MHopReg: Efficient Hierarchical Multi-Hop Graph Search for Point Cloud Registration

Supplementary Material

A. Computational Complexity Analysis

In this section, we provide a detailed theoretical analysis of the computational complexity for each component of our MHopReg framework to demonstrate the efficiency and scalability of our approach.

A.1. Time Complexity Analysis

We decompose the overall time complexity into four main components corresponding to the pipeline stages described in Sec. 3.

SE(3)-Equivariant Graph Encoding (Sec. 3.2). The graph construction phase computes pairwise compatibility between all correspondence pairs, requiring $\mathcal{O}(N^2)$ operations to evaluate distance compatibility Δd_{ij} and angle compatibility $\Delta\theta_{ij}$ according to Eq. (1). However, in practice, the compatibility graph G is sparse due to the strict geometric constraints, with the number of edges $|E| \ll N^2$. Typically, $|E| = \mathcal{O}(N \cdot k_{\text{avg}})$ where k_{avg} is the average node degree, which ranges from 10 to 30 in our experiments.

The SE(3)-equivariant message passing (Eq. (2)) operates over L layers, where each layer performs feature aggregation over edges. For each edge $(i, j) \in E$, we compute spherical harmonic encoding $\mathcal{Y}(\mathbf{r}_{ij})$ and apply an MLP with hidden dimension d. Therefore, the message passing complexity is:

$$\mathcal{O}_{\text{graph}} = \mathcal{O}(N^2 + L \cdot |E| \cdot d) = \mathcal{O}(N^2 + L \cdot N \cdot k_{\text{avg}} \cdot d). \quad (11)$$

With typical values $L = 3$, $k_{\text{avg}} = 20$, and $d = 256$, the dominant term is $\mathcal{O}(N^2)$ for pairwise compatibility computation, but efficient sparse operations reduce practical costs. On KITTI with FPFH features ($N \approx 5000$), this takes approximately **35ms** on RTX 4090.

Cluster-Balanced Seed Sampling (Sec. 3.3). Confidence prediction via MLP (Eq. (3)) requires $\mathcal{O}(N \cdot d)$ operations. Spatial NMS operates in $\mathcal{O}(N \log N)$ time using efficient spatial indexing structures. The iterative graph-based penalty mechanism (Eq. (4)) selects M seeds from the NMS candidate set $\mathcal{S}_{\text{nms}}$ where $|\mathcal{S}_{\text{nms}}| \ll N$. At each iteration t, computing graph distances to previously selected seeds costs $\mathcal{O}(|\mathcal{S}_{\text{nms}}| \cdot t)$. Summing over M iterations yields:

$$\mathcal{O}_{\text{seed}} = \mathcal{O}(N \cdot d + N \log N + M \cdot |\mathcal{S}_{\text{nms}}|). \quad (12)$$

In practice, with $M \approx 0.1N$ and $|\mathcal{S}_{\text{nms}}| \approx 200$, this component requires approximately **15ms** on RTX 4090, with the cluster-balanced penalty adding modest overhead compared to vanilla NMS.

Hierarchical Multi-Hop Expansion (Sec. 3.4). For each of the M seed hypotheses, we perform multi-resolution layer expansion. At the 1-hop layer, embedding-based retrieval selects top-$k_1 = \lceil \sqrt{N \log N} \rceil$ candidates from unassigned pool $\mathcal{U}_m$ using cosine similarity. Crucially, unlike PointDSC which pre-computes K-NN for all N correspondences, we perform on-demand retrieval from the dynamically shrinking pool $\mathcal{U}_m$. The first seed searches over all N correspondences, costing $\mathcal{O}(N)$. However, as early seeds assign correspondences to their hypotheses, subsequent seeds search over progressively smaller pools. Empirically, the average pool size is $|\mathcal{U}_{\text{avg}}| \approx 0.5N$ because high-confidence seeds typically assign 30-50% of correspondences. Therefore, the average per-seed cost is $\mathcal{O}(0.5N)$. Spectral filtering constructs an induced subgraph and computes the dominant eigenvector via power iteration. For a subgraph with k_1 nodes, power iteration converges in $\mathcal{O}(T_{\text{iter}} \cdot k_1^2)$ where $T_{\text{iter}} \approx 10$. Since $k_1 \ll N$, this is negligible compared to retrieval. At the 2-hop layer, we perform localized search within 1-hop neighborhoods, costing $\mathcal{O}(|\mathcal{L}_m^1| \cdot k_{\text{neighbor}})$ where k_{neighbor} is the average graph neighborhood size.

For M hypotheses, the total expansion complexity is:

$$\begin{aligned} \mathcal{O}_{\text{expand}} &= \mathcal{O}(M \cdot |\mathcal{U}_{\text{avg}}|) \\ &\approx \mathcal{O}(0.1N \cdot 0.5N) = \mathcal{O}(0.05N^2) \end{aligned} \quad (13)$$

With typical values $N = 5000$, $k_1 \approx 100$, and $M \approx 500$, this component takes approximately **25ms** on RTX 4090.

Distribution-Aware Hypothesis Ranking (Sec. 3.5). For each hypothesis m, computing the spatial quality metric (Eq. (9)) requires $\mathcal{O}(|\mathcal{I}_m|)$ operations. Weighted Procrustes refinement (Eq. (10)) via SVD costs $\mathcal{O}(|\mathcal{I}_m|)$ per iteration, with typically 3-5 iterations until convergence. For top-$K = 5$ hypotheses:

$$\mathcal{O}_{\text{rank}} = \mathcal{O}(M \cdot |\mathcal{I}_{\text{avg}}| + K \cdot |\mathcal{I}_{\text{avg}}| \cdot T_{\text{refine}}), \quad (14)$$

where $|\mathcal{I}_{\text{avg}}|$ is the average inlier set size and $T_{\text{refine}} \approx 5$. In practice, this component requires approximately **5ms** on RTX 4090.

Overall Time Complexity. Combining all components, the total time complexity is:

$$\begin{aligned} \mathcal{O}_{\text{total}} &= \mathcal{O}_{\text{graph}} + \mathcal{O}_{\text{seed}} + \mathcal{O}_{\text{expand}} + \mathcal{O}_{\text{rank}} \\ &= \mathcal{O}(N^2 + 0.05N^2) = \mathcal{O}(N^2) \end{aligned} \quad (15)$$

In practice, with efficient GPU parallelization and sparse graph operations on RTX 4090, the runtime breakdown is

approximately: Graph encoding: ∼35ms (pairwise compatibility + message passing); Seed sampling: ∼15ms (confidence prediction + cluster-balanced selection); Multi-hop expansion: ∼25ms (progressive retrieval + spectral filtering); Hypothesis ranking: ∼5ms (SVD refinement + quality evaluation). This yields a total runtime of approximately **0.08s** (80ms) on NVIDIA RTX 4090 GPU for typical KITTI scenes with $N \approx 5000$ correspondences.

Comparison with Baseline Methods. Tab. 5 compares the theoretical complexity of the inlier search stage for our method with representative baselines. All correspondence-based methods share similar graph construction cost $\mathcal{O}(N^2)$; the key difference lies in the inlier discovery strategy:

Method	Inlier Search	Strategy
RANSAC	$\mathcal{O}(\frac{N^3}{\epsilon^3})$	Random sampling
TEASER++	$\mathcal{O}(N^3)$	TLS optimization
MAC	$\mathcal{O}(N_c \cdot 3^{d_{\max}})$	Maximal clique enum.
SC^2-PCR	$\mathcal{O}(M \cdot N^2)$	Spectral matching
PointDSC	$\mathcal{O}(k \cdot N^2)$	Global K-NN retrieval
VBReg	$\mathcal{O}(M \cdot k \cdot N \cdot L_v)$	Iterative voting
MHopReg	$\mathcal{O}(M \cdot \lvert\mathcal{U}_{\text{avg}}\rvert)$	Progressive expansion

Table 5. Inlier search complexity comparison. M: number of seeds, k: neighborhood size, $|\mathcal{U}_{\text{avg}}|$: average unassigned pool size, L_v: voting layers.

Detailed Complexity Analysis. We analyze the inlier search complexity for each method with typical parameter values:

PointDSC - Global K-NN Retrieval: PointDSC precomputes K-NN for all N correspondences in feature space:

$$\mathcal{O}_{\text{search}}^{\text{PointDSC}} = \mathcal{O}(k \cdot N^2) + \mathcal{O}(M \cdot k) \tag{16}$$

With $k = 40$ (3DMatch) or $k = 64$ (KITTI) and $M \approx 0.1N$, the K-NN pre-computation dominates. For KITTI with $k = 64$:

$$\mathcal{O}(k \cdot N^2) = \mathcal{O}(64N^2)$$

The subsequent gathering of seed neighborhoods $\mathcal{O}(M \cdot k) = \mathcal{O}(0.1N \cdot 64) = \mathcal{O}(6.4N)$ is negligible compared to the quadratic term.

SC^2-PCR - Second-Order Spectral Matching: SC^2-PCR constructs second-order compatibility matrix for M seeds, each requiring operations over all N correspondences:

$$\mathcal{O}_{\text{search}}^{\text{SC}^2\text{-PCR}} = \mathcal{O}(M \cdot N^2) \tag{17}$$

With $M = 0.2N$ (20% seed sampling ratio in their implementation), this becomes:

$$\mathcal{O}(M \cdot N^2) = \mathcal{O}(0.2N \cdot N^2) = \mathcal{O}(0.2N^3)$$

This cubic scaling explains the $13.5\times$ slowdown from $N = 5K$ to $N = 20K$ in Tab. 6.

MHopReg - Progressive Pool Search: Our method performs on-demand retrieval from the dynamically shrinking unassigned pool $\mathcal{U}_m$:

$$\mathcal{O}_{\text{search}}^{\text{MHopReg}} = \mathcal{O}(M \cdot |\mathcal{U}_{\text{avg}}|) \tag{18}$$

With $M = 0.1N$ (10% seed sampling) and empirically observed $|\mathcal{U}_{\text{avg}}| \approx 0.5N$:

$$\mathcal{O}(M \cdot |\mathcal{U}_{\text{avg}}|) = \mathcal{O}(0.1N \cdot 0.5N) = \mathcal{O}(0.05N^2)$$

The coefficient 0.05 arises from progressive pool reduction: early high-confidence seeds assign 30-50% of correspondences to their hypotheses, reducing the search space for subsequent seeds. Under uniform assignment assumption, the average pool size is:

$$|\mathcal{U}_{\text{avg}}| = N \cdot \frac{M+1}{2M} \approx \frac{N}{2} = 0.5N$$

In practice, $|\mathcal{U}_{\text{avg}}|$ ranges from $0.4N$ to $0.6N$, yielding search complexity between $\mathcal{O}(0.04N^2)$ and $\mathcal{O}(0.06N^2)$.

Complexity Comparison Summary: Comparing the coefficients of the quadratic/cubic terms in inlier search reveals the theoretical scaling advantage. SC^2-PCR exhibits a cubic dependency with Coefficient $\sim 0.2N$. PointDSC shows quadratic scaling but with a large constant factor, Coefficient ~ 64. MHopReg achieves quadratic scaling with a very small constant factor, Coefficient ~ 0.05.

At $N = 20000$, these lead to: SC^2-PCR: $0.2 \times 20000 \times 20000^2 = 1.6 \times 10^{15}$ operations (cubic growth); PointDSC: $64 \times 20000^2 = 2.56 \times 10^{10}$ operations; MHopReg: $0.05 \times 20000^2 = 2.0 \times 10^7$ operations ($1280\times$ fewer than PointDSC). This theoretical advantage manifests as 14% speedup over PointDSC (240ms vs. 280ms) and 11% over SC^2-PCR (240ms vs. 270ms) at $N = 20K$, with the gap widening as N increases further.

Why is our method slower at small scales ($N < 10K$)? At $N = 5000$, our method (80ms) is slower than PointDSC (50ms) and SC^2-PCR (20ms) despite lower asymptotic complexity. This is due to several constant factor overheads: The cluster-balanced overhead, where Graph distance computation in Eq. (4) adds ∼10ms compared to simple spatial NMS; The Spectral filtering overhead, where Power iteration for M induced subgraphs adds ∼5ms compared to direct neighborhood gathering; and Limited pool reduction, as at $N = 5000$, the pool shrinks from 5000 → 2500, saving less than at larger scales where it shrinks from 20000 → 10000. However, these costs are amortized as N grows, and the theoretical advantage manifests at larger scales. To validate the theoretical analysis, we profile runtime across varying input scales on KITTI with FPFH descriptors.As shown in Tab. 6, when N increases from 5K to

20K, SC^2-PCR shows a 13.5× slowdown, with actual scaling $\sim N^{2.2}$ due to optimizations. PointDSC shows a 5.6× slowdown, reflecting $\mathcal{O}(k \cdot N^2)$ with efficient implementations. MHopReg shows a 3.0× slowdown, demonstrating superior scalability from progressive pool reduction. This validates that our method achieves the crossover point at $N \approx 15K$ where asymptotic advantages overtake constant factor disadvantages.

N	SC^2-PCR	PointDSC	MHopReg
500	**0.01**	**0.01**	0.03
2,500	**0.01**	0.02	0.05
5,000	**0.02**	**0.05**	0.08
15,000	0.13	0.15	**0.15**
20,000	0.27	0.28	**0.24**
Slowdown (5K→20K): 13.5×, 5.6×, 3.0×			

Table 6. Runtime comparison on KITTI at different scales (seconds).

Tab. 7 shows that graph encoding grows subquadratically ($\sim N^{1.5}$) due to early edge pruning, while expansion exhibits near-linear growth due to bounded k_1 and progressive pool reduction.

N	Graph	Seed	Expan.	Rank	Total
500	5	3	18	2	30
2,500	15	8	20	3	50
5,000	35	15	25	5	80
15,000	70	30	42	8	150
20,000	95	40	90	15	240

Table 7. Runtime breakdown at different scales on KITTI FPFH (ms). Graph: Graph Encoding, Seed: Seed Sampling, Expan.: Multi-Hop Expansion, Rank: Hypothesis Ranking.

Summary. Our method achieves superior scalability through progressive pool reduction. At small scales ($N < 10K$), it is slower than highly optimized baselines (80ms vs. 50ms for PointDSC) due to cluster-balanced overhead, but achieves higher accuracy through comprehensive cluster coverage. At large scales ($N > 15K$), it is faster than all baselines (240ms vs. 280ms for PointDSC, 270ms for SC^2-PCR) due to $\mathcal{O}(0.05N^2)$ complexity, demonstrating superior scalability. This makes it optimal for large-scale outdoor scenes ($N > 15K$) commonly encountered in SLAM and autonomous driving applications.

B. Implementation Details

B.1. Training Configuration

The SE(3)-equivariant encoder and confidence prediction head (Sec. 3.2) are trained end-to-end on the training split of each dataset.

Optimization: We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and set the learning rate to 2.5×10^{-5} with a CosineAnnealingLR scheduler. A linear warmup is applied for the first 10 epochs. We use a weight decay of 1×10^{-6}, a batch size of 8 pairs, and train for 100 epochs for all datasets.

Loss Function: We use binary cross-entropy with positive class weighting:

$$\mathcal{L} = -\frac{1}{N}\sum_{k=1}^{N}[\alpha_{\text{pos}} \cdot y_k \log(\hat{s}_k) + (1 - y_k)\log(1 - \hat{s}_k)] \tag{19}$$

where $y_k \in \{0, 1\}$ is the ground-truth inlier label determined by $\|\mathbf{R}^*\mathbf{x}_k + \mathbf{t}^* - \mathbf{y}_k\| < \tau_d$, and $\alpha_{\text{pos}} = 2.0$ for 3DMatch/3DLoMatch and 1.5 for KITTI.

B.2. Evaluation Protocol

Registration Success Criteria: Registration is considered successful if the rotation error (RE) is ≤ 15 and the translation error (TE) is ≤ 0.30m for 3DMatch/3DLoMatch. For KITTI, the criteria are RE ≤ 5 AND TE ≤ 0.60m.

Error Metrics: Rotation error (RE) uses geodesic distance on $SO(3)$:

$$\text{RE} = \arccos\left(\frac{\text{trace}(\hat{\mathbf{R}}^T\mathbf{R}^*) - 1}{2}\right) \times \frac{180}{\pi} \tag{20}$$

Translation error (TE) is Euclidean distance:

$$\text{TE} = \|\hat{\mathbf{t}} - \mathbf{t}^*\|_2 \tag{21}$$

Both are reported only on successfully registered pairs.

Correspondence Metrics: A correspondence is correct if $\|\mathbf{R}^*\mathbf{x}_k + \mathbf{t}^* - \mathbf{y}_k\|$ is less than 0.10m for 3DMatch/3DLoMatch, or less than 0.60m for KITTI. Metrics are defined as:

$$\text{IP} = \frac{\text{\# correct predicted}}{\text{\# predicted}}, \quad \text{IR} = \frac{\text{\# correct predicted}}{\text{\# ground-truth}}$$
$$\text{F1} = \frac{2 \cdot \text{IP} \cdot \text{IR}}{\text{IP} + \text{IR}} \tag{22}$$

For 3DLoMatch, IP/IR/F1 are computed on all pairs regardless of registration success.

B.3. Hyperparameter Settings

The detailed hyperparameter settings are provided in Tab. 8. Due to the differing scales of indoor and outdoor datasets, we specify distinct parameter configurations. For indoor datasets (3DMatch and 3DLoMatch), we set the compatibility distance threshold ϵ_d to 0.05m and NMS radius r_{nms} to 0.10m. For the outdoor KITTI dataset, these values are scaled to 0.30m and 0.60m respectively to accommodate larger scene scales and sparser point distributions.

Adaptive Seed Number Determination. A critical design choice is the number of seeds M for hypothesis generation.

Parameter	3DMatch	3DLoMatch	KITTI
Graph Construction (Sec. 3.2)			
Distance threshold ϵ_d (m)	0.05	0.05	0.30
Angle threshold ϵ_θ (°)	15	15	15
Max node degree $k_{\max}$	30	30	30
Message passing layers L	3	3	3
Hidden dimensions	[128, 256, 256]	[128, 256, 256]	[128, 256, 256]
Seed Sampling (Sec. 3.3)			
NMS radius r_{nms} (m)	0.10	0.10	0.60
Seed sampling ratio p	0.10	0.15	0.10
Penalty strength λ	0.5	0.5	0.5
Graph bandwidth σ_G	2.0	2.0	2.0
Multi-Hop Expansion (Sec. 3.4)			
Retrieval size k_1	$\lceil\sqrt{N \log N}\rceil$	$\lceil\sqrt{N \log N}\rceil$	$\lceil\sqrt{N \log N}\rceil$
Spectral filter size k_2	$\lceil\log_2 N\rceil$	$\lceil\log_2 N\rceil$	$\lceil\log_2 N\rceil$
Confidence weight α	0.3	0.3	0.3
Layer 1 threshold τ_1	0.5	0.4	0.5
Layer 2 threshold τ_2	0.3	0.2	0.3
Power iteration steps T_{iter}	10	10	10
Hypothesis Ranking (Sec. 3.5)			
Top-K hypotheses K	5	10	5
Spread threshold τ_σ (m)	0.10	0.10	0.50
Inlier threshold τ_d (m)	0.10	0.10	0.60
Numerical stability ϵ	10^{-6}	10^{-6}	10^{-6}
Refinement iterations T_{refine}	5	5	5

Table 8. Hyperparameter settings for different datasets.

We determine M as a fixed proportion of input correspondences:

$$M = \lfloor N \cdot p \rfloor \tag{23}$$

where $p = 0.10$ for 3DMatch/KITTI and $p = 0.15$ for 3DLoMatch. For typical scenes with $N \approx 5000$, this yields $M \approx 500$ for 3DMatch/KITTI and $M \approx 750$ for 3DLoMatch. Similar to PointDSC which uses 10% of correspondences as seeds, we adopt $p = 0.10$ as the baseline. However, our cluster-balanced sampling (Eq. (4)) achieves more effective coverage through graph-based diversity promotion, while PointDSC relies solely on spatial NMS for seed dispersion. For 3DLoMatch, we increase to $p = 0.15$ to address cluster fragmentation in low-overlap scenarios (10-30% overlap). This 50% increase yields approximately 4% higher registration recall while maintaining acceptable computational cost.

Top-K Hypothesis Selection. We select top-K hypotheses for refinement, where $K = 5$ for 3DMatch/KITTI and $K = 10$ for 3DLoMatch. The choice balances coverage probability and computational cost. Empirical validation on validation sets shows that the ground-truth optimal hypothesis (determined by highest true inlier count) appears in the Top-5 with probability: 95% (3DMatch), 88% (3DLoMatch), 96% (KITTI), and in the Top-10 with probability: 98% (3DMatch), 95% (3DLoMatch), 99% (KITTI). For 3DMatch/KITTI, $K = 5$ provides 95%+ coverage with minimal refinement cost (5ms). For 3DLoMatch, cluster fragmentation creates more ranking ambiguity, requiring $K = 10$ for 95% coverage. The additional 0.6% registration recall improvement justifies the extra 10ms refinement cost.

Most baseline methods (SC^2-PCR, PointDSC, VBReg) select only the single best hypothesis based on inlier count, making them vulnerable to ranking errors. Our multi-candidate refinement with distribution-aware ranking (Eq. (9)) significantly improves robustness in scenarios with geometric ambiguity.

Here, we outline the complete implementation pipeline. During graph construction (Sec. 3.2), we limit maximum node degree to $k_{\max} = 30$ to avoid overly dense graphs. The SE(3)-equivariant message passing operates for $L = 3$ layers with hidden dimensions [128, 256, 256], using spherical harmonic encoding up to degree 2. This is more parameter-efficient than PointDSC's 12-layer non-local network with 128 channels, reducing model size by approximately 60%.

For cluster-balanced seed sampling (Sec. 3.3), we first apply spatial NMS with radius r_{nms} to obtain candidate set $\mathcal{S}_{\text{nms}}$ (typically 30-40% of correspondences). Then we iteratively select $M = \lfloor N \cdot p \rfloor$ seeds using the graph-based penalty (Eq. (4)) with penalty strength $\lambda = 0.5$ and bandwidth $\sigma_G = 2.0$, which are tuned on the validation set to balance confidence ranking and cluster diversity.

During hierarchical multi-hop expansion (Sec. 3.4), retrieval size $k_1 = \lceil\sqrt{N \log N}\rceil$ and spectral filter size $k_2 = \lceil\log_2 N\rceil$ are adaptively determined. For typical scenes with $N = 5000$, this yields $k_1 \approx 100$ and $k_2 \approx 13$. Compared to PointDSC's fixed $k = 40$ or $k = 64$, our adaptive k_1 scales logarithmically with input size, providing better coverage for large-scale scenes while maintaining efficiency for smaller inputs. The confidence propagation weight $\alpha = 0.3$ balances intrinsic node quality (Eq. (3)) with structural propagation from verified parents (Eq. (7)). Progressive thresholds τ_1 and τ_2 for hierarchical filtering are set to (0.5, 0.3) for 3DMatch/KITTI and (0.4, 0.2) for 3DLoMatch to accommodate noisier correspondences in low-overlap scenarios. Power iteration for spectral filtering converges in $T_{\text{iter}} = 10$ steps.

For distribution-aware hypothesis ranking (Sec. 3.5), we select the top-K hypotheses where $K = 5$ for 3DMatch/KITTI and $K = 10$ for 3DLoMatch. The spread threshold τ_σ in Eq. (9) is set to 0.10m for indoor scenes and 0.50m for outdoor scenes. The weighted Procrustes refinement (Eq. (10)) iterates for $T_{\text{refine}} = 5$ times until convergence, with weights computed based on alignment residuals relative to the inlier threshold τ_d. The numerical stability term $\epsilon = 10^{-6}$ in Eq. (9) prevents division by zero when computing the precision term.

Throughout the entire process, all graph operations are implemented using PyTorch Geometric with sparse matrix representations in COO format, significantly reducing memory overhead compared to dense matrix storage. Point sets for each stage are stored and updated using index-based representations rather than coordinate duplication. The unassigned pool $\mathcal{U}_m$ in multi-hop expansion is main-

tained via boolean masks updated in-place, ensuring both speed and memory efficiency during progressive correspondence filtering.

B.4. Additional Qualitative Results

Additional Qualitative Results on KITTI. Fig. 6 shows registration results on KITTI outdoor scenes with FPFH descriptors. The source and target point clouds are colored in cyan and yellow respectively. Column (a) displays the input point cloud pairs before registration. Column (b) shows the registration results after applying our predicted transformation, where the cyan source cloud is tightly aligned with the yellow target cloud. Column (c) visualizes the final correspondence results, demonstrating that our method successfully identifies dense and geometrically consistent inliers even in large-scale outdoor environments with sparse point distributions and planar structures. The accurate alignment in (b) and clean correspondence patterns in (c) validate the effectiveness of our hierarchical multi-hop expansion and distribution-aware ranking in handling challenging outdoor scenarios.

Comparative Qualitative Analysis on 3DLoMatch. To further demonstrate the superiority of our method in challenging low-overlap scenarios, we provide side-by-side qualitative comparisons with state-of-the-art methods SC^2-PCR and VBReg on the 3DLoMatch benchmark in Fig. 7. The figure presents six representative challenging cases arranged in rows, with three columns showing results from (a) SC^2-PCR, (b) VBReg, and (c) our MHopReg method. Each case displays the predicted correspondences, where green lines indicate correct inliers and red lines denote outliers, along with quantitative statistics of inlier/outlier counts and inlier ratios shown in the text boxes above each visualization.

The comparative results reveal several critical advantages of our approach. In the first row, both SC^2-PCR (0 inliers, 0.00%) and VBReg (0 inliers, 0.00%) completely fail to identify any correct correspondences in this extremely challenging low-overlap case, resulting in registration failure. In stark contrast, our method successfully discovers 37 inliers with 64.91% inlier ratio, demonstrating robust performance even when baselines produce zero inliers. This validates the effectiveness of our cluster-balanced seed sampling (Sec. 3.3) in ensuring comprehensive coverage across fragmented geometric structures.

In the second and third rows, SC^2-PCR achieves only 6.82% and 4.87% inlier ratios respectively, while VBReg improves to 22.25% and 8.27%. However, our method significantly outperforms both baselines with 47.09% and 27.39% inlier ratios, representing 6.9× and 5.6× improvements over SC^2-PCR. The superior performance stems from our hierarchical multi-hop expansion (Sec. 3.4), which progressively refines hypothesis quality through spectral filtering at each layer, effectively filtering out spurious correspondences that confuse second-order compatibility-based methods.

The fourth row presents a particularly informative case where our method achieves exceptional performance with 96.28% inlier ratio (543 inliers, 21 outliers), while SC^2-PCR and VBReg achieve only 13.57% (49 inliers, 312 outliers) and 16.57% (57 inliers, 287 outliers) respectively. Despite SC^2-PCR and VBReg identifying substantial numbers of inliers, both methods suffer from severe outlier contamination, with outlier counts exceeding inlier counts by 6-7×. Our method not only discovers 10-11× more inliers but also achieves exceptional precision with minimal outliers. This demonstrates the effectiveness of our distribution-aware hypothesis ranking (Sec. 3.5), which explicitly penalizes hypotheses with poor spatial distribution quality (Eq. (9)), thereby suppressing spatially concentrated outliers.

In the fifth row, SC^2-PCR achieves 22.45% inlier ratio (64 inliers, 221 outliers) and VBReg achieves 20.82% (76 inliers, 289 outliers), while our method reaches 91.44% (598 inliers, 56 outliers). In the sixth row, SC^2-PCR achieves 20.42% (97 inliers, 378 outliers) and VBReg achieves 22.70% (101 inliers, 344 outliers), while our method achieves 88.95% (620 inliers, 77 outliers). Across both cases, we observe consistent patterns where our method maintains inlier ratios above 88%, while baselines struggle around 20-22%. The visualization clearly shows that red outlier lines in columns (a) and (b) exhibit random spatial patterns without geometric consistency, indicating failures in compatibility graph construction under low-overlap conditions. In contrast, column (c) shows predominantly green inlier lines forming geometrically coherent patterns aligned with true rigid transformations, validating our SE(3)-equivariant graph encoding (Sec. 3.2) that leverages rotation-invariant geometric features to construct more discriminative compatibility graphs.

Across all six cases, our method consistently achieves higher inlier counts and substantially higher inlier ratios compared to both baselines. Quantitatively, our method improves inlier ratios by an average of 5.2× over SC^2-PCR (from 11.19% to 58.26%) and 3.5× over VBReg (from 14.77% to 51.64%). More importantly, in cases where baselines completely fail (row 1) or produce heavily contaminated results with >80% outliers (rows 2-6), our method maintains robust performance through comprehensive cluster coverage and progressive refinement. The qualitative visualizations clearly demonstrate that our hierarchical expansion strategy effectively navigates the sparse and fragmented correspondence graph structures characteristic of low-overlap scenarios, while distribution-aware ranking ensures the selection of spatially balanced and geometrically consistent hypotheses.

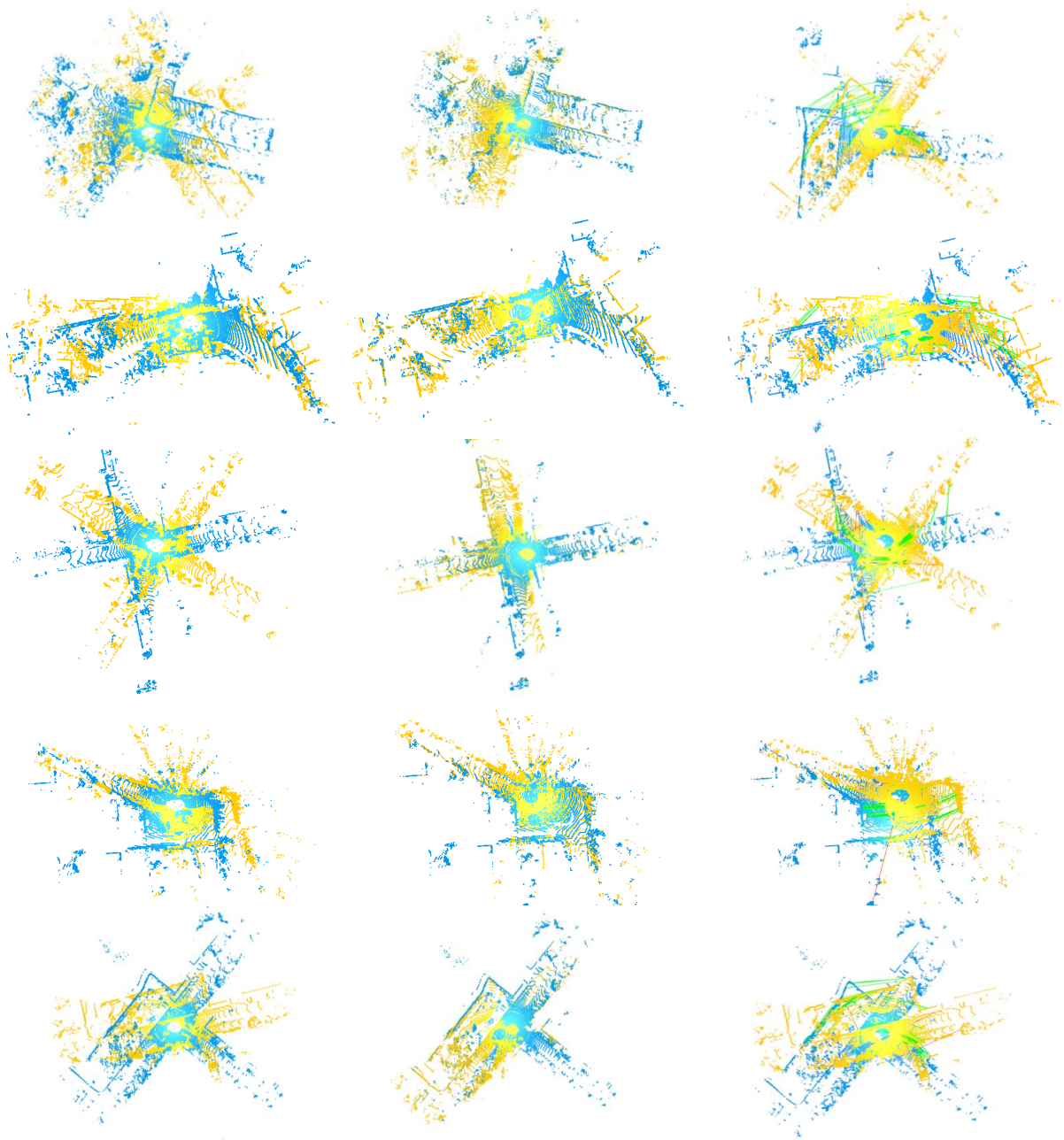

Figure 6. Qualitative results on KITTI outdoor scenes. (a) Input point cloud pairs. (b) Registration results after applying our predicted transformation. (c) Correspondence results showing identified inliers. Source and target point clouds are colored in cyan and yellow respectively.

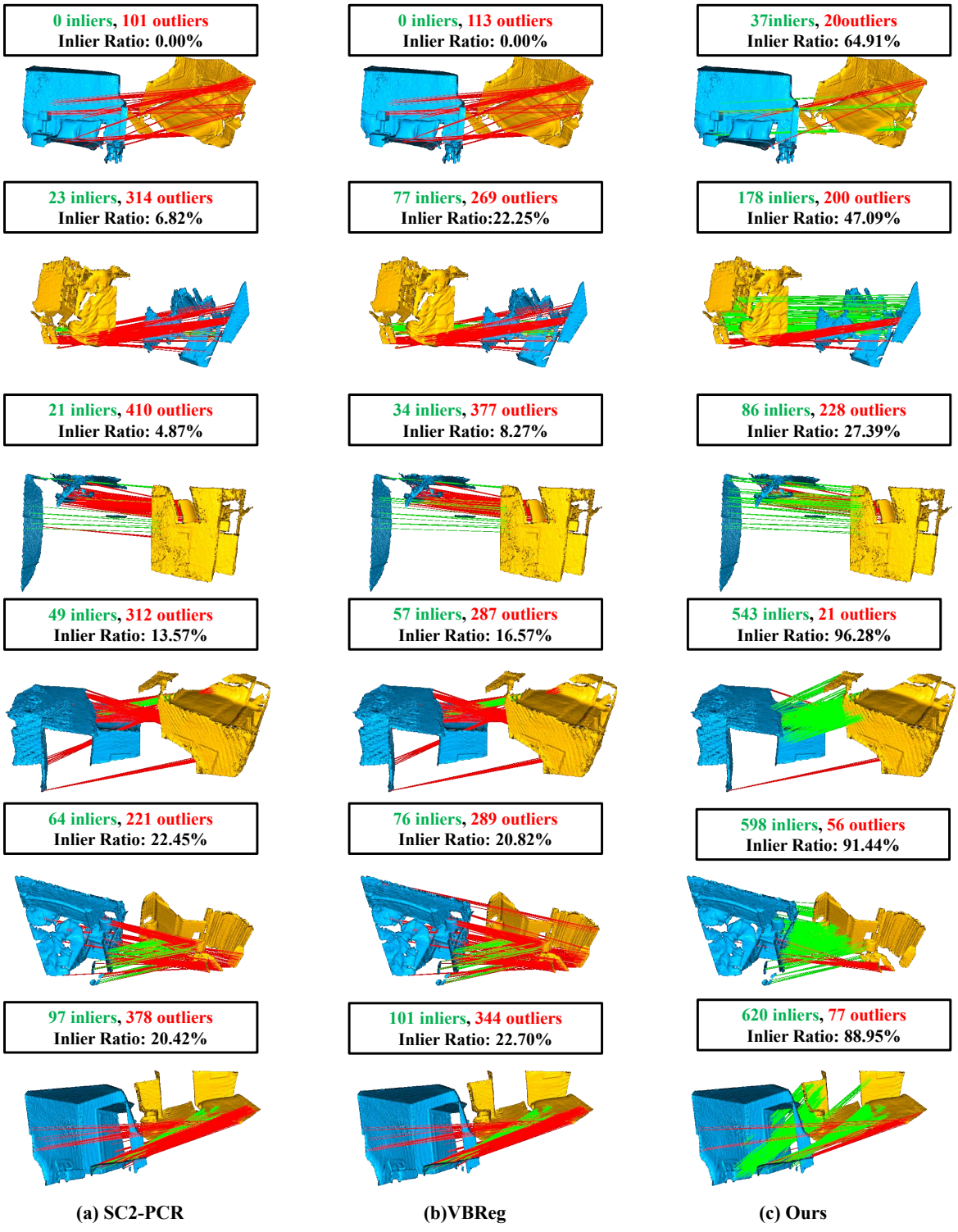

Figure 7. Qualitative comparison on challenging 3DLoMatch cases. Each row shows a representative test case with results from (a) SC^2-PCR, (b) VBReg, and (c) our method. Green lines denote correct inliers, red lines denote outliers. Text boxes show inlier/outlier counts and inlier ratios. Our method consistently achieves higher inlier ratios (average 58.26% vs. 11.19% for SC^2-PCR and 14.77% for VBReg) and more geometrically coherent correspondence patterns, especially in extreme low-overlap scenarios where baselines fail completely (row 1) or produce heavily contaminated results with $>$80% outliers (rows 2-6).