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 means either a manual gizmo 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 nobody defines. We built SPLATREG, an open pure-PyTorch library, to find out what registering splats natively actually takes, and this paper reports the three things we learned. 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, 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; 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. MAC keeps a decisive edge exactly where its theory says it should, on contaminated correspondence sets (0.16° vs a 78° failure on a structured decoy). We argue published hypothesis-stage gains should be presumed pipeline-conditional until re-measured inside the receiving system. 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 want to combine captures today either drag a manual gizmo (the dominant editor ships no automatic registration; its maintainer states "you can't combine splats right now"[1]), or export the Gaussian means as a point cloud and run a classical pipeline, discarding anisotropy, opacity, and view-dependent color, and then re-import a transform whose effect on those discarded attributes is left 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]. Three 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 audited analytic Jacobian over SE(3) and Sim(3), beneath interchangeable seed generators (FPFH+RANSAC, learned correspondences, 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).

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: rotate the points and you are done. A Gaussian

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

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

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 [6], 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 it engages on every pair, and measured it as a wash on both official splits, 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).

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. 7), and we describe honestly the one pending experiment, a head-to-head against GaussReg on its ScanNet-GSReg benchmark, without projecting numbers (Sec. 8).

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 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.
3. **A negative result with mechanism:** MAC hypothesis generation, reimplemented faithfully and engaged on 100% of pairs, is a wash on both official splits 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.
4. **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 [7] 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 [8] 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` [9] runs multi-scale point-to-point ICP from identity, and `GaussianSplattingRegistration` [10] wraps Open3D [11] 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 registration. Our learned seed builds on GeoTransformer [5] and its local-to-global registration (LGR), evaluated under the canonical 3DMatch [12] / 3DLoMatch [13] protocol with the covariance-weighted error of Choi et al. [14]. Classical seeds use FPFH [15] with RANSAC, and our global fallback sweeps super-Fibonacci SO(3) samples [16]. MAC [6] represents the hypothesis-generation line we test in RQ3. ICP variants [17, 18] underlie the refinement stage, and Umeyama's closed form [19] anchors similarity fitting.

Registration needs inside 3DGS systems. GS-SLAM systems increasingly need splat-level registration as a subroutine: LoopSplat [20] 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.

SH rotation. Rotating real spherical harmonics by Wigner-D matrices is classical [21,22]; 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:

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

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 $J^\top WJ$ 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), `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 ×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 [12–14]: 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 $e^\top Ce \leq 0.2^2$ from `gt.info`. Table 1 and Fig. 1a summarize.

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

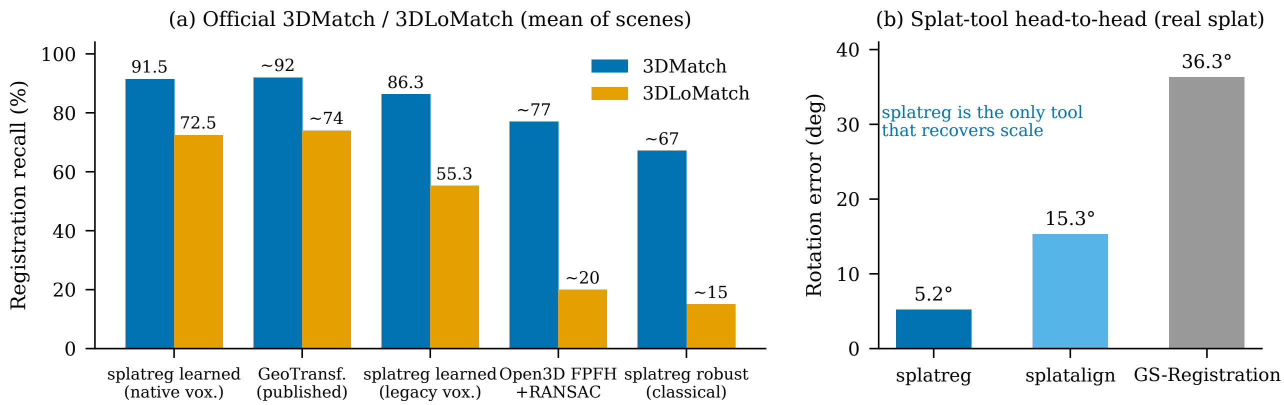

Figure 1: The RQ1 accuracy landscape. (a) Registration recall (mean of scenes) on the official 3DMatch and 3DLoMatch protocols: the splat-native SPLATREG `learned` path (91.5 / 72.5) matches the published GeoTransformer baseline it refines (~92 / ~74) while adding Sim(3) scale, splat-native attribute 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.

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.

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 2: 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.

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. We flag this because the same artefact would silently penalize any refinement-style method evaluated behind an aggressive pre-voxelization.

4.2 Against splat-registration tools

On a real captured splat under a known ground-truth Sim(3) (Table 2, Fig. 1b), SPLATREG achieves 5.2° rotation error (SE(3)) and 15.7 mm translation versus 15.3° for `splatalign` (ICP from identity) and 36.3° for `GaussianSplattingRegistration` (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.

4.3 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 (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. Honestly, on rigid SE(3) with a clean centroid offset, ICP is ∼1000× faster (0.03 s vs 33 s); the SDF residual buys scale and implicit-field robustness, not rigid speed. The warm-start tracking path runs ∼17 ms/frame.

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.

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 [21, 22], 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 $\leq$ 3 and random rotations to a measured ∼2.4e-15 in float64 (Fig. 2a, 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).

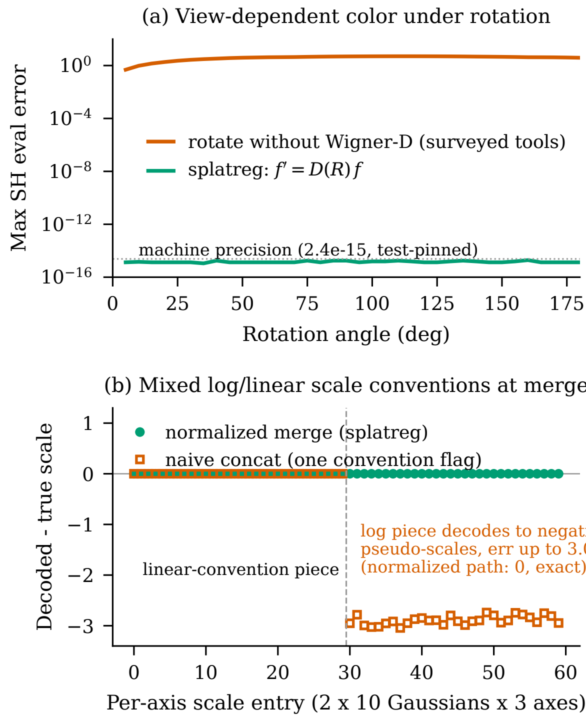

Figure 2: The RQ2 correctness checklist, computed from the released library on the setups its tests pin. (a) Maximum error of the view-direction appearance function under rotation, degree-3 SH in float64: with the real-basis Wigner-D rotation $f' = D(R)f$, evaluating the rotated coefficients at d equals 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.

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 by a regression test that fuses a mixed-convention pair and checks the result (`test_merge_preserves_scales_across_log_convention`); Fig. 2b 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 143 passing tests, including the Jacobian audit, Lie-group ops, solver, IO round-trips, CLI, photometric/exposure/ladder, SH rotation, and pose covariance.

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 [6] 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.

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 3: 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.

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 3, Fig. 3b). All-outlier inputs return an honest `success=False` identity, never a silent wrong pose.

6.2 The official-split verdict: a wash

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 4 and Fig. 3a 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 wash, stated in order of importance. (1) At native voxel resolution, GeoTransformer's correspondence sets are already *consensus-dominated*: the median pair carries 600–800 MAC-consensus inliers out of the ≤1000 graphed correspondences, i.e. 60–80% inliers. In that regime any sane hypothesis stage finds the same pose; the multi-

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 4: 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.

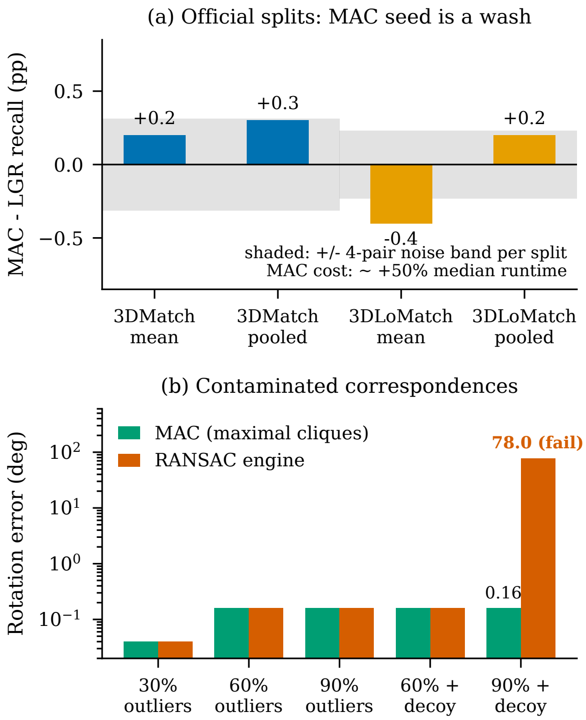

Figure 3: The RQ3 verdict in one figure: a wash on the official splits, decisive on decoys. (a) Recall delta of the MAC hypothesis stage relative to LGR on the full official splits (identical forward pass, voxel, and refine): every delta, mean-of-scenes and pooled, lies inside the shaded ±4-pair noise 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.

consensus, high-outlier regime where MAC provably wins (Table 3) 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 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.

8 Pending Experiment, Stated Honestly

The natural published head-to-head is the GaussReg protocol [7]: 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 when it lands. 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 [8] 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.

9 Threats to Validity

Self-designed splat-tool benches. The splat head-to-heads (Table 2, 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. 8) is the external check. **Single learned backbone.** The learned path and the RQ3 analysis use one matcher (GeoTransformer). The mechanism of Sec. 6 predicts the same wash for any matcher producing 60–80% inlier correspondences, but we have not measured a second backbone, so the generalization is argued, not demonstrated. **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 3) is demonstrated only on synthetic contaminated correspondence sets; we never observed a natural decoy on the official splits, which is the point 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. 8) is pending data arrival; until it runs, no number in this paper compares SPLATREG to GaussReg on shared inputs.

10 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). Along the way we obtained a calibrated negative result: a celebrated hypothesis-generation method, faithfully reimplemented and fully engaged, adds nothing inside a refine-equipped pipeline fed with high-inlier learned correspondences, and we argue such gains should be presumed pipeline-conditional (RQ3). The library, benchmarks, and tests that back every number are released.

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] 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.

[7] 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.

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

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

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

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

[12] 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.

[13] 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.

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

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.