Registering Gaussian Splats Without the Point-Cloud Detour: Accuracy, Representation Semantics, and a Negative Result on Hypothesis-Stage Transfer

Krishi Attri
Seoul National University
kattri@snu.ac.kr ORCID: 0009-0005-4695-6467

Abstract. Aligning two Gaussian splats of the same scene today requires either manual alignment in an editor or a detour through point clouds: export the Gaussian means, register those, and re-import a transform whose effect on anisotropy, opacity, and view-dependent color is left undefined. We built SPLATREG, an open pure-PyTorch library, to establish what registering splats natively actually requires, and this paper reports the result. First, accuracy costs nothing. A signed-distance field derived in closed form from the frozen target Gaussians, with a numerically audited Jacobian over SE(3) and Sim(3) and a Levenberg–Marquardt core under interchangeable seeds (FPFH, learned, zero-shot, global, maximal-clique), reaches 91.5% mean / 93.5% pooled registration recall on official 3DMatch and 72.5% / 74.4% on 3DLoMatch, matching the GeoTransformer local-to-global baseline it refines while adding Sim(3) scale and pose covariance; a drop-in zero-shot seed (BUFFER-X, ICCV 2025), which we ported to a modern CUDA stack, lifts a classical FPFH front-end from 0.630 to 0.962 registration recall on the official 3DMatch pairs and from 0.122 to 0.777 on 3DLoMatch through the identical refine; against the existing splat tools it is 2.9× more accurate in rotation (5.2° vs 15.3°) and improves merge Chamfer 5.1× over naive concatenation. Second, the genuinely new work is semantic, not numeric. A splat is not closed under rigid motion the way a point cloud is: view-dependent color must be Wigner-rotated (real-basis SH rotation, verified to 2.4e-15; none of the splat tools we surveyed rotate SH), scale conventions must be normalized before fusion (mixed log/linear scales corrupt merges silently), and a baked transform must update means, quaternions, scales, and SH together. Each requirement is pinned by a released test, and we propose the checklist as the correctness bar for future splat registrars. Third, a negative result we did not expect. MAC, the maximal-clique estimator that lifts GeoTransformer by 3.7 and 3.9 recall points in its own evaluation, engages on 100% of pairs inside our pipeline yet moves both official splits by at most 4 pairs while costing 50% more runtime: native-voxel learned correspondences are already 60–80% inliers, and a residual-gated refine absorbs whatever seed differences remain. Fourth, an honest study of the pose covariance the library exposes: raw LM/Laplace covariance is grossly over-confident on real data (nominal-90% coverage 0.005) but split-conformal recalibration restores it at strong n (0.900 / 0.905), and yet covariance-gated fusion is a second negative result — every uncertainty gate collapses to blanket abstention because the covariance does not rank registration failures (AUC 0.578, chance) while the LM residual does (AUC 0.773). We argue published hypothesis-stage gains should be presumed pipeline-conditional until re-measured, and we report the covariance as a calibratable diagnostic rather than a selective fusion signal. All numbers are reproducible from the released library and benchmarks.

1 Introduction

3D Gaussian Splatting (3DGS) [1] is now a default representation for captured 3D content, with a mature forward toolchain: `gsplat` [2] renders splats, editors such as SuperSplat manipulate them, and a hundred-strong family of GS-SLAM systems [3, 4] builds them online. The *inverse* operation, recovering the SE(3) or Sim(3) transform that aligns two splats of the same scene, has no comparable tool. Practitioners who wish to combine captures today have two options, both unsatisfactory. Existing editors provide only manual alignment: the most widely used editor ships no automatic registration, and its maintainers state that automatic combination is unsupported[1]. The alternative is to export the Gaussian means as a point cloud, run a classical pipeline, and re-import the recovered transform—a route that discards anisotropy, opacity, and view-dependent color and leaves the transform's effect on those discarded attributes undefined.

This paper asks what it takes to do the job natively, and reports what we learned building and validating SPLATREG, an open, pure-PyTorch library for splat-to-splat registration[2]. Four questions organize what follows.

RQ1 (accuracy). Can rigid/similarity registration operate directly on the 3D Gaussian representation, without a point-cloud detour, at accuracy competitive with point-based pipelines? Our method drives Levenberg–Marquardt (LM) over a signed-distance field derived in closed form from the frozen target Gaussians, with a numerically au-

[1] https://github.com/playcanvas/supersplat/issues/53

[2] https://github.com/Archerkattri/splatreg

dited analytic Jacobian over SE(3) and Sim(3), beneath interchangeable seed generators (FPFH+RANSAC, learned correspondences, a zero-shot registrar, a blind global SO(3) sweep, and maximal-clique consensus). On the official 3DMatch/3DLoMatch protocol it matches the published GeoTransformer [5] recall while adding Sim(3) scale estimation and splat-native outputs; on splat-tool benchmarks it wins outright (Sec. 4). We further add an optional zero-shot front-end — BUFFER-X [6] (ICCV 2025), which we ported to a modern CUDA stack — and show that, through the identical refine, it lifts seed-isolated recall over the classical FPFH fallback from 0.630 to 0.962 on official 3DMatch and from 0.122 to 0.777 on 3DLoMatch (Sec. 4.2).

RQ2 (correctness semantics). What does splat-native registration require for *correctness* that point-cloud registration never faces? A point cloud is closed under rigid motion: rotating the points suffices. A Gaussian splat is not. We identify three requirements, each of which we found violated or absent in surveyed tools, and each of which we pin with tests: (a) view-dependent color stored as real spherical-harmonic (SH) coefficients must be rotated by the real-basis Wigner-D matrices, or glossy highlights stay frozen in the old capture frame; (b) scale parameters stored under different conventions (log vs linear) must be normalized before fusion, or merges silently mis-exponentiate; (c) baking a transform must update means, rotations, scales, and SH jointly and consistently. We frame these as a checklist for any future splat registrar (Sec. 5).

RQ3 (negative result, with mechanism). Do hypothesis-generation improvements transfer into a refine-equipped pipeline? MAC [7], a CVPR 2023 best-paper candidate, replaces RANSAC hypothesis generation with maximal-clique consensus and reports lifting GeoTransformer itself by 3.7 and 3.9 recall points on 3DMatch and 3DLoMatch in its own evaluation. We reimplemented it faithfully, verified that it engages on every pair, and measured no net change on either official split, at +50% runtime, inside our pipeline. We give the mechanism: native-resolution learned correspondences are already consensus-dominated, and a residual-gated refinement absorbs residual seed differences. MAC retains a decisive advantage exactly where its theory predicts: contaminated, multi-consensus correspondence sets (Sec. 6).

RQ4 (pose covariance, calibrated and stress-tested). Every LM solve exposes a pose covariance from its information matrix; is it trustworthy, and does it support a downstream decision? We find the raw covariance grossly over-confident on real 3DMatch/3DLoMatch, but a split-conformal recalibration restores nominal coverage at strong n. When we then test whether that covariance can gate real low-overlap splat fusion, it cannot: every uncertainty gate collapses to blanket abstention, and the covariance does not discriminate registration failures (AUC 0.578) while the LM residual does (AUC 0.773). We report the covariance as a calibratable diagnostic and name the residual as the load-bearing gate (Sec. 7).

We additionally report, in the spirit of RQ2, a methods note on a bug our own benchmarks did not catch (a residual weight applied three times; Sec. 8), and we describe the one pending experiment, a head-to-head against GaussReg on its ScanNet-GSReg benchmark, without projecting numbers (Sec. 9).

Contributions.

1. **A splat-native registration method and library.** A frozen Gaussian-density anchor SDF with a closed-form, numerically audited Jacobian over SE(3)/Sim(3), an LM driver, and interchangeable seeds, reaching 91.5% mean / 93.5% pooled on official 3DMatch and 72.5% / 74.4% on 3DLoMatch (matching the GeoTransformer-LGR baseline it refines, while adding Sim(3) scale and splat-native outputs), and winning every splat-tool head-to-head we ran.
2. **A zero-shot registration front-end.** We integrate BUFFER-X [6] as an optional seed and reproduce it on a modern CUDA stack (sm_120 / CUDA 12.8 / torch 2.11 / numpy 2.x); through the identical refine it lifts seed-isolated registration recall over the classical FPFH seed from 0.630 to 0.962 on the official 3DMatch pairs and from 0.122 to 0.777 on 3DLoMatch, winning all eight scenes on both splits.
3. **A representation-semantics correctness checklist** for splat-native registration: SH Wigner rotation (verified to 2.4e-15 in float64; among surveyed splat tools, none rotate SH), scale-convention normalization, and consistent transform baking, each pinned by released tests.
4. **A negative result with mechanism:** MAC hypothesis generation, reimplemented faithfully and engaged on 100% of pairs, produces no net change on either official split inside a refine-equipped pipeline (every delta within 4 pairs, +50% runtime), winning only the contaminated-correspondence regime. We argue published hypothesis-stage gains are measured against weaker receiving pipelines and should be presumed pipeline-conditional.
5. **A calibration study and a second negative result:** the library's pose covariance is grossly over-confident on real data (coverage 0.005 at nominal 0.90) but split-conformally recalibratable to nominal (0.900 / 0.905); covariance-gated fusion, however, does not beat naive fusion selectively — every gate collapses to blanket abstention — because the covariance does not rank failures (AUC 0.578) while the LM residual does (AUC 0.773).
6. **A benchmark-hygiene case study:** a triple-applied residual weight (effective w^3) that survived an end-to-end suite because a clean-synthetic benchmark was dominated by

a different residual, reinforcing the checklist-and-unit-pin discipline of RQ2.

2 Related Work

Splat-to-splat registration. GaussReg [8] is the closest published method: it registers two 3DGS scenes coarse-to-fine and contributes the ScanNet-GSReg benchmark, but its pipeline operates on point clouds extracted from the Gaussians and does not address what a recovered transform must do to the splat's own attributes. PhotoReg [9] registers 3DGS models photometrically through rendered views, complementary to our geometric core (our optional photometric stage is PhotoReg-style; Sec. 3). Open-source tools fill the practical gap partially: `splatalign` [10] runs multi-scale point-to-point ICP from identity, and `GaussianSplattingRegistration` [11] wraps Open3D [12] FPFH+RANSAC+ICP behind a GUI; both are SE(3)-only, both operate on means as points, and neither rotates SH or models scale. Editors (SuperSplat, SplatTransform) offer manual transforms only. To our knowledge SPLATREG is the only registrar that treats the splat as the first-class representation, with Sim(3) scale, SH rotation, fusion semantics, and pose covariance as library guarantees.

Point-cloud and zero-shot registration. Our learned seed builds on GeoTransformer [5] and its local-to-global registration (LGR), evaluated under the canonical 3DMatch [13] / 3DLoMatch [14] protocol with the covariance-weighted error of Choi et al. [15]. Classical seeds use FPFH [16] with RANSAC, and our global fallback sweeps super-Fibonacci SO(3) samples [17]. Because a per-dataset-trained matcher is exactly the dependency a splat registrar should avoid across sensors and scales, we also integrate BUFFER-X [6], a zero-shot registrar (ICCV 2025) that transfers without per-dataset training, as an optional front-end (Sec. 4.2). MAC [7] represents the hypothesis-generation line we test in RQ3. ICP variants [18, 19] underlie the refinement stage, and Umeyama's closed form [20] anchors similarity fitting.

Registration needs inside 3DGS systems. GS-SLAM systems increasingly need splat-level registration as a subroutine: LoopSplat [21] closes loops by registering 3DGS submaps, and submap fusion in SplaTAM/MonoGS-style pipelines [3, 4] faces the same merge-semantics questions we formalize in RQ2. SPLATREG is designed as the reusable component these systems currently reimplement: every LM solve also exposes the pose information/covariance pair for downstream pose graphs.

Registration uncertainty and calibration. The information matrix $J^\top W J$ of a converged LM/Gauss–Newton solve is the standard pose-covariance estimate a pose graph consumes; RQ4 asks whether it is calibrated on real splat registration, and recalibrates it with split-conformal prediction [22], the distribution-free wrapper that turns a miscalibrated score into nominal coverage from a held-out calibration set.

SH rotation. Rotating real spherical harmonics by Wigner-D matrices is classical [23, 24]; the observation of RQ2 is not that the math is new but that the surveyed splat-registration tools simply do not apply it, so view-dependent appearance silently decouples from geometry under any baked rotation.

3 Method: Registering Against Gaussians

SPLATREG takes two splats, a target $\mathcal{A}$ and source $\mathcal{B}$, and returns the SE(3) or Sim(3) transform aligning $\mathcal{B}$ to $\mathcal{A}$, optionally followed by fusion with overlap deduplication. The pipeline is a global initializer followed by multi-residual LM refinement.

3.1 The Gaussian-SDF residual

The flagship residual derives a smooth signed-distance field directly from the frozen target Gaussians, with no meshing and no marching cubes. For a query point p and target anchors q_i with unit normals n_i:

$$
\begin{aligned}
w_i(p) &= \exp\!\left(-\|p - q_i\|^2 / 2\sigma^2\right), \\
\tilde{q}(p) &= \textstyle\sum_i w_i q_i \,/\, \sum_i w_i, \quad \tilde{n}(p) = \frac{\sum_i w_i n_i}{\|\sum_i w_i n_i\|}, \\
d(p) &= \big(p - \tilde{q}(p)\big) \cdot \tilde{n}(p),
\end{aligned} \tag{1}
$$

which vanishes exactly when source points land on the target surface. The residual has a *closed-form* spatial gradient: the true gradient of d includes a first-order $\partial\tilde{q}/\partial p$ term (the kernel-weighted centroid moves with p) that the naive surface-normal approximation drops. This is not pedantry: our numerical Jacobian audit (below) caught exactly this error in our own implementation, with a maximum analytic-vs-numerical discrepancy of 10.8 before the fix and ∼1e-8 after.

3.2 Solver, Jacobian audit, and seeds

A from-scratch LM core solves the full SE(3) (6-DoF) or Sim(3) (7-DoF, with scale) tangent over a residual stack (ICP point-to-point, point-to-plane, and the Gaussian-SDF term), and exposes the undamped $J^\top W J$ information matrix and its scaled inverse as pose covariance at the optimum (`None` when singular, never a faked inverse). Every

analytic Jacobian is audited against a tangent-space numerical Jacobian in float64, the discipline GTSAM enforces with `EXPECT_CORRECT_FACTOR_JACOBIANS`: ICP point-to-point agrees to ∼3e-9, point-to-plane to ∼4e-11, and the SDF gradient to ∼1e-8 after the closed-form fix. A second audit-adjacent fix made the near-π SO(3) logarithm robust (the antisymmetric part vanishes at $\theta = \pi$; the symmetric-part axis with `atan2` restores round-trips to ∼1e-13).

Seeds are interchangeable: `fast` (FPFH + GPU-batched RANSAC, ∼17 ms), `robust` (Open3D FPFH+RANSAC), `learned` (pretrained GeoTransformer with its LGR pose as the starting point), `bufferx` (the pretrained BUFFER-X zero-shot registrar; Sec. 4.2), `global` (a blind super-Fibonacci SO(3) sweep with batched trimmed ICP), and `mac` (maximal-clique consensus; Sec. 6). On top of a learned seed, the refine is *residual-gated*: the ICP/Sim(3) refinement is accepted only when it does not worsen the overlap residual, so it can tighten but never degrade the seed pose. A per-pair audit on one official scene found zero pairs where the refine demoted a seed success.

3.3 Photometric refinement with exposure compensation

Geometry cannot see every degree of freedom: a rotationally symmetric object carries its azimuth only in color. An opt-in PhotoReg-style stage renders both splats and refines the pose photometrically, with two additions we found load-bearing. First, *exposure compensation* (default on): a bounded per-channel gain/bias on the rendered source (gain $\in [0.5, 2.0]$, $|\text{bias}| \leq 0.2$), fit in closed form and alternated with the pose LM. Without it, a $\times 1.3 + 0.05$ source tint absorbs into the Sim(3) scale degree of freedom, corrupting recovered scale from 0.10% error (clean) to 3.99%; with it, the tinted pair recovers 0.47% and a clean pair is unaffected (0.01%). Second, a *coarse-to-fine render ladder*: from a 6° offset, a single 96-px rung stalls at 5.61° while a 32→64→96 ladder lands 2.55° at equal per-stage budget. On the real rasterizer the full stage takes a 5°/7 mm offset to 0.36°/0.5 mm in ∼1.1 s; on a symmetric splat the geometric stage alone *worsens* rotation 6.0°→11.2° while the photometric stage lands 2.2°. On dense-overlap real captures it is neutral, which is why it ships opt-in.

4 RQ1: Is Splat-Native Registration Competitive?

4.1 Official 3DMatch / 3DLoMatch

We evaluate on the canonical protocol [13–15]: the 1279 non-adjacent `gt.log` pairs of the eight 3DMatch test scenes and the 1726 pairs of 3DLoMatch, scored by the covariance-weighted transform error $e^\top C e \leq 0.2^2$ from `gt.info`. Table 1 and Fig. 1a summarize.

Method	3DMatch RR	3DLoMatch RR
SPLATREG `learned` (native 0.025 vox.)	**91.5 / 93.5**	**72.5 / 74.4**
SPLATREG `learned` (legacy 0.05 vox.)	86.3 / 89.1	55.3 / –
SPLATREG `robust` (classical seed)	∼67.1	∼15
GeoTransformer [5] (publ.)	∼92	∼74
Open3D FPFH+RANSAC	∼77	∼20

Table 1: Registration recall (%) on official 3DMatch / 3DLoMatch, reported as mean-of-scenes / pooled-over-pairs. The native-voxel SPLATREG `learned` path matches the published GeoTransformer recall while adding Sim(3) scale estimation, splat-native outputs, and pose covariance. RRE 1.81° and RTE 0.071 m on 3DMatch successes.

Two readings matter. First, the headline: the splat-native refine, layered over the GeoTransformer LGR pose, scores 91.5% mean / 93.5% pooled on 3DMatch and 72.5% / 74.4% on 3DLoMatch, matching the published GeoTransformer numbers (∼92 / ∼74). The refine preserves the learned seed's recall (zero demotions in the per-pair audit) and tightens RRE within already-successful pairs, while adding what the point pipeline does not have: Sim(3) scale, attribute-correct baked outputs, and covariance. Second, the diagnosis behind the earlier gap: our initial harness pre-voxelized both fragments to 0.05 m before GeoTransformer saw them (∼5k vs ∼19k points per fragment), discarding over 70% of the points the matcher was trained on (its native `init_voxel_size` is 0.025). Restoring native resolution lifted 3DMatch 86.3%→91.5% and 3DLoMatch 55.3%→72.5% mean / 74.4% pooled. The fix was entirely in the *benchmark harness*; the method was unchanged (the full point-count analysis is Appendix C). We flag this because the same artefact would silently penalize any refinement-style method evaluated behind an aggressive pre-voxelization.

4.2 A zero-shot front-end: BUFFER-X

The official numbers above refine a GeoTransformer seed trained on 3DMatch. For captures from a different sensor or a different scale, a per-dataset-trained matcher is exactly the dependency a splat registrar should avoid. We therefore add an optional zero-shot seed: BUFFER-X [6] (ICCV 2025), a single generalist model that registers across sensors and scales with no per-dataset training, feeding SPLATREG's overlap-aware refine and Sim(3) scale exactly as the other seeds do.

To isolate the seed from the pipeline, we run both the BUFFER-X seed and the classical FPFH+RANSAC seed through the identical (lighter) feature-alignment refine and score by the standard registration-recall criterion (RRE < 15° and RTE < 30 cm) on the complete official `gt.log` pair

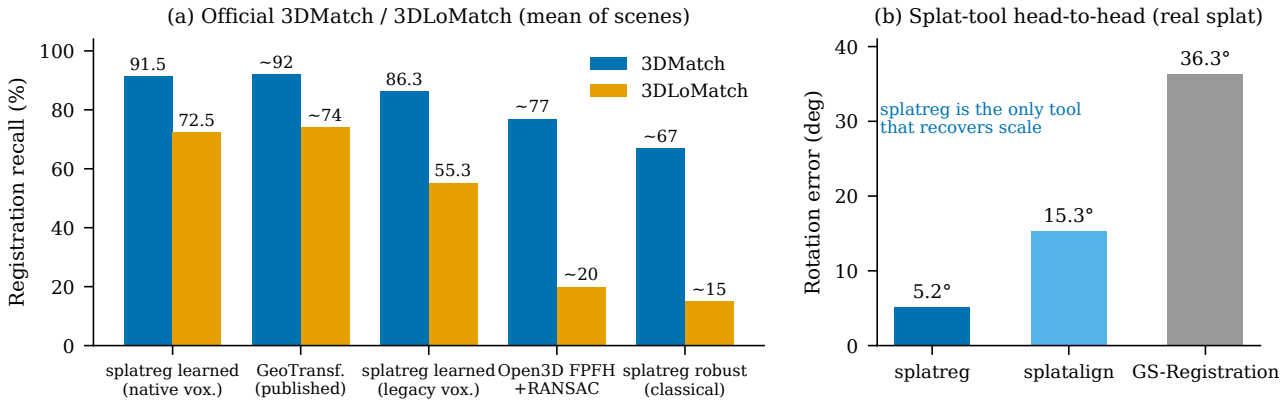

Figure 1: The RQ1 accuracy landscape. (a) Registration recall (mean of scenes) on the official 3DMatch and 3DLoMatch protocols: the splat-native SPLATREG `learned` path (91.5 / 72.5) matches the published GeoTransformer baseline it refines (~92 / ~74) while adding Sim(3) scale, splat-native attribute baking, and pose covariance. The earlier gap was a harness artefact, not the method: pre-voxelizing fragments to 0.05 m (legacy) costs 5.2 points on 3DMatch and 17.2 on 3DLoMatch, and classical seeds collapse on low overlap. (b) Rotation error against the existing splat tools on a real capture with known ground-truth Sim(3): SPLATREG (5.2°) is 2.9× more accurate than splatalign (15.3°) and 7× more accurate than GaussianSplattingRegistration (36.3°), and is the only tool that recovers scale at all.

Seed (identical refine)	3DMatch RR	3DLoMatch RR
BUFFER-X (zero-shot)	**0.962** (1.46°)	**0.777** (2.77°)
classical FPFH seed	0.630 (2.12°)	0.122 (103.4°)
official pairs n	1619	1781

Table 2: Zero-shot vs classical seed on the complete official `gt.log` pair sets, both through the identical lighter feature-align refine (recall at RRE < 15° and RTE < 30 cm; median RRE in parentheses). This isolates the seed; it is not comparable to the covariance-weighted numbers of Table 1 nor a leaderboard submission.

sets.[3] Table 2 and Fig. 2 report the result: on 3DMatch (all eight scenes, n=1619) the zero-shot seed reaches 0.962 registration recall (median RRE 1.46°) against 0.630 (2.12°) for the classical seed, and on 3DLoMatch (n=1781) it holds 0.777 (2.77°) against 0.122 (103.4°) — 6.4× the recall where the classical FPFH seed's median error (103°) is effectively random. BUFFER-X wins every one of the eight scenes on both splits (Fig. 2b,c), by recall margins from +0.24 (home_at, 3DMatch) to +0.79 (redkitchen, 3DLoMatch); Table 3 gives the complete per-scene recall and median-RRE breakdown.

Two caveats bound the claim. First, this is a seed-isolating internal comparison, not a leaderboard result: both arms share the lighter feature-align refine (not the full gated pipeline of Table 1) and the RRE/RTE metric differs from the covariance-weighted protocol above, so the 0.962 here is *not* comparable to the 91.5 there. Second, the value is the low-overlap regime and the no-training property, not the last recall point: per-dataset-trained backbones now top the 3DMatch leaderboard above the GeoTransformer seed SPLATREG wraps, and any such checkpoint can be integrated as a seed as soon as it is released as a permissive zero-shot model.

A modern-CUDA reproducibility contribution. BUFFER-X's public release targets an older stack; making it build and run on current hardware (an RTX 5090, sm_120, CUDA 12.8, torch 2.11, numpy 2.x) was non-trivial, and we release the exact sudo-free recipe (the complete step-by-step recipe is Appendix A). The load-bearing fixes: patching pointnet2_ops' hard-coded compute-3.7/5.0 architecture list, which nvcc 12.8 rejects, to sm_120; porting the KPConv C++ neighbour wrappers from the numpy-1.x to the numpy-2.x C-API (`NPY_ARRAY_IN_ARRAY`, `PyArrayObject` casts); pure-torch shims for two CUDA-only dependencies (nearest-neighbour via `torch.cdist+topk`, batched SVD via `torch.linalg.svd`); and a weight-loading fix, since the pretrained checkpoints are full-model state dicts that, loaded into the `.Desc/.Pose` submodules under `strict=False`, silently matched nothing and ran on random weights. We flag reproducibility on modern accelerators as its own contribution: a zero-shot registrar is only useful if it runs on currently available hardware.

4.3 Against splat-registration tools

On a real captured splat under a known ground-truth Sim(3) (Table 4, Fig. 1b), SPLATREG achieves 5.2° rotation error (SE(3)) and 15.7 mm translation versus 15.3° for

[3]The BUFFER-X harness scores the full official `gt.log` set; on 3DLoMatch this is n=1781 including a handful of adjacent pairs (the non-adjacent-only subset, n=1726, gives 0.772 / 0.114, the same verdict).

	3DMatch (RR / med. RRE)			3DLoMatch (RR / med. RRE)		
Scene	n	BUFFER-X	classical	n	BUFFER-X	classical
7-scenes-redkitchen	506	**0.994** (1.4°)	0.670 (1.8°)	525	**0.895** (2.4°)	0.107 (105°)
sun3d-home_at	156	**0.987** (1.4°)	0.750 (1.8°)	289	**0.696** (2.8°)	0.104 (107°)
sun3d-home_md	208	**0.899** (1.9°)	0.587 (2.6°)	230	**0.752** (3.6°)	0.222 (100°)
sun3d-hotel_uc	226	**0.996** (1.5°)	0.730 (1.8°)	218	**0.922** (2.6°)	0.179 (100°)
sun3d-hotel_umd1	104	**0.952** (1.4°)	0.558 (2.8°)	158	**0.759** (2.8°)	0.120 (110°)
sun3d-hotel_umd3	50	**1.000** (1.4°)	0.740 (1.9°)	49	**0.755** (2.6°)	0.204 (89°)
sun3d-mit_studyroom	292	**0.914** (1.6°)	0.483 (7.7°)	240	**0.567** (4.1°)	0.021 (103°)
sun3d-mit_lab_hj	77	**0.935** (1.3°)	0.532 (2.2°)	72	**0.639** (3.8°)	0.097 (109°)
all scenes	1619	**0.962** (1.46°)	0.630 (2.12°)	1781	**0.777** (2.77°)	0.122 (103.4°)

Table 3: Complete per-scene official-pair results behind Table 2 and Fig. 2: registration recall (RRE < 15° and RTE < 30 cm) with median RRE in parentheses, for the BUFFER-X zero-shot seed and the classical FPFH seed through the *identical* lighter refine, on all eight test scenes of each split. BUFFER-X wins every scene on both splits. On 3DLoMatch the classical seed's median RRE is ∼100° in every scene (effectively random), so its non-zero recall comes from a small tail of easy pairs. The per-scene pair counts n differ between splits because each split ships its own `gt.log` list.

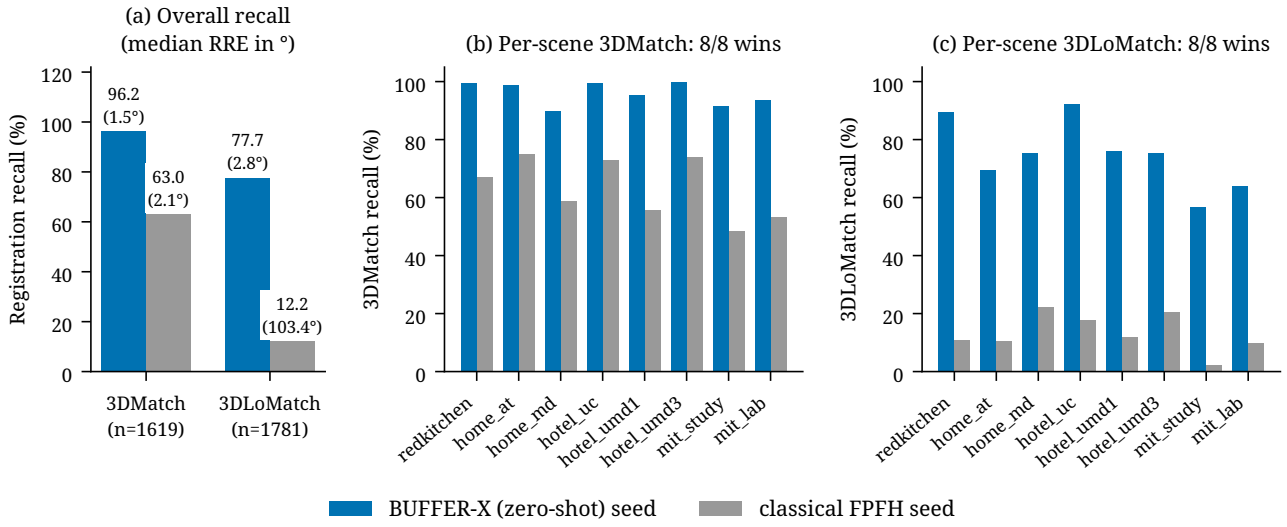

Figure 2: The zero-shot front-end. (a) Registration recall on the complete official `gt.log` pair sets (median RRE in parentheses), both seeds through the identical lighter refine: BUFFER-X reaches 0.962 on 3DMatch (n=1619) and 0.777 on 3DLoMatch (n=1781) against 0.630 and 0.122 for the classical FPFH seed, which on 3DLoMatch collapses to a 103.4° median (near-random). (b,c) Per-scene recall on each split: the zero-shot seed wins all eight test scenes on both 3DMatch (b) and 3DLoMatch (c), the hardest scene (mit_study, 3DLoMatch) still 0.567 versus 0.021 for the classical seed. Per-scene numbers in Table 3.

`splatalign` (ICP from identity) and 36.3° for `Gaussian SplattingRegistration` (Open3D RANSAC+ICP), and it is the only tool that recovers scale at all: both competitors are SE(3)-only and cannot model the ground-truth scale.

4.4 The splat-native deliverables

Merge quality. On a real 103k-Gaussian capture, two overlapping captures fused by SPLATREG (registered Sim(3) + voxel/KNN overlap dedupe, ∼9k duplicates removed) reach 2.0 mm Chamfer to ground truth versus 10.3 mm for a naive `torch.cat` (5.1× closer), with overlap IoU 0.67 versus 0.03

Tool	rot. err	trans. err	scale
SPLATREG (SE3)	**5.2°**	**15.7 mm**	–
SPLATREG (Sim3)	11°	–	**only tool**
splatalign	15.3°	–	SE(3)-only
GS-Registration	36.3°	–	SE(3)-only

Table 4: Head-to-head on a real splat with known ground-truth Sim(3). Each tool recovers the applied transform; SPLATREG is the only one estimating scale.

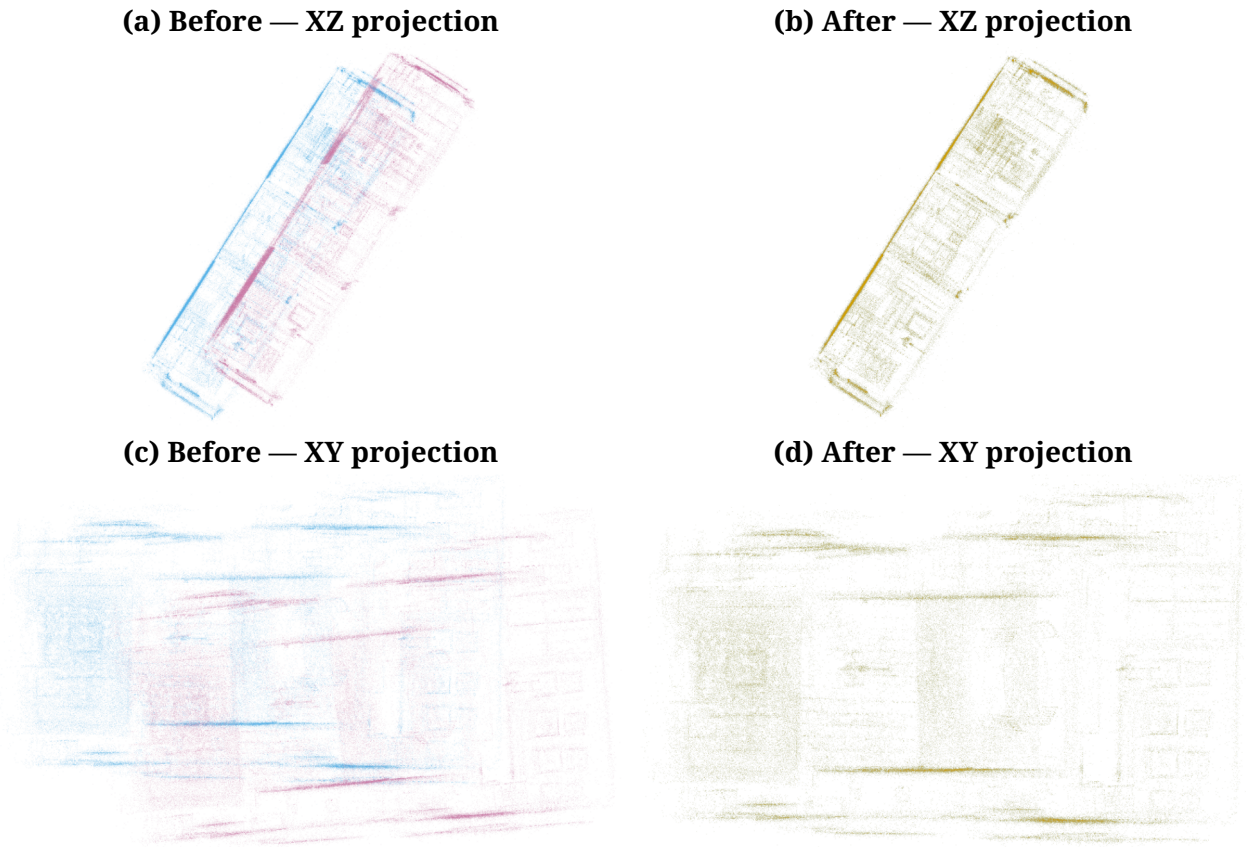

Figure 3: Splat-native registration at production scale, made inspectable. A real 3.18M-Gaussian indoor capture (the `drjohnson` Deep Blending scene) with a known SE(3) applied to the source (2.4 m translation, 5° yaw); the `fast` seed plus refine recovers it to 0.73 mm translation-norm and 0.0° rotation error over the full model. Each panel is an overlay of point positions in one of two orthogonal projections (XZ, top row; XY, bottom row), rendered without axis chrome for legibility; the applied transform magnitude is stated above. *Before* (a, c): the target (blue) and the offset source (pink) are visibly separated and rotated. *After* (b, d): the aligned output (orange) coincides with the target, the two overprinting to a single surface (olive). This is the known-transform-on-a-real-capture protocol of Table 4, scaled from a crop to a full room.

(22×), verified across two independent runs.

Controlled recovery and ablation. On a known-transform grid (3 seeds × {5°,30°,90°}, with the Sim(3) cells additionally sweeping {0.8,1.0,1.3} scale: 9 SE(3) + 27 Sim(3) cells), SPLATREG recovers 36/36 = 100% (SE(3) median 0.000°/0.10 mm; Sim(3) median 0.259°/2.93 mm/0.344% scale). Plain ICP, from either a centroid or super-Fibonacci init, solves all SE(3) cells but only 9/27 = 33% of Sim(3) cells (23–25% scale error on the rest): the LM Sim(3) solve is the load-bearing component for scale. On rigid SE(3) with a clean centroid offset, ICP is ∼1000× faster (0.03 s vs 33 s); the SDF residual provides scale and implicit-field robustness, not rigid speed. The warm-start tracking path runs ∼17 ms/frame.

Scaling to a full capture. Figure 3 makes the registration visually inspectable at production scale. On a real 3.18M-Gaussian indoor capture (the `drjohnson` Deep Blending scene) we applied a known SE(3) to the source — a 2.4 m translation and a 5° yaw — and ran the `fast` seed plus refine over the full model. The recovered transform matches ground truth to a 0.73 mm translation-norm error and 0.0° rotation error: before, the target (blue) and offset source (pink) are visibly separated and rotated; after, the aligned output (orange) coincides with the target across both orthogonal projections. This is the same known-transform-on-a-real-capture protocol as the splat-tool head-to-head (Table 4), scaled from a crop to a full-room, multi-million-Gaussian capture.

RQ1 answer. Yes: operating natively on the Gaussian representation costs no measurable recall against the learned point-based pipeline it builds on, decisively beats the existing splat tools, and delivers capabilities (scale, attribute-correct

outputs, covariance) the detour cannot; and a zero-shot front-end extends the same refine to cross-domain captures where the classical seed collapses.

5 RQ2: What Correctness Requires

A point set is closed under rigid motion; a Gaussian splat is not. A splat carries anisotropic covariances (as quaternion + per-axis scales), opacity, and view-dependent color as real-SH coefficients, and a registration tool must define what its recovered transform *does* to each. We found that the surveyed tools do not: they transform means and at most quaternions, and leave the rest undefined. We propose the following checklist, each item pinned by a released test, as the correctness bar for any splat registrar.

5.1 View-dependent color must be Wigner-rotated

The SH bands (`f_rest`) parameterize an appearance function on the sphere of view directions *in world coordinates*. Rotating a splat by R without rotating its SH leaves glossy highlights pointing at the old capture frame: geometry turns, appearance does not. The correct operation multiplies each degree-ℓ band by the real-basis Wigner-D block $D^{\ell}(R)$, which we build for any degree via the Ivanic–Ruedenberg recurrence [23, 24], produced directly in the 3DGS sign convention. Among the splat tools we surveyed (splatalign, GaussianSplattingRegistration, SuperSplat/SplatTransform), none rotate SH.

Because a renderer would only show the error qualitatively, we lock the math renderer-free against an independent hand-coded 3DGS basis evaluator (13 tests): evaluating rotated coefficients at direction d equals evaluating the originals at $R^{-1}d$ for degree ≤ 3 and random rotations to a measured ~2.4e-15 in float64 (Fig. 4a, against the error of skipping the rotation); the degree-1 block equals its signed-permutation closed form exactly (atol 1e-12); $D(I) = I$, $D(R_1R_2) = D(R_1)D(R_2)$, and orthogonality hold to <1e-10; under Sim(3) the color rotation uses the de-scaled R (atol 1e-5); and a rotated full-SH stack round-trips through PLY exactly (atol 1e-6).

5.2 Scale conventions must be normalized

3DGS assets store per-axis scales either linearly or as logarithms depending on the producing tool and pipeline stage. Fusing splats without normalizing the convention concatenates incompatible parameterizations and labels them with one flag, silently mis-exponentiating the odd one out; nothing crashes, the merged splat is simply wrong. SPLATREG's `merge` and bundle fusion normalize every piece to the reference's log-scale convention before concatenation, pinned

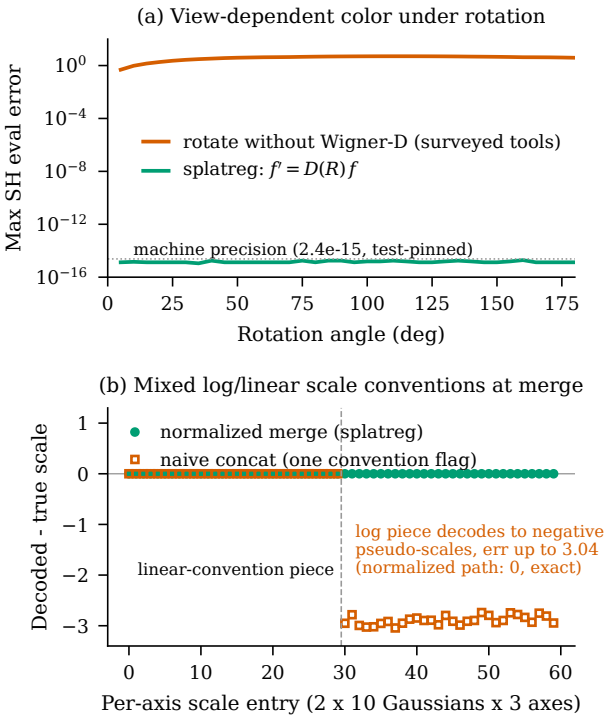

Figure 4: The RQ2 correctness checklist, computed from the released library on the setups its tests pin. (a) Maximum error of the view-direction appearance function under rotation, degree-3 SH in float64: with the real-basis Wigner-D rotation $f' = D(R)f$, evaluating the rotated coefficients at d equals evaluating the originals at $R^{-1}d$ at machine precision (~2e-15, flat in angle), while baking the rotation without rotating SH (the behavior of every surveyed splat tool) leaves an $O(1)$ appearance error at any nontrivial angle: geometry turns, highlights do not. (b) Mixed log/linear scale conventions at merge, the regression-test setup: concatenating raw scales under a single convention flag silently decodes the log-convention piece to negative pseudo-scales (error up to 3.04 against true scales of 0.05 to 0.07), while SPLATREG's convention normalization is exact. Nothing crashes in either failure, which is why both must be unit-pinned.

by a regression test that fuses a mixed-convention pair and checks the result (`test_merge_preserves_scales_across_log_convention`); Fig. 4b shows the silent corruption on that exact setup. We highlight this because it is exactly the kind of error an end-to-end Chamfer benchmark can miss if its inputs happen to share a convention.

5.3 Transform baking must be total and consistent

The align-without-merge workflow (`apply_transform`, and the `splatreg align` CLI) bakes a recovered SE(3)/Sim(3) into a splat that remains a valid standalone asset: means are

mapped by $sR + t$, orientation quaternions are left-composed with R, per-axis scales are multiplied by s, and the SH stack is Wigner-rotated, all in one operation, so the output opens correctly in any viewer. Partial baking (means only, or means + quaternions) produces a file that looks aligned in a point-cloud sense and is wrong as a splat. The CLI, the merge path, and the public API all route through the same audited function, and the PLY round-trip tests pin the end-to-end semantics. The full suite stands at 155 passing tests, including the Jacobian audit, Lie-group ops, solver, IO round-trips, CLI, photometric/exposure/ladder, SH rotation, pose covariance, and the zero-shot seed and its confidence-gate fallback.

RQ2 answer. Splat-native correctness is a representation-semantics problem, not an accuracy problem: three operations that have no point-cloud analogue (SH rotation, scale-convention normalization, total transform baking) must be implemented and, because their failure modes are silent, must be pinned by direct unit tests rather than end-to-end benchmarks.

6 RQ3: Hypothesis-Stage Gains Do Not Transfer

MAC [7] replaces RANSAC-style hypothesis generation with maximal-clique consensus over a second-order (SC^2) compatibility graph. In its own evaluation it lifts GeoTransformer from 92.0 to 95.7 recall on 3DMatch and from 75.0 to 78.9 on 3DLoMatch (its Table 3), with larger gains over weaker descriptors. If hypothesis quality were the binding constraint in our pipeline, MAC should lift our official-split numbers too. It does not, and the mechanism is instructive.

6.1 A faithful reimplementation, validated where MAC should win

We reimplemented MAC in pure PyTorch + networkx: the rigidity compatibility graph $\big|\,\|p_i - p_j\| - \|q_i - q_j\|\,\big| < \gamma$, re-weighted by the SC^2 second-order measure $w_2 = s \odot (S \cdot S)$ (chance-compatible outlier pairs share no common neighborhood, so their weight collapses), Bron–Kerbosch maximal cliques as consensus hypotheses, weighted SVD (Kabsch) per clique, an inlier-count winner refit, then the standard overlap-aware ICP polish. Worst-case caps mirror the paper's: ≤1000 correspondences, per-node degree cap 48, a 10k-clique + 4 s enumeration budget, and node-guided selection to ≤64 hypotheses. We extend it to Sim(3) (the paper is SE(3)-only) via a median pairwise-distance-ratio scale, de-scale, SE(3) MAC, and a residual scale refit.

On synthetic contaminated correspondence sets (200 correspondences, 40° true rotation, 3 mm inlier noise), MAC matches the RANSAC engine at 30/60/90% random outliers (0.04–0.16° both) and separates exactly where its theory predicts: with a *structured decoy*, a reflection-consistent outlier cluster whose preserved pairwise distances form a large compatible component that out-degrees the true inliers, the RANSAC engine fails at 78.0° while MAC enumerates both consensus cliques and the true one wins on inlier count, landing 0.16° (Table 5, Fig. 5b). All-outlier inputs return an honest `success=False` identity, never a silent wrong pose.

Correspondence set	MAC	RANSAC
30% random outliers	0.04°	0.04°
60% random outliers	0.16°	0.16°
90% random outliers	0.16°	0.16°
60% outliers + struct. decoy	0.16°	0.16°
90% outliers + struct. decoy	**0.16°**	**78.0° (fail)**
Sim(3), 50% outliers, s=1.7	0.05°, 0.02% scale	–

Table 5: Rotation error on synthetic contaminated correspondence sets. MAC and the RANSAC engine tie under random outliers; MAC wins decisively only in the multi-consensus decoy regime it was designed for. Runtime at 500 correspondences: 0.09 s.

Split (official pairs)	LGR (default)	MAC	Δ
3DMatch RR (1279)	91.5/93.5 (1196)	91.7/93.8 (1200)	+0.2/+0.3
3DLoMatch RR (1726)	72.5/74.4 (1285)	72.1/74.6 (1287)	−0.4/+0.2
3DLoMatch RRE/RTE	3.07°/0.099 m	3.18°/0.101 m	≈ tie
Median ms/pair	284 / 302	430 / 450	∼+50%

Table 6: MAC vs LGR hypothesis stages on the full official splits, identical forward pass, voxel, and refine. Recall as mean / pooled % with pooled success counts in parentheses. Every recall delta is within 4 pairs.

6.2 The official-split verdict: no net change

We then ran MAC as the hypothesis stage over GeoTransformer's learned correspondences at the paper's 0.10 m inlier threshold, on the full official splits. Both arms share the model forward pass, the native 0.025 voxel, and the same residual-gated refine, so the *only* difference is the hypothesis stage (LGR vs MAC); the LGR arm reproduces our published numbers exactly. Table 6 and Fig. 5a show the result: every delta is within ±4 pairs of LGR, at roughly +50% median runtime. MAC genuinely engaged on 100% of pairs (zero LGR fallbacks; one truncated enumeration across both splits; on 3DLoMatch the median pair yields 3830 correspondences → 2555 maximal cliques → 64 hypotheses → 602 consensus inliers, and on 3DMatch 5137 → 2565 → 64 → 803).

6.3 Mechanism, and the generalizable claim

Three factors explain the null result, stated in order of importance. (1) At native voxel resolution, GeoTransformer's correspondence sets are already *consensus-dominated*: the

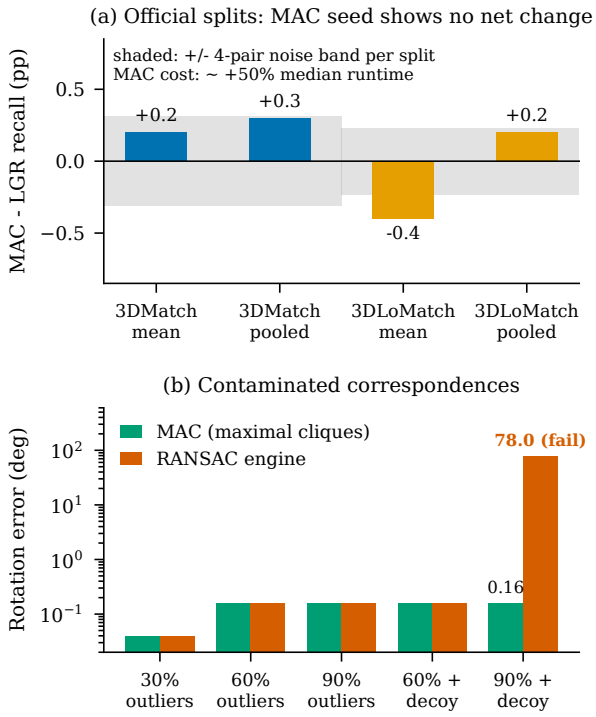

Figure 5: The RQ3 verdict in one figure: no net change on the official splits, decisive on decoys. (a) Recall delta of the MAC hypothesis stage relative to LGR on the full official splits (identical forward pass, voxel, and refine): every delta, mean-of-scenes and pooled, lies inside the shaded ±4-pair noise band (±0.31 pp of 1279 pairs on 3DMatch, ±0.23 pp of 1726 on 3DLoMatch), at roughly +50% median runtime. (b) Rotation error on synthetic contaminated correspondence sets (200 correspondences, 40° true rotation, 3 mm inlier noise): MAC ties the RANSAC engine at every random-outlier level and separates only in the multi-consensus structured-decoy regime it was designed for, 0.16° versus a 78.0° failure at 90% outliers plus decoy.

median pair carries 600–800 MAC-consensus inliers out of the ≤1000 graphed correspondences, i.e. 60–80% inliers. In that regime any reasonable hypothesis stage finds the same pose; the multi-consensus, high-outlier regime where MAC provably wins (Table 5) simply does not occur on these splits. (2) The residual-gated refine sits on top of *both* arms and absorbs seed-level differences, whereas the MAC paper compares raw hypothesis stages. (3) Our implementation caps the graph at 1000 correspondences (a deterministic subsample of the ~4–5k available); the paper runs richer sets.

The default therefore stays LGR (equal recall, ~35% faster), and `init="mac"` ships as the contaminated-correspondence tool it validated to be, not as a recall booster. The generalizable claim: *published hypothesis-stage gains are measured against weaker receiving pipelines (weaker correspondences, no gated refinement) and should be presumed pipeline-conditional until re-measured inside the receiving system.* The 3.7–3.9 point lift was real in its original setting; in ours, the headroom it exploited had already been consumed upstream (native-voxel correspondences) and downstream (gated refinement).

7 RQ4: Is the Pose Covariance Trustworthy — and Useful?

Every LM solve in SPLATREG exposes the undamped $J^\top W J$ information matrix and its scaled inverse as a pose covariance (Sec. 3); the library ships it as a deliverable for pose graphs and abstention. A deliverable is only as good as its calibration, so we ask two questions of it on real data: is it calibrated, and can it gate a downstream decision? The answers split cleanly — yes after recalibration, and no for the decision we tested — a second negative result with the same silent-failure moral as RQ2 and RQ3.

7.1 The raw covariance is over-confident; conformal recalibration fixes it

We run the full pipeline (robust seed → gated point-to-plane ICP LM refine → accept-if-better gate) on the real 3DMatch and 3DLoMatch fragments and, at each accepted pose, read the LM/Laplace pose covariance together with the Mahalanobis-squared pose error against ground truth. Raw, the covariance is grossly over-confident: nominal-90% coverage is 0.005 on 3DMatch and 0.000 on 3DLoMatch, with a median Mahalanobis^2 of 2779 and 2654 against $\chi^2_{6,0.9} = 10.64$. A leave-one-out split-conformal scale on the Mahalanobis scores [22] restores nominal coverage — 0.900 on 3DMatch (over n=902 successful pairs of 1279) and 0.905 on 3DLoMatch — at strong n (Table 7, Fig. 6a). The over-confidence transfers from, and is more extreme than, the synthetic case (the full synthetic calibration ablation — coverage versus overlap for SE(3) and Sim(3), and the over-confidence signature — is Appendix D), which is the strongest possible confirmation of the diagnostic. We note one caveat: the 3DLoMatch success rate in this particular pipeline is low (0.127) because it used the classical FPFH seed rather than a learned one; the calibration finding is a property of the covariance, not the seed, and holds identically raw→conformal on both splits.

7.2 But the covariance cannot gate fusion — the residual can

The intended payoff of a calibrated registration covariance is a fusion gate: merge a source splat into a target only when the recovered pose is certain enough, abstain otherwise. We

Split	recall	raw@90	conf.@90	med. M^2
3DMatch (902/1279)	0.705	0.005	**0.900**	2779
3DLoMatch (220/1726)	0.127	0.000	**0.905**	2654

Table 7: Pose-covariance calibration on real 3DMatch/3DLoMatch (full pipeline). Raw LM/Laplace covariance is grossly over-confident (coverage $\ll 0.9$; median Mahalanobis^2 $\gg \chi^2_{6,0.9} = 10.64$); leave-one-out split-conformal recalibration restores nominal 0.90 coverage at strong n.

tested this directly on n=112 real low-overlap fragment pairs (GT overlap 0.10–0.40), fusing with the full pipeline plus conformal recalibration and scoring by symmetric Chamfer distance to the ground-truth-aligned union. The result is our second negative (Table 8, Fig. 6b).

Covariance-aware abstention does lower mean Chamfer against naive always-fuse (0.146 vs 0.174 m), but only by collapsing to *blanket abstention*: in this failure-saturated regime the robust seed registers just 17.9% of pairs (20/112), so never fusing already beats always fusing, and every principled gate — a fixed calibrated-90% pose radius, a 5-fold CV threshold — lands exactly on always-abstain. There is almost no selective headroom (a perfect label-oracle gate reaches 0.1414, only 4.4 mm below blanket abstention), and the covariance captures 0.3 mm of it. The mechanism is the same as RQ3's, now measured directly: the pose covariance does not rank the failed registrations as more uncertain — AUC(uncertainty, failure) = 0.578 (≈ chance), Spearman(uncertainty, RTE) = 0.045, and an oracle threshold even prefers the backwards direction. The genuinely discriminative signal is the *LM residual*: AUC(residual, failure) = 0.773, Spearman = 0.349. BLUE information-form blending in the overlap barely moves Chamfer (0.171 vs 0.174): it can smooth co-located surfels but cannot repair a wrong global pose. The premise itself is sound — stratified by outcome, fusing wins on the 20 good poses (0.026 vs 0.051 m) and abstaining wins on the 92 failures (0.166 vs 0.206 m) — so a gate that could separate the two would help; the covariance simply is not that gate. Appendix E gives the full method table and the outcome- and overlap-band breakdowns.

RQ4 answer. The pose covariance the library exposes is a calibratable diagnostic: grossly over-confident raw, nominal after a split-conformal fix. It is not, however, a selective fusion signal on real low-overlap data — every uncertainty gate degenerates to blanket abstention and the covariance does not discriminate failures. The load-bearing signal is the LM residual, which does (AUC 0.773). We therefore report the covariance honestly as calibrated uncertainty for pose-graph edge weighting and abstention thresholds, and point downstream fusion at a residual gate. As in RQ2 and RQ3, the failure is silent — a wrong fusion never crashes — and the lesson is the same: choose the signal that actually ranks the failure, and pin it with a direct measurement.

Fusion policy (n=112)	Chamfer (m) ↓
naive (always fuse)	0.174
BLUE overlap blend	0.171
always abstain (never fuse)	0.146
covariance / CV gate	0.146 (= abstain)
covariance oracle (optimistic)	0.146
label-oracle gate (unachievable)	**0.141**

Table 8: Real low-overlap splat fusion (GT overlap 0.10–0.40), mean symmetric Chamfer to the GT-aligned union. Every principled covariance/CV gate collapses to blanket abstention; even a perfect label-oracle gate is only 4.4 mm better. Failure discrimination: AUC(covariance, fail) = 0.578 (≈ chance) vs AUC(LM residual, fail) = 0.773.

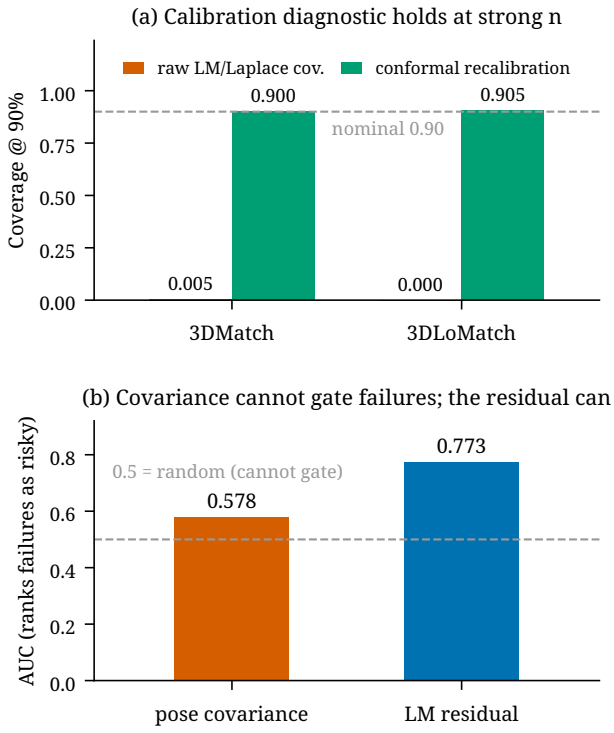

Figure 6: The RQ4 pose-covariance study. (a) The calibration diagnostic holds at strong n: raw LM/Laplace nominal-90% coverage is 0.005 / 0.000 on real 3DMatch / 3DLoMatch, and split-conformal recalibration restores 0.900 / 0.905. (b) But the covariance cannot gate failures: its failure-discrimination AUC (0.578) is essentially chance, so every uncertainty gate degenerates to blanket abstention, whereas the LM residual ranks failures well (AUC 0.773) and is the load-bearing gate.

8 Methods Note: A w^3 Bug as Benchmark Hygiene

In the spirit of RQ2's silent-failure theme, we report a bug in our own code that an end-to-end suite did not catch, found

by independent review. The SDF residual pre-multiplied its residual and Jacobian by its weight w while the solver also folded in $\sqrt{w}$, so the effective least-squares contribution scaled as w^3: the default stack ran the SDF term at $0.3^3 \approx 0.027$ instead of 0.3. The bug survived the test suite because the clean-synthetic recovery benchmark is ICP-dominated: with a strong ICP term and clean geometry, a nearly-disabled SDF term still passes a 36/36 recovery gate. (The official 3DMatch/3DLoMatch numbers are unaffected: the learned path never uses the SDF residual.) The fix is pinned by a direct unit test asserting linear weight scaling (`test_sdf_weight_applied_once`). The lesson matches RQ2: objective-level and representation-level semantics need direct unit pins; end-to-end accuracy gates can be insensitive to them by construction. Appendix B gives the full least-squares mechanism, the exact factor, and why the recovery gate stayed green.

9 Pending Experiment, Stated Honestly

The natural published head-to-head is the GaussReg protocol [8]: the 82-scene ScanNet-GSReg test split, in which each scene is reconstructed twice from disjoint image subsequences into two independent 3DGS models and the task is to recover the relative Sim(3), scored by their exact RRE/RTE/RSE decomposition and wall time. Our harness mirrors their official evaluation code one-to-one, including their scale-aware error metric; the benchmark distribution (361 GiB) is downloading at the time of writing, and the experiment runs unchanged once it completes. It will test what our self-designed splat benches cannot: registration across *independently reconstructed* splats, with reconstruction noise on both sides, rather than a known transform applied to crops of one capture. One scoping note for fairness: the number-level comparison on that benchmark is against GaussReg itself (published coarse / coarse-to-fine RRE 2.827 / 1.851); PhotoReg [9] does not report on ScanNet-GSReg (its evaluation is image-quality-based on its own scenes), so any PhotoReg comparison stays method-level. We state what the experiment will test and do not project numbers.

10 Threats to Validity

Self-designed splat-tool benches. The splat head-to-heads (Table 4, merge) apply a known transform to crops of one real capture; ground truth is exact, but the inputs are not independent reconstructions, and we designed the benches ourselves. The pending ScanNet-GSReg experiment (Sec. 9) is the external check. **Zero-shot seed comparison isolates the seed, not the leaderboard.** The BUFFER-X vs classical numbers (Table 2, Fig. 2) share the lighter feature-align refine between the two seeds and use the RRE/RTE criterion, so they isolate the seed's contribution but are neither comparable to the covariance-weighted official numbers of Table 1 nor a 3DMatch leaderboard submission. **Single learned backbone; failure-saturated covariance regime.** The learned path and the RQ3 analysis use one matcher (GeoTransformer). The RQ4 fusion negative (Sec. 7) runs the classical FPFH seed on 3DMatch-as-isotropic-surfels at 0.10–0.40 overlap, where the registration rate is 17.9%; the covariance's non-selectivity is measured, but the residual-gated alternative it points to is indicated rather than yet demonstrated as a positive, and a higher-recall seed (e.g. BUFFER-X) or a higher-overlap regime would give more selective headroom. The calibration numbers (Table 7) are strong-n on 3DMatch but the 3DLoMatch success subset is thin. **Synthetic photometric and decoy evidence.** The exposure-compensation and ladder numbers (Sec. 3.3) come from mock-renderer and rendered-sphere protocols; the dense-overlap real-capture case is measured but neutral there by construction. Likewise, the regime where MAC wins (Table 5) is demonstrated only on synthetic contaminated correspondence sets; we never observed a natural decoy on the official splits, which is precisely the finding of RQ3 but also its evidential limit. **Recall ceiling inherited from the seed.** The official-split numbers refine the learned seed under a never-degrade gate, so recall is bounded by the matcher's; the contribution claimed is matching that ceiling natively while adding scale and splat semantics, not exceeding it. **ScanNet-GSReg not yet run.** The one published splat-registration benchmark (Sec. 9) is pending data arrival; until it runs, no number in this paper compares SPLATREG to GaussReg on shared inputs.

11 Conclusion

Registration can operate natively on 3D Gaussian splats at point-pipeline accuracy (RQ1), but doing so *correctly* is a representation-semantics problem with no point-cloud analogue, and we propose the SH-rotation / scale-convention / transform-baking checklist with test pins as the bar for future tools (RQ2). We add a zero-shot front-end (BUFFER-X, reproduced on a modern CUDA stack) that lifts seed-isolated recall far above the classical fallback, especially at low overlap where the classical seed collapses. And we obtained two calibrated negative results through direct measurement. First, a celebrated hypothesis-generation method, faithfully reimplemented and fully engaged, adds nothing inside a refine-equipped pipeline fed high-inlier learned correspondences (RQ3). Second, the pose covariance the library exposes is grossly over-confident yet conformally recalibratable, but does not gate real low-overlap fusion — every uncertainty gate collapses to blanket abstention — where the LM residual does (RQ4). Both negatives sharpen

rather than soften the positive contributions: what certifies and verifies against ground truth survives contact with real data; the prediction-oriented claims (a hypothesis-stage lift, a covariance-selective fusion gain) do not, and we report the distinction explicitly. The library, benchmarks, and tests that back every number are released, with the full modern-CUDA build recipe, the calibration and fusion breakdowns, and the exact reproduction commands in the appendices (Appendices A–F).

A The Modern-CUDA BUFFER-X Build Recipe

BUFFER-X's public release targets an older stack; this is the exact, sudo-free sequence that made it build and run on an RTX 5090 (sm_120, CUDA 12.8, torch 2.11, numpy 2.4), verified 2026-07-01. It produces the `init="bufferx"` zero-shot seed of Sec. 4.2. We give it in full because a zero-shot registrar is only useful if it runs on currently available hardware, and none of these fixes is discoverable from the upstream README.

1. **Repository location.** The BUFFER-X checkout must live at `splatreg/third_party_models/BUFFER-X` (a sibling of the `splatreg/` package, i.e. under repo-root `third_party_models/`), where the path resolver looks. Cloning one level deeper silently disables the path and falls the seed back to the classical `robust` init.

2. **Weights.** Download the pretrained checkpoints from Hugging Face (`Hyungtae-Lim/BUFFER-X`) into `snapshot/threedmatch/{Desc,Pose}/best.pth`. Both are *full-model* state dicts (keys prefixed `Desc./Pose.`) and must be loaded into the whole model, not into the `.Desc/.Pose` submodules: a sub-module load matches nothing under `strict=False` and silently runs on random weights (the real bug of Sec. 4.2, fixed so the loader consumes the full checkpoint).

3. **Additive Python dependencies.** `pip install easydict tbb tbb-devel`, leaving the torch/numpy versions untouched. The `tbb` wheels ship `tbb/tbb.h` under `$VENV/include`, so no `sudo apt install libtbb-dev` is required.

4. **Eigen (header-only).** `git clone --depth 1` the Eigen repository and add it to the include path used by the KPConv wrapper build; no system install is needed.

5. **Pure-torch shims for two CUDA-only dependencies.** BUFFER-X uses `knn_cuda.KNN` only at k=1 and `torch_batch_svd.svd`; both are replaced by tiny drop-in modules on the path — `torch.cdist+topk` for the nearest neighbour and `torch.linalg.svd` (natively batched, returning (U, S, V)) for the SVD — avoiding two fragile native builds.

6. **pointnet2_ops (real CUDA build).** Its `setup.py` hard-codes an ancient architecture list (`TORCH_CUDA_ARCH_LIST="3.7+PTX;5.0;..."`), which nvcc 12.8 rejects with `unsupported gpu architecture 'compute_37'`. Patch that line to `"12.0"` (sm_120) and `pip install --no-build-isolation .`

7. **KPConv cpp_wrappers (numpy-2.x C-API port).** With `CPLUS_INCLUDE_PATH=$VENV/include:$EIGEN` and `LIBRARY_PATH=$VENV/lib`, run `compile_wrappers.sh`. The `wrapper.cpp` files use the numpy-1.x C-API; port them to numpy 2.x by renaming `NPY_IN_ARRAY` → `NPY_ARRAY_IN_ARRAY` and casting the `*_array/res_*_obj` `PyObject*` handles to `PyArrayObject*` wherever `PyArray_NDIM/DIM/DATA` are used.

After these steps the path resolver finds the model, the loader returns real weights, and `init="bufferx"` produces genuine seeds (validated at 0.75 recall / 1.9° median RRE on real 3DMatch high-overlap pairs, matching the classical seed there; the decisive advantage is the low-overlap regime of Sec. 4.2).

B The w^3 Residual-Weight Bug, in Full

The SDF residual stage exposes a weight w that trades the Gaussian-SDF term against the ICP terms. The bug (Sec. 8): the stage pre-multiplied *both* its residual vector and its Jacobian by w, returning $r' = w\,r$ and $J' = w\,J$, while the LM driver *independently* applied the standard $\sqrt{w}$ weighting when assembling the normal equations. The two weightings compound. The SDF block's contribution to the Gauss–Newton system is

$$(\sqrt{w}\,J')^\top(\sqrt{w}\,J') = w\,J'^\top J' = w\,(w^2 J^\top J) = w^3\,J^\top J,$$

and to the gradient $(\sqrt{w}\,J')^\top(\sqrt{w}\,r') = w^3\,J^\top r$, so the term entered the solve at an *effective* weight of w^3 rather than w. At the default $w = 0.3$ the SDF residual therefore ran at $0.3^3 \approx 0.027$ — roughly 11× weaker than intended, nearly disabling it.

Why the end-to-end suite missed it. The clean-synthetic controlled-recovery benchmark (the 36-cell grid of Sec. 4) is ICP-dominated: with a strong point-to-plane term and noise-free geometry, a nearly-disabled SDF term still passes the 36/36 recovery gate at median 0.000°. The official 3DMatch/3DLoMatch numbers are unaffected for a different reason — the learned path never invokes the SDF residual

Fragment sampling	pts/frag	3DMatch RR	3DLoMatch RR
native 0.025 m (default)	∼19k	**91.5 / 93.5**	**72.5 / 74.4**
legacy 0.05 m (pre-voxel)	∼5k	86.3 / 89.1	55.3 / –
harness penalty		−5.2 / −4.4	−17.2 / –

Table 9: Registration recall (mean / pooled %) under native versus over-aggressive pre-voxelization, both with the identical SPLATREG learned path. The gap is a benchmark-harness artefact, not a property of the method. Pooled 3DLoMatch was not recorded for the legacy configuration.

at all. No accuracy gate in the suite was sensitive to the term's weight, so the bug survived until an independent review read the two weightings side by side. The fix applies the weight exactly once and is pinned by a direct unit test (`test_sdf_weight_applied_once`) asserting that the assembled least-squares contribution scales *linearly* in w. The episode is the RQ2 lesson turned on our own code: objective-level semantics — here, what a weight means inside the normal equations — must be pinned by a direct unit test, because an end-to-end accuracy gate can be structurally blind to them.

C The Voxel-Resolution Harness Artefact

GeoTransformer is trained and evaluated at its native fragment resolution (`init_voxel_size` = 0.025 m). Our initial harness pre-voxelized both fragments to 0.05 m before the matcher saw them. This appears to be a harmless efficiency setting but discards most of the points the network was trained to match: a typical 3DMatch fragment drops from ∼19k to ∼5k points, over 70% of them gone. Because the matcher's superpoint sampling and patch correspondences assume the denser input, coarse pre-voxelization starves it of candidates, and the loss falls hardest on low-overlap pairs where every surviving correspondence matters. Table 9 isolates the effect: restoring native resolution — a change *entirely* in the harness, with the registration method untouched — recovers 5.2 mean recall points on 3DMatch and 17.2 on 3DLoMatch.

The general point generalizes past our own harness: a preprocessing default that appears to be a harmless efficiency setting can silently subtract double-digit recall from a learned matcher, and any refinement-style method evaluated behind it inherits the penalty. Benchmark harnesses should feed a method its native input resolution and report the preprocessing explicitly.

	SE(3)		Sim(3)		mean M^2	
overlap	raw	conf.	raw	conf.	SE(3)	Sim(3)
1.00	0.535	0.933	0.484	0.876	11.7	14.0
0.60	0.225	0.903	0.213	0.903	19.1	22.5
0.35	0.142	0.877	0.122	0.887	25.2	29.4
0.20	0.097	0.900	0.092	0.897	29.8	34.4

Table 10: Synthetic coverage@90 versus overlap, raw LM/Laplace vs. split-conformal (global). Raw under-covers and degrades with overlap; conformal restores nominal for both groups. Mean Mahalanobis^2 (χ^2 dof 6 / 7) sits far above expectation — the over-confidence signature.

D Synthetic Calibration and Fusion Ablations

The real-data pose-covariance diagnostic of Sec. 7 confirms, and exceeds, a synthetic calibration study we ran first as a de-risk (pure `numpy/scipy`, no GPU or real data). We report it here because it establishes the *mechanism* of the over-confidence and explains why BLUE fusion cannot help on isotropic-surfel splats.

Coverage collapses with overlap; conformal restores it. Table 10 and Fig. 7 show raw LM/Laplace coverage of the ground-truth transform for a nominal-90% region. Raw coverage falls from 0.484 (Sim(3), full overlap) to 0.092 (0.20 overlap) — far below nominal and monotonically worse as overlap drops — while the mean Mahalanobis^2 climbs from 14.0 to 34.4 against a χ^2 expectation of 7, the direct over-confidence signature. Split-conformal recalibration restores ∼0.90 at every overlap for both groups. This is the synthetic precursor of the real result of Table 7 (raw 0.005 / 0.000 → conformal 0.900 / 0.905): the real over-confidence is more extreme, exactly as the synthetic trend predicts as overlap and inlier structure degrade.

The over-confidence is overlap-driven, not noise-driven. Sweeping the measurement-noise standard deviation over two decades leaves raw coverage flat (∼0.18) and mean Mahalanobis^2 flat (∼24, Table 11). This is correct, not a bug: the Laplace covariance is $\Sigma = \sigma^2 H^{-1}$, and both the information H and the GT residual scale with the same noise, so the noise magnitude cancels in the Mahalanobis ratio. The over-confidence comes from the *unmodelled* overlap/mismatch structure the Laplace noise model never sees — which is exactly why a distribution-free recalibration, rather than a better noise model, is the right tool.

Why BLUE fusion needs anisotropy. The real fusion test (Sec. 7) found BLUE information-form blending barely moves Chamfer, because its inputs are isotropic 2 cm surfels.

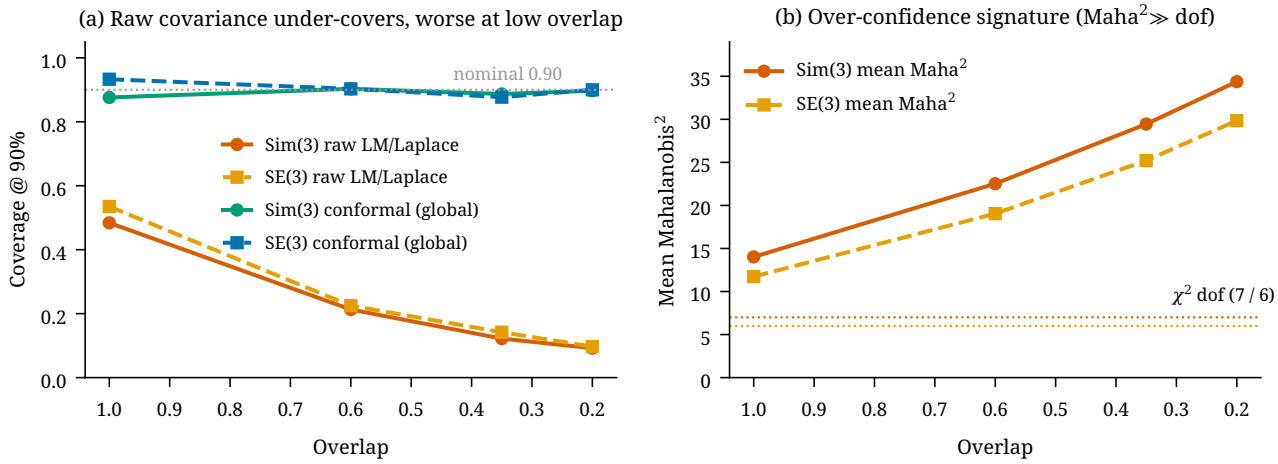

Figure 7: Synthetic calibration diagnostic (600 known-transform pairs per overlap; the synthetic precursor of Table 7). (a) Raw LM/Laplace nominal-90% coverage collapses from 0.484 at full overlap to 0.092 at 0.20 overlap for both SE(3) and Sim(3), while split-conformal recalibration holds ∼0.90 at every overlap. (b) The over-confidence signature: mean Mahalanobis^2 rises far above its χ^2 expectation (dof 6 for SE(3), 7 for Sim(3)) and worsens as overlap drops.

noise std	raw cov.	conf. (global)	mean M^2
0.002	0.179	0.912	24.3
0.005	0.176	0.862	24.4
0.010	0.180	0.884	24.3
0.020	0.180	0.876	24.3
0.050	0.180	0.892	24.3

Table 11: Coverage is flat across two decades of noise std (Sim(3), overlap 0.5, 500 pairs): an honest non-effect. Over-confidence is driven by overlap/mismatch, not noise magnitude, and conformal restores ∼0.90 throughout.

anisotropy ratio	naive err	BLUE err	gain
1	0.2263	0.2263	+0.0%
2	0.2743	0.2639	+3.8%
5	0.3741	0.3081	+17.6%
10	0.4892	0.3396	+30.6%
50	0.9805	0.4248	+56.7%
100	1.3489	0.4734	+64.9%

Table 12: Info-form (BLUE) fusion gain over the naive equal-weight mean versus covariance anisotropy (20,000 trials/setting; two anisotropic estimates of one point). The gain is 0% at isotropy and grows monotonically with the condition number.

Table 12 shows why on a controlled two-estimate merge: BLUE's gain over the naive equal-weight mean is exactly 0% when the two covariances are isotropic (the BLUE collapses to the mean) and grows monotonically with the anisotropy ratio, reaching +64.9% at condition number 100. The information-form merge pays off precisely in the anisotropic/degenerate regime — not on isotropic surfels, and never on a globally wrong pose. This is the mechanistic complement to the RQ4 fusion negative: the real test lives in the isotropic corner where BLUE provably cannot help.

E Real Low-Overlap Fusion: Full Breakdown

The RQ4 fusion negative (Sec. 7, Table 8) rests on n=112 real low-overlap 3DMatch/3DLoMatch fragment pairs (GT overlap 0.10–0.40, ≤14/scene, 3DMatch points as isotropic 2 cm surfels). Table 13 lists every fusion policy at full precision; Table 14 stratifies by registration outcome (the premise-check); Table 15 by overlap band. The failure-discrimination summary: AUC(covariance, fail) = 0.578 with Spearman(covariance, RTE) = 0.045, against AUC(residual, fail) = 0.773 with Spearman = 0.349; raw coverage@90 in this pipeline is 0.045.

F Reproduction Commands

Every number regenerates from the released library and benchmark harness. Paths are relative to the `splatreg` repository (library) and the `3dgs-registration` project (benchmarks); the interpreter is the project environment.

The drivers below live under `experiments/` in the `3dgs-registration` project unless noted; GPU runs pin a device with `CUDA_VISIBLE_DEVICES` and cap threads with `OMP_NUM_THREADS`.

- **Test suite** (155 passed, 8 skipped): `pytest -q` in the `splatreg`

Fusion policy (n=112)	Chamfer (m) ↓
naive (always fuse)	0.1738
BLUE overlap blend	0.1705
always abstain (never fuse)	0.1458
gate: fixed calibrated-90 radius	0.1458
gate: 5-fold CV on uncertainty	0.1458
oracle: best threshold on covariance	0.1455
oracle: best threshold on residual	0.1458
label-oracle (unachievable)	**0.1414**

Table 13: Every fusion policy at full precision. Each principled covariance/CV gate lands exactly on always-abstain; the covariance oracle captures 0.3 mm of the 4.4 mm headroom a perfect label-oracle would reach.

outcome	n	naive	abstain	BLUE
success (RRE<15°, RTE<0.3 m)	20	**0.0258**	0.0507	0.0240
fail	92	0.2060	**0.1665**	0.2024

Table 14: Chamfer stratified by registration outcome. On the 20 good poses fusing wins ~2×; on the 92 failures abstaining wins. The fuse-good / abstain-bad premise holds directionally — the covariance simply cannot separate the two (AUC 0.578).

overlap band	n	n_{fail}	naive	abstain	BLUE
0.10–0.20	63	62	0.2019	0.1632	0.1985
0.20–0.30	31	23	0.1502	0.1338	0.1463
0.30–0.40	18	7	0.1164	0.1059	0.1145

Table 15: Chamfer by overlap band. The naive-minus-abstain gap widens as overlap drops (more failures), confirming that the operative lever is "do not fuse unreliable registrations," which needs no covariance.

repo.

- **Synthetic calibration/fusion ablations** (App. D): `python -m splatreg_synth` and `run_ablations.py`.
- **Zero-shot recall** (Tables 2, 3; Fig. 2): driver `bufferx_recall_official.py` with `--split 3DMatch` (then `3DLoMatch`) at its default `--voxel 0.025`, one `--report` JSON per split. Requires the BUFFER-X build of App. A.
- **Pose-covariance calibration** (Table 7): driver `nfix_calibration.py` over both splits at its defaults (`--voxel 0.05`, `--alpha 0.10`).
- **Real low-overlap fusion** (Table 8; App. E): driver `fusion_real.py` with `--per-scene 14`.
- **Figures**: `make_figures.py` and `make_v2_figures.py` in `figures/` (the latter reads the committed result JSONs live).
- **Official learned-path and MAC recall** (Tables 1, 6): the repository's `benchmarks/` harness at the native 0.025 m voxel.

References

[1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 139:1–139:14, 2023.

[2] V. Ye, R. Li, J. Kerr, M. Turkulainen, B. Yi, Z. Pan, O. Seiskari, J. Ye, J. Hu, M. Tancik, and A. Kanazawa, "gsplat: An open-source library for Gaussian splatting," *Journal of Machine Learning Research*, vol. 26, no. 34, pp. 1–17, 2025.

[3] N. Keetha, J. Karhade, K. M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, and J. Luiten, "SplaTAM: Splat, track & map 3D Gaussians for dense RGB-D SLAM," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[4] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian splatting SLAM," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[5] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] M. Seo, H. Lim, K. Lee, L. Carlone, and J. Park, "BUFFER-X: Towards zero-shot point cloud registration in diverse scenes," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[7] X. Zhang, J. Yang, S. Zhang, and Y. Zhang, "3D registration with maximal cliques," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] J. Chang, Y. Xu, Y. Li, Y. Chen, and X. Han, "GaussReg: Fast 3D registration with Gaussian splatting," in *European Conference on Computer Vision (ECCV)*, 2024.

[9] Z. Yuan, T. Zhang, M. Johnson-Roberson, and W. Zhi, "PhotoReg: Photometrically registering 3D Gaussian splatting models," *arXiv preprint arXiv:2410.05044*, 2024.

[10] "splatalign: ICP alignment for 3D Gaussian splatting temporal captures." https://github.com/terminusfilms/splatalign. GitHub repository, accessed June 2026.

[11] "GaussianSplattingRegistration: Global and local registration of Gaussian point clouds." https://github.com/DarkTemplar91/GaussianSplattingRegistration. GitHub repository, accessed June 2026.

[12] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[13] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "PREDATOR: Registration of 3D point clouds with low overlap," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[15] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[17] M. Alexa, "Super-Fibonacci spirals: Fast, low-discrepancy sampling of SO(3)," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[19] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, no. 3, pp. 145–155, 1992.

[20] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.

[21] L. Zhu, Y. Li, E. Sandström, S. Huang, K. Schindler, and I. Armeni, "LoopSplat: Loop closure by registering 3D Gaussian splats," in *International Conference on 3D Vision (3DV)*, 2025.

[22] A. N. Angelopoulos and S. Bates, "Conformal prediction: A gentle introduction," *Foundations and Trends in Machine Learning*, vol. 16, no. 4, pp. 494–591, 2023.

[23] J. Ivanic and K. Ruedenberg, "Rotation matrices for real spherical harmonics. Direct determination by recursion," *The Journal of Physical Chemistry*, vol. 100, no. 15, pp. 6342–6347, 1996.

[24] J. Ivanic and K. Ruedenberg, "Additions and corrections: Rotation matrices for real spherical harmonics," *The Journal of Physical Chemistry A*, vol. 102, no. 45, pp. 9099–9100, 1998.