## A. Proof of Lemma 1

Lemma 1 serves as the basis of our analysis, indicating the fundamental incompetence for PCL, while hinting an *i.i.d.* solution towards density-invariance.

*Proof.* Since we need to investigate the effect of loss on a single feature $f_i^S$, we need to marginalize the effect of $f_j^T$. We start by selecting a specific first feature $\hat{f}_i^S$, regarding it as a constant, and take out all correspondences for the specific feature, which is $\hat{C} = \{(\hat{f}_i^S, f_j^T) \in C\}$. We focus on a part of the loss involving this specific $\hat{f}_i^S$, referred to as a function $\hat{L}_{pos}(\hat{f}_i^S)$, in equation 8.

$$\hat{L}_{pos}(\hat{f}_i^S) = \frac{1}{|\hat{C}|} \sum_{(\hat{f}_i^S, f_j^T) \in \hat{C}} max(||\hat{f}_i^S - f_j^T||_p - m, 0) \tag{8}$$

Next, we marginalize the effect of $f_j^T$ through sampling infinitely many $f_j^T$. Assuming $\hat{f}_i^S, f_j^T$ are i.i.d., then countless $f_j^T$ approximates the distribution $D$. We can write out the limitation of $\hat{L}_{pos}$ when $|\hat{C}| \to \infty$ as Equation 9.

$$\lim_{|\hat{C}| \to \infty} \hat{L}_{pos}(\hat{f}_i^S) = \mathbb{E}_{f_j^T \sim D} max(||\hat{f}_i^S, f_j^T||_p - m, 0) \tag{9}$$

Equation 9 is convex and has a single global minimum at $\hat{f}$ which solely depends on $D$ (cases where a minimal plateau exists is impossible in real setup). The effect of minimizing $L_{pos}$ converges in probability to all features $f \sim D$ heading towards the same location $\hat{f}$ in feature space.

$$\lim_{|\hat{C}| \to \infty} \hat{L}_{pos}(\hat{f}_i^S) = \mathbb{E}_{(\hat{f}_i^S, f_j^T) \in \hat{C}} max(||\hat{f}_i^S, f_j^T||_p - m, 0) \tag{10}$$

Otherwise, if $\hat{f}_i^S, f_j^T$ are non-*i.i.d.*, it is impossible to marginalize $f_j^T$, and the loss in Equation 10 is the expectation on a subset of correspondences $\hat{C}$ that is correlated with $\hat{f}_i^S$. All likely features have different loss formulation with different global minimums. This means that different features will converge towards different locations.

Note that the loss we investigate is a partial representation of the complete loss function, as negative losses are not considered. However, the result is highly likely true even with negative loss added. That is because positive loss controls the sub-structure inside a specific positive cluster, while negative loss controls the large-scale relative structure between different positive clusters, and the negative loss should not disturb positive structures too much when the feature representation stabilizes after the first few epochs.

## B. Detailed Experiment Setup

### B.1. Dataset Preparation

Two kinds of datasets are used in this paper, *i.e.*, pairwise contrastive learning (PCL) datasets and group-wise contrastive learning (GCL) datasets. The PCL datasets contain point cloud pairs that are sampled with a random distance interval $b$ denoting the distance between two LiDARs. The distance $b$ is randomly picked for every point cloud pair, and we refer to a sub-divided dataset where $b_1 \leq b \leq b_2$ as *[$b_1$,$b_2$]*. Both during training and testing, we always reset the random seed to 0 before finding the required point clouds to produce the exact same point cloud pairs for repeatable results. To create the GCL datasets, we sample central point clouds $C$ at a fixed interval of 11 frames, then randomly sample neighboring point clouds around each central point cloud according to the process described in Section 3.3. The GCL datasets are never used during testing.

Following Huang *et al*. [20], we define *overlap* $O$ between a pair of point clouds $S \in \mathbb{R}^{N \times 3}, T \in \mathbb{R}^{M \times 3}$ as a subset of $S$ according to Equation 11.

$$O = \left\{ p_S^i \in S \;\middle|\; \min_{P_T^j \in T} \|p_S^i - p_T^j\|_2 \leq \delta \right\} \tag{11}$$

The overlap denotes the part of $S$ where at least a corresponding point in $T$ could be found through nearest-neighbor search of radius $\delta = 0.45m$. $S$ and $T$ are downsampled using a voxel size of 0.3m before the search. Overlap ratio is then defined as $\frac{|O|}{|S|}$. All point cloud pairs with $\leq 30\%$ overlap ratio in *[5,20], [20,30], [30,40], [40,50]* datasets are collected on *KITTI* and *nuScenes*, referred to as *LoKITTI* and *LoNuScenes*, respectively. They represent the hardest cases for the distant point cloud registration task.

We follow previous literature [5] to divide OdometryKITTI with sequences 0-5 for training, 6-7 for validation, and 8-10 for testing. NuScenes is divided sequentially with the first 700 sequences for training, the next 150 sequences for validation and the last 150 sequences for testing.

### B.2. Metrics

Both traditional and new metrics are used during evaluation. Following previous work [16, 20, 5, 9], we report 3 metrics including Registration Recall (RR) defined as percentage of pairs successfully registered, Relative Rotation Error (RRE) defined as the geodesic distance between estimated rotation and ground-truth rotation, and Relative Translation Error (RTE) defined as the euclidean distance between estimated translation and the ground-truth translation. We forge a new metric as the average of RR on $[5, 10], [10, 20], [20, 30], [30, 40], [40, 50]$ datasets, referred

| | Dataloader | Inference | RANSAC | Total |
|---|---|---|---|---|
| FCGF | 6.2 | 45.4 | 576.1 | 627.7 |
| GCL+Conv (ours) | 5.4 | 44.2 | 523.0 | 572.6 |
| Predator | 635.1 | 78.7 | 66.3 | 780.1 |
| GCL+KPConv (ours) | 637.5 | 64.4 | 76.4 | 778.3 |

Table 6: **Inference time (ms) analysis on *LoKITTI*.** GCL is always more lightweight than their existing counterparts with the same backbone (FCGF: Conv; Predator: KPConv) in terms of inference time.

to as mean Registration Recall (mRR), which measures the overall registration performance.

### B.3. Network Structure

We adopt the popular Res-UNet network structure [9], and implement it on both sparse voxel convolution [8] and KPConv [41], referred to as GCL+Conv and GCL+KPConv, respectively. As depicted in Figure 10, both GCL+Conv and GCL+KPConv adopt three layers of skip connections with a roughly symmetric encoder-decoder design. Features are all normalized onto a unit sphere after the final layer.

### B.4. Loss Configuration

There are several parameters that need specifying for network convergence. The distance margins are set to $m_1 = 0.1, m_2 = 0.1, m_3 = 0.2, m_4 = 1.4$. The loss terms are reweighed differently on two datasets, where we set $\lambda_1 = \lambda_2 = \lambda_3 = 1$ on KITTI and $\lambda_1 = \lambda 2 = 0.7, \lambda_3 = 1$ on nuScenes.

## C. Additional Experiments

**Inference time.** We list the inference time breakdown for FCGF, Predator, GCL+Conv and GCL+KPConv in Table 6. The inference time of GCL is always lower than counterparts with the same backbone. While GCL+KPConv performs faster registration, it requires extended data loading time due to underlying KPConv architecture conducting repeated nearest neighbor calculation. In contrast, GCL+Conv runs faster during data loading and inference, and the extended RANSAC registration time can be reduced given recent progress on fast registration pipelines [4]. The focus of GCL is to propose a contrastive learning based training method which can be plugged into any existing registration pipelines that incorporate feature matching in it [10, 15, 32, 1, 50, 9, 5, 20, 51], and GCL is the general solution to the distant registration problem on all these methods since they are all based on either Voxel Convolution [8] or KPConv [41]. We conclude that GCL is a universal lightweight feature extraction method.

| Loss | *LoKITTI* | | | *KITTI [10,10]* | | |
|---|---|---|---|---|---|---|
| | RR | RTE | RRE | RR | RTE | RRE |
| C | 53.8 | 32.5 | 1.41 | 99.0 | 7.8 | 0.27 |
| F | 18.3 | 38.9 | 1.92 | 98.8 | 7.6 | **0.25** |
| PP | 53.8 | **27.2** | **1.28** | **99.2** | 7.6 | 0.26 |
| PV | 45.0 | 29.1 | 1.39 | 98.6 | 7.6 | **0.25** |
| BF+PP | 45.7 | 31.1 | 1.40 | 98.6 | 7.6 | 0.26 |
| F+PV | 50.5 | 28.4 | 1.30 | **99.2** | **7.5** | **0.25** |
| **F+PP** | **55.4** | 27.8 | **1.28** | **99.2** | 7.9 | 0.26 |

Table 7: **Ablation of loss designs** for GCL+KPConv on *KITTI [10,10]* and *LoKITTI*, measured by RR (%), RTE (cm), and RRE (°). F+PP is selected according to performance on *LoKITTI*. The gray column is the main metric.

| Dataset | mRR | *[5,10]* | *[10,20]* | *[20,30]* | *[30,40]* | *[40,50]* |
|---|---|---|---|---|---|---|
| FCGF [9] | 77.4 | 98.4 | 95.3 | 86.8 | 69.7 | 36.9 |
| Predator [20] | 87.9 | **100.0** | 98.6 | **97.1** | 80.6 | 63.1 |
| SpinNet [1] | 39.1 | 99.1 | 82.5 | 13.7 | 0.0 | 0.0 |
| D3Feat [5] | 66.4 | 99.8 | 98.2 | 90.7 | 38.6 | 4.5 |
| CoFiNet [52] | 82.1 | 99.9 | **99.1** | 94.1 | 78.6 | 38.7 |
| GeoTransformer [35] | 42.2 | **100.0** | 93.9 | 16.6 | 0.7 | 0.0 |
| GCL+KPConv (ours) | 89.6 | **100.0** | 98.2 | 93.2 | 88.3 | 68.5 |
| GCL+Conv (ours) | **93.5** | 99.0 | 98.8 | 96.1 | **91.7** | **82.0** |

Table 8: **Comparison of RR (%) between SOTA methods and GCL on five *KITTI* $[b_1, b_2]$ datasets**, with increasing LiDAR distance and registration difficulty. Registration metrics are loosened to 5°, 2m compared to Table 1. The mean RR is displayed in the first column.

| Dataset | mRR | *[5,10]* | *[10,20]* | *[20,30]* | *[30,40]* | *[40,50]* |
|---|---|---|---|---|---|---|
| FCGF [9] | 39.5 | 87.9 | 63.9 | 23.6 | 11.8 | 10.2 |
| Predator [20] | 51.0 | 99.7 | 72.2 | 52.8 | 16.2 | 14.3 |
| GCL-Conv (ours) | 85.5 | 99.3 | 97.7 | 91.8 | 77.8 | 60.7 |
| GCL-KPConv (ours) | **90.3** | **99.9** | **98.5** | **96.1** | **85.4** | **71.6** |

Table 9: **Comparison of RR (%) between SOTA methods and GCL on five *nuScenes* $[b_1, b_2]$ datasets**, with increasing LiDAR distance and registration difficulty. Registration metrics are loosened to 5°, 2m compared to Table 2. The mean RR is displayed in the first column.

**Loss ablation with GCL+KPConv.** We ablate various loss components for GCL+KPConv and display the registration performance of on both *KITTI [10,10]* and *LoKITTI* in Table 7. Similar to results with GCL+Conv, Finest Loss in combination with a positive loss performs the best among all methods, as F+PP achieves both the best RR of 55.4% on *LoKITTI* and 99.2% *KITTI [10,10]*. All methods perform roughly the same on the close point cloud dataset *KITTI [10,10]*. With the KPConv backbone, however, Finest loss alone does not lead to a decent performance on LoKITTI as it does with the voxel convolution backbone. We select F+PP as the optimal configuration for GCL+KPConv during all other experiments.

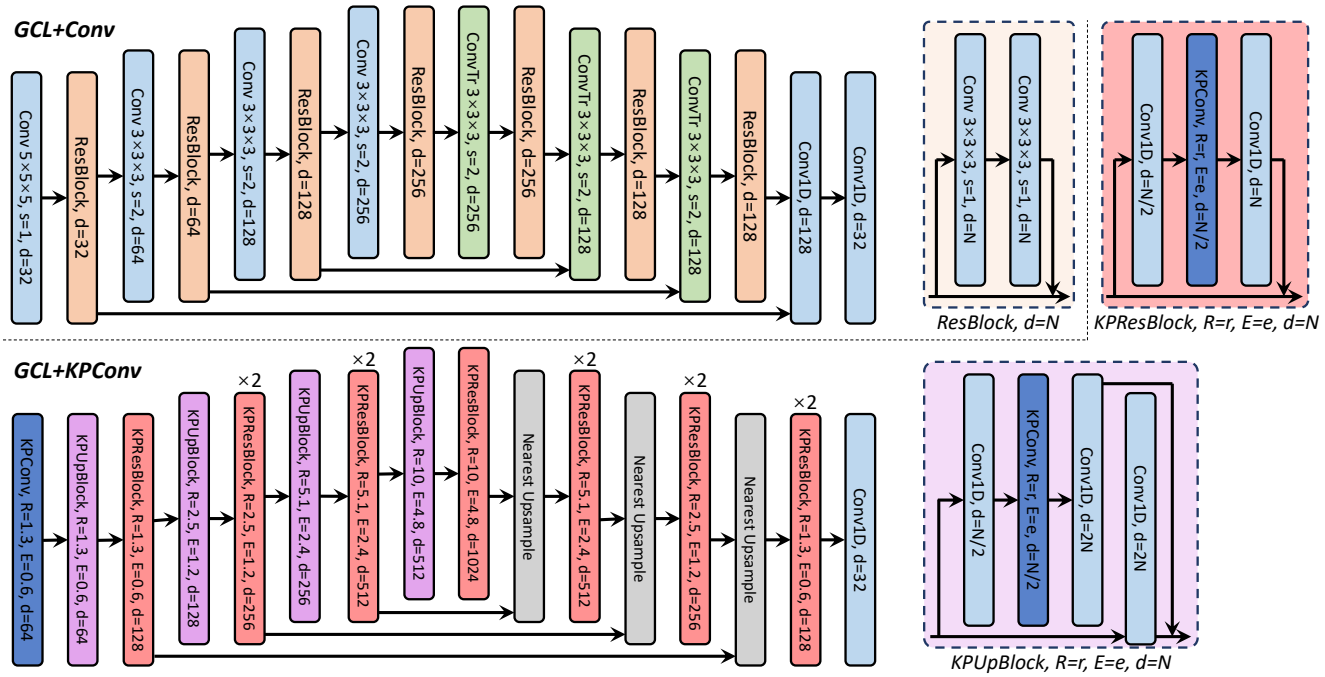


Figure 10: **Network structures for GCL.** Batch Normalization and ReLU activation are used after all Conv blocks except for the last layer, while batch normalization and leaky ReLU are used in KPConv blocks with a 0.1 slope. Voxel Convolution is parameterized by the kernel size, stride $s$, and output dimension $d$. The kernel size and stride are both omitted for Conv1D. Non-deformable KPConv [41] is parameterized by kernel point offset radius $R$, kernel point influence extent $E$, and feature dimension $d$.

**Performance comparison under loose registration criterion.** We additionally provide the comparison between GCL and SOTA methods on both *KITTI* and *nuScenes* under a loose registration criterion of $RTE \leq 2m, RRE \leq 5°$, where the registration recalls are generally elevated due to the loosen criterion. The mean RR is shown in the first column. As listed in Table 8, GCL+Conv and GCL+KPConv achieve the highest overall performance on *KITTI* with 89.6% (+1.7%) and 93.5% (+4.6%) mRR over Predator [20], respectively. Furthermore, GCL methods receive greater improvements on distant scenarios including *[30,40]* and *[40,50]* on *KITTI*. On the other hand, GCL methods beat SOTA methods by a larger margin on *nuScenes* than on *KITTI*, achieving 85.5% (+34.5%) and 90.3% (+39.3%) mRR for GCL+Conv and GCL+KPConv compared to Predator [20], respectively on *nuScenes* according to Table 9. We mark that GCL methods beat SOTAs on every sub-divided dataset on *nuScenes*, and that GCL+KPConv always performs the best. We conclude that, under a loose registration criterion, GCL still achieves giant improvements comparable to the scenario under a stricter criterion, setting a new SOTA for the distant point cloud registration problem.

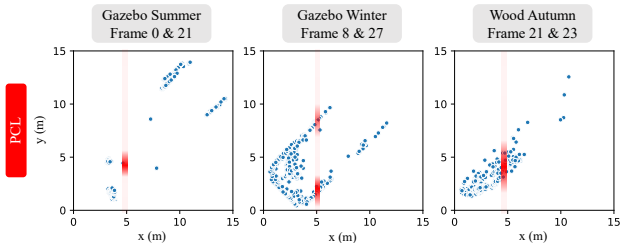


Figure 11: **Distribution of correspondences in ETH dataset** with PCL and $batch\_size = 1$, where the $x$ and $y$ coordinates of a dot denotes the distance from a correspondence to both LiDARs. The red stripes denote the conditional distribution of $y$ on a fixed $x$. Density correlation still exists in ETH, where linear structures are less dominant.

**Generalization to ETH.** We demonstrate the generalization results from KITTI [14] to ETH [33] in Table 10, by shrinking the voxel sizes from 0.3m to 0.05m during testing without any finetuning. ETH is an outdoor dataset featuring a majority of vegetation over linear structures. However, density correlation still exists in ETH, as depicted in Figure 11, which confirms the wide applicability of our analysis in Section 3.2. It can be seen that GCL effectively improves the generalization capability of two baseline back-

| | Gazebo | | Wood | | Avg. |
|---|---|---|---|---|---|
| | Summer | Winter | Autumn | Summer | |
| Predator | 21.2 | 20.8 | 23.5 | 30.4 | 24.0 |
| FCGF | 40.2 | 26.0 | 54.8 | 67.2 | 47.0 |
| GCL+KPConv (ours) | 46.2 | 28.4 | 56.5 | 72.0 | 50.8 |
| GCL+Conv (ours) | **46.7** | **30.8** | **61.7** | **73.6** | **53.2** |

Table 10: **Generalization test from KITTI to ETH**, by shrinking voxel size from 0.3m to 0.05m during testing. The FMR scores at $\tau_1$ = 10cm, $\tau_2$ = 5% are compared.

bones, where GCL+Conv achieves the best overall FMR of 53.2% (+6.2%). We conclude that GCL can generalize to other scenarios other than autonomous driving.

## D. Discussion and Limitation

**More explanation on non-*i.i.d.* PCL positives.** A pair of close-range point clouds also have non-*i.i.d.* positives, as their positives have roughly the same density, *i.e.*, their densities are positively correlated. This may sound weird, as close-range LiDAR point cloud registration has already been well-solved [20, 52]. Actually, non-*i.i.d.* positives will not hinder close-range registration problems because the problem is so simple that even a density-variant feature extractor will solve the problem nicely. Now consider a hand-crafted density-variant feature that upon the input coordinate $(x, y, z)$, outputs the vector length of the coordinate $\sqrt{x^2 + y^2 + z^2}$. Intuitively, this density-variant feature combined with RANSAC will likely produce a decent guess for two concentric (*i.e.*, extremely close) point clouds. However, this special solution will not work for distant scenarios with severe density mismatch, which means that a more powerful solution like GCL is needed to solve the distant point cloud registration problem.

**Training time.** As listed in Table 4, GCL has a linearly growing training time consumption *w.r.t.* $\phi$. This is mainly caused by increased data loading time where repeated nearest neighbor searches are carried out from the central point cloud to all neighborhood point clouds. However, the heavy time consumption is a necessary cost for building the positive groups. Luckily, only training time is affected for GCL and the testing time remain unchanged when registering two point clouds.

**Information exchange.** Information exchange serves as a key source of improvement for SOTA registration methods [20, 52, 44, 25, 35]. It is carried out between a pair of point clouds, which calls for a non-trivial extension of GCL that contains $\phi + 1$ point clouds. Note that features after the exchange will vary according to different companion point clouds. Consequently, a naive traversal of $C^2_{\phi+1}$ pairs for GCL will not only suffer from $O(\phi^2)$ complexity but also have to deal with $\phi$ different features for a single point. We hope to extend the information exchange module (mainly composed of cross-attention) to a group-wise version in future work.