate performance.

4.3 Does the choice of proposals matter?

If GI-ACE only grew less, any admission rule ending at the same population should reach the same quality. GI-ACE's own operating points already span a wide range (Appendix Table 6): halving the shared threshold τ grows the model to 3.13M Gaussians and reaches 24.93 dB on Cook Spinach,

Table 3: Panoptic Sports benchmark (six-sequence means; per-scene results in Appendix Table 9). 4,050 held-out images per sequence, no point budget. The 4C4D row uses the released code and configurations. G: final Gaussians (M).

Method	PSNR ↑	LPIPS ↓	DSSIM1 ↓	DSSIM2 ↓	G (M) ↓	Time (h) ↓
4C4D	17.64	**0.338**	0.163	0.102	3.14	2.91
GI-ACE	**18.09**	0.367	**0.143**	**0.089**	**0.28**	**0.54**



Figure 2: Held-out comparisons with the official 4C4D baseline (Cook Spinach under staggered and synchronized capture, Section 4.5; Flame Salmon). Crops come from a sliding-window search over all held-out cameras and every tenth frame, restricted to windows where GI-ACE is accurate (crop PSNR $\geq$ 26 dB, $\geq$ 28 dB for Flame Salmon, at most 3% of pixels with error > 0.1) and ranked by the baseline's excess error. Corner numbers are crop PSNR.

above the published 4C4D result on the scene (24.44 dB), while doubling it still yields 23.93 dB at 0.39M.

Matched-count selection. The sharper test fixes the population. Random recovery (three seeds) and demand-ranked recovery select, at every event, the same number of candidates that the geometric rule would admit from the eligible pool. Their final populations are 1.13–1.15M, compared with GI-ACE's 1.16M. GI-ACE's PSNR (24.646) exceeds all three random runs (24.361, 24.468, 24.626; s.d. 0.13 dB) and demand ranking (24.517), and its LPIPS matches the best control to within 0.001 (Table 4). At comparable capacity, the geometric score selects better candidates than either alternative on every run.

4.4 Operating limits

Sampling imbalance. With per-camera means (Equation 3), oversampling camera 01 by 4× and 8× changes the final population only from 1.16M to 1.13M and 1.09M, while PSNR drops by 0.20 and 0.33 dB. Replacing the mean with an unnormalized sum under 4× oversampling grows the model to 1.80M (+59%) and adds 48 minutes for only 0.09 dB (Appendix Table 7). Normalization thus decouples sampling frequency from population growth; it does not make quality invariant to a biased training stream.

Camera layout. With clustered cameras 01–04, GI-ACE falls to 15.57 dB and recovers only 3.7 splits per event, about $1/57$ of the wide-layout activity (Appendix Table 8); a medium-spread set of the same four cameras reaches 25.01 dB, above the wide default, so the deficit tracks the arrangement of views rather than their number. The geometric route becomes inactive under nearly collinear support, as the angular factor and the three-observer boundary of Section 3.5 predict, while held-out quality additionally depends on how well the training views cover the test views.

4.5 Asynchronous evidence under staggered observations

Because demands are aggregated over a window and rays are cast at each camera's own observation times, the pair test never requires two cameras to see the same instant. We stress this on Cook Spinach by giving each training camera only the frames with a distinct phase modulo four (75

Table 4: Cook Spinach matched-count recovery controls. Each count is the population the run converged to; no budget was set.

Recovery policy	PSNR ↑	LPIPS ↓	Final Gaussians	Train time
GI-ACE	**24.646**	0.1172	1,160,856	2:22:12
Random, seed 1	24.361	0.1182	1,133,486	**2:14:40**
Random, seed 2	24.468	0.1169	1,151,861	2:17:07
Random, seed 3	24.626	0.1177	1,138,208	2:17:35
Demand-ranked	24.517	**0.1163**	**1,131,947**	2:17:45

Table 5: Staggered-observation study on Cook Spinach with an identical 300-image training budget. Async assigns each training camera a distinct frame phase modulo four (each timestamp observed once); Sync assigns all four cameras the same phase (75 timestamps $\times$ 4 views). Evaluation uses the unchanged 17-camera test set. 4C4D rows use the released code with a 4M point cap.

Protocol	Method	PSNR ↑	LPIPS ↓	DSSIM1 ↓	DSSIM2 ↓	G (M)	Time
Async	4C4D	23.818	0.1395	0.0852	0.0533	4.014	5:36:25
Async	GI-ACE	24.614	**0.1203**	0.0760	0.0466	1.010	**2:25:07**
Async	GI-ACE, canonical rays	**24.737**	0.1207	**0.0755**	**0.0463**	**1.000**	3:18:37
Sync	4C4D	24.260	**0.1166**	0.0788	0.0487	3.197	4:47:12
Sync	GI-ACE	**24.442**	0.1220	**0.0778**	**0.0479**	**1.172**	**2:37:49**

frames each), so every timestamp is seen by exactly one camera. A synchronized control gives all cameras the same phase and the same number (300) of training images; test views are unchanged (Table 5).

Staggering improves GI-ACE from 24.44 to 24.61 dB while reducing the final model from 1.17M to 1.01M Gaussians. The gain comes from temporal coverage: the synchronized model loses 0.09 dB on the 225 unobserved timestamps, where the temporally uniform staggered model's advantage widens to 0.21 dB (Appendix Figure 3). The asynchronous run admits a smaller fraction of its evaluated pair candidates (9.4% vs. 24.6%) yet still recovers more splits per event (105 vs. 55), so staggering both widens the usable evidence pool and keeps admission strict. With a quarter of the training images, the staggered model comes within 0.03 dB of the standard protocol. The official backbone instead degrades from 24.26 to 23.82 dB under staggering, leaving GI-ACE 0.80 dB ahead with a quarter of the Gaussians and better on 97.6% of the 5,100 held-out images.

A canonical-center variant of the same rule scores even higher (24.74 dB): when each timestamp is seen by a single camera, the per-camera anchors $x_i(\bar{t}_{ic})$ are less constrained, so in this regime the measured benefit comes from demand aggregation and the parallax test.

5 Conclusion

Under sparse views, a large gradient says that a primitive wants to change but not whether the scene needs a new one. GI-ACE answers the second question with the camera structure of the evidence: a growth request must survive losing its strongest camera, or be backed by two cameras with real parallax at the times they observed the moving primitive. This simple, parameter-free rule changes the quality–population frontier of 4D Gaussian splatting. It exceeds, to our knowledge, all published four-camera baselines on Neural3DV while using 2.9–3.7$\times$ fewer Gaussians and training 1.6–2.0$\times$ faster than our backbone reproductions, and transfers to Panoptic Sports with an order-of-magnitude smaller model and 5.4$\times$ shorter training at higher mean PSNR. More broadly, the results suggest that in sparse multiview learning, how evidence is distributed across views decides where capacity goes.

Limitations and scope. Results are single runs on two benchmarks; scene-level differences near the 0.13 dB seed-level spread should be read conservatively, although our dataset-level margins are several times larger. The geometric route by construction requires at least three observing cameras and non-collinear support. The natural next step is to place the compact, explicit scene inside downstream systems—controllable 4D generation, streaming, and learned world models—where population and training cost are part of the asset budget.

AI USE STATEMENT

Generative AI tools assisted literature and dataset discovery, manuscript editing, LaTeX preparation, figure scripting, and the execution and organization of experiments specified by the authors. Reported measurements come from the recorded runs or the cited published tables. The authors remain responsible for verifying the final claims and artifacts.

REPRODUCIBILITY STATEMENT

Section 4 specifies datasets, camera splits, baselines, and configurations, and separates published results, released-code runs, and our reproductions. Appendices A–D provide derivations, implementation details, additional controls, and evaluation details. Every training run records its configuration, command-line arguments, camera set, initialization hash, population statistics, runtime, and peak memory; every evaluation records image coverage. An anonymized reference implementation containing the method, all configurations, the held-out evaluator, and per-experiment reproduction commands is included in the supplementary material; a public release will follow.

REFERENCES

Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B. Lindell. 4D-fy: Text-to-4d generation using hybrid score distillation sampling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.17984.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. OpenAI technical report, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.15391.

Xiaobin Deng, Changyu Diao, Min Li, Ruohan Yu, and Duanqing Xu. Improving densification in 3d gaussian splatting for high-fidelity rendering. *arXiv preprint arXiv:2508.12313*, 2025. URL https://arxiv.org/abs/2508.12313.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. LightGaussian: Unbounded 3D Gaussian compression with 15x reduction and 200+ FPS. In *Advances in Neural Information Processing Systems*, 2024.

Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Transactions on Graphics*, 2019.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. URL https://arxiv.org/abs/2308.04079.

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3D gaussian splatting as Markov chain Monte Carlo. In *Advances in Neural Information Processing Systems*, 2024.

Junoh Lee, ChangYeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. In *Advances in Neural Information Processing Systems*, 2024.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. DNGaussian: Optimizing sparse-view 3D Gaussian radiance fields with global-local depth normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Tianye Li, Mira Slavcheva, Michael Zollhöfer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5521–5531, 2022.

Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8508–8520, 2024b.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision (3DV)*, 2024. URL https://arxiv.org/abs/2308.09713.

Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Francisco Vicente Carrasco, Markus Steinberger, and Fernando De La Torre. Taming 3DGS: High-quality radiance fields with limited resources. In *SIGGRAPH Asia Conference Papers*, 2024.

Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. Revising densification in Gaussian splatting. In *European Conference on Computer Vision*, 2024.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20310–20320, 2024.

Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.10642.

Zongxin Ye, Wenyu Li, Sidun Liu, Peng Qiao, and Yong Dou. AbsGS: Recovering fine details for 3d gaussian splatting. *arXiv preprint arXiv:2404.10484*, 2024. URL https://arxiv.org/abs/2404.10484.

Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-GS: Density control with pixel-aware gradient for 3d gaussian splatting. *arXiv preprint arXiv:2403.15530*, 2024. URL https://arxiv.org/abs/2403.15530.

Junsheng Zhou, Zhifan Yang, Liang Han, Wenyuan Zhang, Kanle Shi, Shenkun Xu, and Yu-Shen Liu. 4C4D: 4 camera 4d gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2604.04063.

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: Real-time few-shot view synthesis using Gaussian splatting. In *European Conference on Computer Vision*, 2024.

A Derivations

A.1 Strongest-camera removal

For an observed-camera set S_i of size $V_i \geq 2$, removing camera a gives the mean $(\sum_{c\in S_i} d_{ic} - d_{ia})/(V_i - 1)$. The denominator is fixed, so the minimum over a is obtained by removing the largest demand, and Equation 4 is the worst single-camera-removal mean.

A.2 Time-conditioned representative position

With current primitive parameters fixed, the conditional trajectory is affine. Therefore,

$$\frac{\sum_{q\in W_{ic}} a_{iq} x_i(t_q)}{\sum_{q\in W_{ic}} a_{iq}} = x_i + v_i \left(\frac{\sum_{q\in W_{ic}} a_{iq} t_q}{\sum_{q\in W_{ic}} a_{iq}} - \mu_i \right) = x_i(\bar{t}_{ic}).$$

The time moment thus suffices to compute a representative current position for each camera's accumulated demand. Different cameras may have different representative times; the score measures directional support for the current dynamic primitive using those observations.

A.3 Geometry and the pair bound

For two unit rays, let $R = [r_{ia}, r_{ib}]$. Their Gram matrix satisfies

$$\det(R^\top R) = 1 - (r_{ia}^\top r_{ib})^2 = \|r_{ia} \times r_{ib}\|_2^2.$$

The geometric factor in Equation 6 is the area spanned by the unit directions and is zero when their span collapses. Since this area lies in $[0, 1]$, every pair score is at most its weaker demand, which is at most the second-largest demand. Taking the maximum proves Equation 8.

For a single nonzero camera demand, every pair minimum is zero and the robust score is zero, so both routes reject. For collinear rays, the geometric route rejects, and the robust route remains governed by its own criterion. Camera permutations reorder the candidate pairs without changing the maximum. Common rotations, translations, and uniform positive scaling preserve ray angles, which establishes the invariances with fixed demands.

A.4 Two-observed-camera boundary

If $V_i = 2$, then $q_i^{\mathrm{rob}} = \min(d_{i1}, d_{i2})$. Since the angular factor is at most one,

$$q_i^{\mathrm{geo}} = \min(d_{i1}, d_{i2})\|r_{i1} \times r_{i2}\|_2 \le q_i^{\mathrm{rob}}.$$

For a native proposal rejected by the core, $q_i^{\mathrm{rob}} < \tau$, hence $q_i^{\mathrm{geo}} < \tau$. The boundary concerns the number of cameras observing a primitive during its evidence window, not only the total number of training cameras.

A.5 Preservation of the robust core

Equation 7 is a union of the robust decision and a disjoint set of rejected-core refinements. Hence every core action remains accepted on the same state. The native scale predicate is unchanged, so accepted core clones remain clones and accepted core splits remain splits. This property concerns the local admission decision; later optimization and pruning determine the final population.

B Implementation details

The native proposal statistic is maintained separately from the per-camera means. For a minibatch where a primitive is visible in n images, the backbone sums per-image projected-gradient norms, multiplies by B/n, and averages these visible-batch scores across the topology window. GI-ACE keeps that statistic intact and uses Equation 3 for camera evidence.

The large-primitive predicate is $\max(s_i) > \rho E$, where s_i is the native spatial scale vector, E is the scene extent, and $\rho = 0.01$ is the backbone's density parameter. The Gaussian representation uses a full spacetime rotation, and velocity is extracted from the actual covariance with a floor of 10^{-8} on temporal variance. Ray normalization and time-weight division use numerical guards; these do not introduce fitted admission controls.

The float32 evidence arrays use $12NC$ bytes, about 48 MB for one million primitives and four cameras. Camera statistics follow the native topology reset and pruning remapping and are not part of the final scene. Optimizer handling and split sampling are inherited from the backbone.

GI-ACE uses the backbone's opacity-decay schedule, under which densification continues until iteration 30,000: topology events occur every 100 iterations from 600 through 29,900 (294 events). The opacity-reset interval is 10k iterations; all other hyperparameters follow the released 4C4D configuration.

Algorithm 1 GI-ACE training

```
Initialize the native 4D Gaussian scene and camera evidence arrays
for each ordinary shuffled camera–time minibatch do
    Render the images and run the native photometric backward passes
    Update the native densification statistics
    for all visible primitives do
        Accumulate per-camera gradient, count, and time
    end for
    if a native topology event occurs then
        Form proposals using the existing gradient threshold
        Accept the robust core using strongest-camera removal
        for each rejected large proposal with multiple observed cameras do
            Evaluate conditional centers at weighted observation times
            Score camera pairs using weaker demand times ray-angle sine
            if the best score reaches the same threshold then
                Recover a split
            end if
        end for
        Execute accepted native clones and splits; apply native pruning
        Reset the window statistics using the native topology bookkeeping
    end if
    Continue the native optimizer update and training schedule
end for
return the trained 4D Gaussian scene
```

C Additional experiments

C.1 Threshold sensitivity

Table 6: GI-ACE threshold sensitivity on Cook Spinach. The same threshold is used by the native proposal, robust, and geometric tests.

Threshold multiplier	PSNR ↑	LPIPS ↓	Final Gaussians	Train time
2	23.932	0.1567	**387,735**	**1:34:21**
1	24.646	0.1172	1,160,856	2:22:12
0.5	**24.928**	**0.1022**	3,133,850	3:33:35

Halving τ adds 0.28 dB at 2.7× the default population, and doubling it loses 0.71 dB at one third of the population; halving τ also exceeds the published 4C4D result on the scene (24.44 dB).

C.2 Topology statistics

The three random PSNR values in Table 4 have mean 24.485 and sample standard deviation 0.133 dB. LPIPS differs from GI-ACE by at most 0.001.

Across 294 topology events, the geometric route admits 69,026 additional split parents on Cut, 63,968 on Cook, 53,951 on Sear, 53,981 on Coffee, and 65,503 on Flame Steak. These counts are event-level parent admissions; some are removed by later pruning. On Cut and Cook they represent about 3.3% and 2.5% of admitted parent actions. At Cut's first event, seven of 2,090 eligible non-core large proposals pass the geometric threshold. On Cut, the robust core alone reaches 24.002 dB / 0.1246 LPIPS with 751,542 Gaussians, compared with 24.101 / 0.1226 / 761,790 for GI-ACE.

C.3 Sampling imbalance and camera layout

The scene-center pair-angle medians are approximately 17°, 110°, and 140° for the clustered, medium, and wide layouts. In the clustered layout, geometric recovery drops by roughly $57\times$ relative to the wide set, and held-out views outside the narrow cone fail. Medium outperforms wide in PSNR because its cameras cover the test views differently, despite lower recovery activity. Cam-

Table 7: Cook Spinach sampling-imbalance controls. Ratios are sampling weights for cameras 01/10/13/20. Mean uses Equation 3; raw sum removes observation-count normalization.

Method	Sampling / demand	PSNR ↑	LPIPS ↓	G (M)	Time
GI-ACE	1:1:1:1, mean	**24.646**	0.1172	1.161	2:22:12
GI-ACE	4:1:1:1, mean	24.446	0.1247	1.130	2:12:16
GI-ACE	8:1:1:1, mean	24.315	0.1332	**1.085**	**2:08:50**
GI-ACE	4:1:1:1, raw sum	24.531	**0.1171**	1.799	3:00:00

Table 8: Cook Spinach camera-layout controls: clustered 01–04, medium 05/08/12/15, and wide 01/10/13/20. Rec./event counts geometric recoveries per topology event; held-out sets differ across layouts.

Layout	Method	PSNR ↑	LPIPS ↓	G (M)	Time	Rec./event
Clustered	GI-ACE	15.569	0.2999	2.240	3:03:47	3.7
Medium	GI-ACE	25.012	0.1051	1.447	2:36:47	15.9
Wide	GI-ACE	24.646	0.1172	1.161	2:22:12	∼210

era geometry controls admission bandwidth, whereas novel-view quality depends jointly on that bandwidth and train–test coverage.

C.4 Per-frame quality under staggering

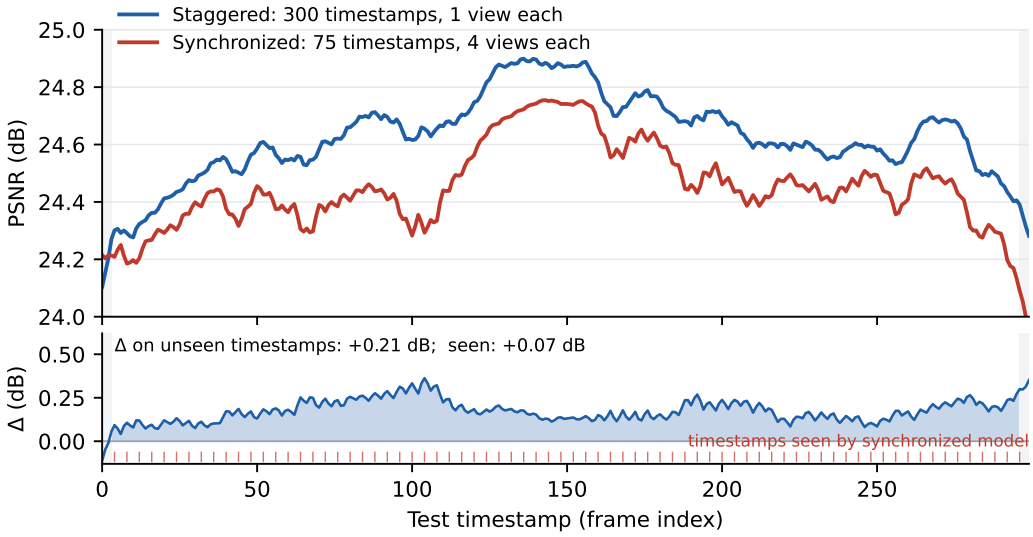

Figure 3: Held-out PSNR per test timestamp on Cook Spinach (mean over the 17 held-out cameras). The staggered model observes every timestamp once and is temporally uniform; the synchronized model concentrates quality near the 75 timestamps it observed (tick marks) and loses 0.09 dB on the 225 unobserved ones, where the staggered advantage widens to 0.21 dB. Shaded margins mark boundary-extrapolation frames.

C.5 Per-scene transfer results

D Evaluation details

Coverage. All reported evaluations render the complete held-out set with no skipped images.

Initialization and calibration. Each paired Neural3DV comparison uses the same released initialization (529k–574k points). Panoptic Sports uses the released calibration. No meshes, masks, or depths are used as supervision.

Table 9: Paired Panoptic Sports comparison at native 640×360 resolution (4,050 held-out images per scene, no point budget). 4C4D rows use the released code and configurations. G: final Gaussians (M).

Scene	Method	PSNR ↑	LPIPS ↓	DSSIM1 ↓	DSSIM2 ↓	G (M) ↓	Time ↓
Tennis	4C4D	**18.437**	**0.3319**	0.1589	0.0980	4.200	5:09:47
	GI-ACE	17.132	0.3707	**0.1448**	**0.0925**	**0.308**	**0:32:41**
Football	4C4D	16.469	**0.3460**	0.1768	0.1147	3.174	2:36:15
	GI-ACE	**18.491**	0.3721	**0.1426**	**0.0880**	**0.306**	**0:33:51**
Boxes	4C4D	17.888	**0.3283**	0.1583	0.0992	1.715	1:45:00
	GI-ACE	**18.524**	0.3524	**0.1383**	**0.0873**	**0.177**	**0:30:21**
Basketball	4C4D	17.461	**0.3468**	0.1652	0.1035	3.472	2:47:11
	GI-ACE	**18.366**	0.3738	**0.1504**	**0.0926**	**0.401**	**0:35:34**
Juggle	4C4D	17.312	**0.3461**	0.1673	0.1045	2.575	2:17:20
	GI-ACE	**18.269**	0.3732	**0.1353**	**0.0838**	**0.218**	**0:31:10**
Softball	4C4D	**18.279**	**0.3277**	0.1526	0.0939	3.671	2:51:36
	GI-ACE	17.736	0.3590	**0.1441**	**0.0905**	**0.297**	**0:31:54**
Mean	4C4D	17.641	**0.3378**	0.1632	0.1023	3.135	2:54:32
	GI-ACE	**18.086**	0.3669	**0.1426**	**0.0891**	**0.284**	**0:32:35**

Runtime and memory. Training time covers optimization, periodic validation, and checkpoint writing, and excludes test rendering. All runs use the same GPU (32 GB memory, ≈52 TFLOPS peak FP32); official and GI-ACE runs use their respective released software environments, so time ratios measure practical end-to-end implementations rather than isolated admission overhead.

Table 10: Neural3DV structural metrics. 4C4D values are published in Tables 6–7 of Zhou et al. (2026); GI-ACE uses our evaluator with the same DSSIM definition. Our rendered official-code Flame Salmon model gives 0.1332 / 0.0879, close to the published values.

	DSSIM1 ↓		DSSIM2 ↓	
Scene	4C4D	GI-ACE	4C4D	GI-ACE
Cut Roasted Beef	0.083	**0.0766**	0.051	**0.0472**
Cook Spinach	0.075	**0.0739**	0.046	**0.0453**
Sear Steak	0.078	**0.0729**	0.048	**0.0451**
Coffee Martini	0.140	**0.1275**	0.093	**0.0844**
Flame Steak	0.075	**0.0682**	0.046	**0.0413**
Flame Salmon	0.136	**0.1163**	0.091	**0.0753**
Mean	0.098	**0.0892**	0.062	**0.0564**