Self-supervised Geometric Perception

Heng Yang[*] MIT LIDS
Wei Dong[*] CMU RI
Luca Carlone MIT LIDS
Vladlen Koltun Intel Labs

Abstract

We present self-supervised geometric perception *(*SGP*), the first general framework to learn a feature descriptor for correspondence matching without any ground-truth geometric model labels (*e.g.*, camera poses, rigid transformations). Our first contribution is to formulate geometric perception as an optimization problem that* jointly *optimizes the feature descriptor and the geometric models given a large corpus of visual measurements (*e.g.*, images, point clouds). Under this optimization formulation, we show that two important streams of research in vision, namely* robust model fitting *and* deep feature learning*, correspond to optimizing one block of the unknown variables while fixing the other block. This analysis naturally leads to our second contribution – the* SGP *algorithm that performs* alternating minimization *to solve the joint optimization.* SGP *iteratively executes two meta-algorithms: a* teacher *that performs robust model fitting given learned features to generate* geometric pseudo-labels*, and a* student *that performs deep feature learning under noisy supervision of the pseudo-labels. As a third contribution, we apply* SGP *to two perception problems on large-scale real datasets, namely relative camera pose estimation on* MegaDepth *and point cloud registration on* 3DMatch*. We demonstrate that* SGP *achieves state-of-the-art performance that is on-par or superior to the supervised oracles trained using ground-truth labels.*[1]

1. Introduction

Geometric perception is the task of estimating geometric models (*e.g.*, camera poses, rigid transformations, and 3D structures) from visual measurements (*e.g.*, images or point clouds). It is a fundamental class of problems in computer vision that has extensive applications in object detection and pose estimation [82, 91], motion estimation and 3D reconstruction [18, 26], simultaneous localization and mapping (SLAM) [13], structure from motion (SfM) [66], and virtual and augmented reality [46], to name a few.

Modern geometric perception typically consists of a *front-end* that detects, represents, and associates (sparse or dense) keypoints to establish *putative correspondences*, and a *back-end* that performs estimation of the geometric models while being robust to *outliers* (*i.e.*, incorrect correspondences). Traditionally, hand-crafted keypoint detectors and feature descriptors, such as SIFT [55] and FPFH [64], have been used for feature matching in 2D images and 3D point clouds. Despite being general and efficient to compute, hand-crafted features typically lead to an overwhelming number of outliers so that robust estimation algorithms struggle to return accurate estimates of the geometric models. For example, it is not uncommon to have over 95% of the correspondences estimated from FPFH be outliers in point cloud registration [61, 85]. As a result, learning feature descriptors from data, particularly using deep neural networks, has become increasingly popular. Learned feature descriptors have been shown to consistently and significantly outperform their hand-crafted counterparts across applications such as relative camera pose estimation [74, 65], 3D point cloud registration [22, 34], and object detection and pose estimation [62, 91, 69, 77].

However, existing feature learning approaches have several major shortcomings. First, a large number of *ground-truth* geometric model labels are required for training. For example, ground-truth relative camera poses are needed for training image keypoint descriptors [74, 57, 28], pairwise rigid transformations are required for training point cloud descriptors [22, 34, 79, 90, 75], and object poses are used to train image keypoint predictors [62, 91]. Second, although obtaining ground-truth geometric labels is trivial in some controlled settings such as robotic manipulation [31], in general the labels come from full 3D reconstruction pipelines (*e.g.*, COLMAP [66], Open3D [96]) that require delicate parameter tuning, partial human supervision, and extra sensory information such as IMU and GPS. As a result, the success of feature learning is limited to a handful of datasets with ground-truth annotations [92, 24, 53, 77, 11].

In this paper, we ask the key question: *Can we design a general framework for feature learning that requires no ground-truth geometric labels or sophisticated reconstruction pipelines?* Our answer is affirmative.

Contributions. We formulate geometric perception as an optimization problem that jointly searches for the best feature descriptor (for correspondence matching) and the best geometric models given a large corpus of visual measurements. This formulation incorporates robust model fitting and deep feature learning as two *subproblems*: (i) *robust estimation* only searches for the geometric models, while consuming putative correspondences established from a given feature descriptor; (ii) *feature learning* searches purely for the feature descriptor, while relying on

[*] Equal contribution. Work performed during internship at Intel Labs.
[1] Code available at https://github.com/theNded/SGP.

full supervision from the ground-truth geometric models. This generalization naturally endows geometric perception with an iterative algorithm that solves the joint optimization based on alternating minimization, which we name as *self-supervised geometric perception* (SGP). At each iteration, SGP alternates two meta-algorithms: a *teacher*, that generates geometric pseudo-labels using correspondences established from the learned features, and a *student*, that refines the learned features under the *noisy* supervision from the updated geometric models. SGP is initialized by generating geometric pseudo-labels using a bootstrap descriptor, *e.g.*, a descriptor that is hand-crafted or is trained using synthetic data. We apply SGP to solve two perception problems – relative camera pose estimation and 3D point cloud registration – and demonstrate that (i) SGP achieves on-par or superior performance compared to the supervised oracles; (ii) SGP sets the new state of the art on the MegaDepth [53] and 3DMatch [92] benchmarks.

2. Related Work

Deep feature learning. With the recent advance of deep learning, a plethora of deep features have been developed to replace classical hand-crafted feature descriptors such as SIFT [55] and FPFH [64] for correspondence matching, and boost the performance of geometric perception tasks. For 2D features, Choy *et al.* [21] develop Universal Correspondence Network (UCN) for visual correspondence estimation with metric contrastive learning. Tian *et al.* [71] introduce L2-Net to extract patch descriptors for keypoints. While these methods require direct correspondence supervision, Wang *et al.* [74] only use 2D-2D camera poses to supervise the learning of feature descriptors. The success of 2D feature learning extends to 3D. Khoury *et al.* [45] created Compact Geometric Features (CGF) by optimizing deep networks that map high-dimensional histograms into low-dimensional Euclidean spaces. Gojcic *et al.* [34] propose 3DSmoothNet for 3D keypoint descriptor generation with its network structure based on L2-Net. Choy *et al.* [22] developed fully convolutional geometric features (FCGF) based on sparse convolutions. Bai *et al.* [3] build D3Feat on kernel point convolution (KPConv) [70] and emphasize 3D keypoint detection. Since ground-truth 3D correspondences are non-trivial to obtain, nearest neighbor search using known 3D transformations is the standard supervision signal.

Robust estimation. Robust estimation ensures reliable geometric model estimation in the presence of outlier correspondences. *Consensus maximization* [17] and *M-estimation* [10] are the two popular formulations. Algorithms for solving both formulations can be divided into *fast heuristics*, *global solvers*, and *certifiable algorithms*. Fast heuristics, such as RANSAC [30, 4, 5] and GNC [95, 80, 1], are efficient but offer few performance guarantees. Global solvers, typically based on branch-and-bound [7, 61, 42, 86, 6, 50] or exhaustive search [29, 2, 16, 14], are globally optimal but often run in exponential time. Recently proposed certifiable algorithms [84, 85, 83, 81] combine fast heuristics with scalable optimality certification. Outlier-pruning methods [61, 82, 67] can significantly boost the robustness and efficiency of estimation algorithms. In this paper, we use robust estimation to *teach* feature learning.

Self-supervision. Self-supervision has been widely adopted in visual learning [44] to avoid massive human annotation. In such tasks, labels can be automatically generated by standard image operations [48, 93], classical vision algorithms [52, 43], or simulation [27, 63]. In real-world setups, geometric vision has actively employed self-supervision in optical flow [54], depth prediction [73, 33], visual odometry [97, 87], and registration [89, 22, 3]. These tasks rely on the supervision from camera poses or relative rigid transformations for image warping and correspondence generation, and thus benefit from well-established SLAM [58], 3D reconstruction [96], and SfM [66] pipelines. Although these systems are off-the-shelf, they usually require long execution times, delicate parameter tuning, and human supervision to safeguard their correctness. In this paper, we show how to perform self-supervised feature learning without 3D reconstruction pipelines and ground-truth geometric labels.

Self-training. Self-training [88, 38], as a special case of semi-supervised learning, has gained popularity in visual learning due to its potential to adapt to large-scale unlabeled data. Self-training first trains a model on a labeled dataset, then applies it on a larger unlabeled dataset to obtain *pseudo-labels* [49] for further training. Although pseudo-labels can be noisy, recent studies have shown that SOTA performance can be achieved on image classification [78, 98], and initial theoretical analyses have been proposed [76]. Our work uses robust estimation to generate pseudo-labels without initial supervised training, the first work to showcase the effectiveness of pseudo-labels in training feature descriptors for geometric perception.

3. The SGP Formulation

In this section, we first formulate geometric perception as a problem that *jointly* optimizes a correspondence matching function (*i.e.*, learning a descriptor) and the geometric models given a corpus of visual data (Section 3.1). Then we show that two of the most important research lines in computer vision, namely *robust estimation* and *feature learning*, correspond to fixing one part of the joint problem while optimizing the other part (Sections 3.2 and 3.3).

3.1. Joint Feature Learning and Model Estimation

We focus on geometric perception with pairwise correspondences between visual measurements.

Problem 1 (Geometric Perception)**.** *Consider a corpus of M* pairwise *visual measurements $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^M$, such as images or point clouds, and assume $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are related through a* geometric model *with unknown parameters $\boldsymbol{x}_i \in \mathcal{X}$, where $\mathcal{X}$ is the domain of the geometric models such as 3D poses. Suppose there is a preprocessing module ϕ that can extract a sparse or dense set of keypoint locations for each measurement,* i.e.,

$$\boldsymbol{p}_i^a = \phi\left(\boldsymbol{a}_i\right) \in \mathbb{R}^{d_a \times N_{a_i}}, \quad \boldsymbol{p}_i^b = \phi\left(\boldsymbol{b}_i\right) \in \mathbb{R}^{d_b \times N_{b_i}}, \tag{1}$$

*for all $i = 1, \dots, M$, where d_a, d_b are the dimensions of the keypoint locations (*e.g.*, 2 for images keypoints and 3 for point cloud keypoints), and N_{a_i}, N_{b_i} are the number of keypoints in $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ (w.l.o.g., assume $N_{a_i} \le N_{b_i}$), then the problem of geometric perception seeks to jointly learn a correspondence function $\mathcal{C}$ and estimate the unknown geometric models $\boldsymbol{x}_i$ by solving the following optimization:*

$$\min_{\mathcal{C}, \{\boldsymbol{x}_i\}_{i=1}^M \in \mathcal{X}^M} \quad \sum_{i=1}^{M} \sum_{k=1}^{N_{a_i}} \rho\left(r\left(\boldsymbol{x}_i, \boldsymbol{p}_{i,k}^a, \boldsymbol{q}_{i,k}^b\right)\right) \tag{2}$$

$$s.t. \qquad \boldsymbol{q}_{i,k}^b = \mathcal{C}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i, \boldsymbol{p}_i^b, \boldsymbol{b}_i), \tag{3}$$

where $\boldsymbol{p}_{i,k}^a \in \mathbb{R}^{d_a}$ denotes the location of the k-th keypoint in $\boldsymbol{a}_i$, $\boldsymbol{q}_{i,k}^b \in \mathbb{R}^{d_b}$ denotes the location of the corresponding *keypoint in $\boldsymbol{b}_i$, $r\left(\cdot\right)$ is the residual function that quantifies the mismatch between the two keypoints $\boldsymbol{p}_{i,k}^a$ and $\boldsymbol{q}_{i,k}^b$ under the geometric model $\boldsymbol{x}_i$, $\rho\left(\cdot\right)$ is a robust cost function that penalizes the residuals, and $\mathcal{C}\left(\cdot\right)$ is a function that takes each keypoint in $\boldsymbol{a}_i$ as input and predicts the corresponding keypoint in $\boldsymbol{b}_i$, by learning features from the visual data.*

To the best of our knowledge, Problem 1 is the first formulation that considers *joint* feature learning and model estimation in geometric perception. The correspondence function $\mathcal{C}$ typically contains a *learnable* feature descriptor (*e.g.*, parametrized by a deep neural network) and a matching function (*e.g.*, soft or hard nearest neighbor search) that generates correspondences using the learned descriptor. We now give two examples of Problem 1.

Example 1 (Relative Pose Estimation)**.** *Consider a corpus of image pairs $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^M$ with known camera intrinsics, where $\boldsymbol{a}_i, \boldsymbol{b}_i$ are RGB images, let $\phi(\cdot)$ be a keypoint detector,* e.g., SIFT *[55],* SuperPoint *[25], or a dense random pixel location sampler [74], such that $\boldsymbol{p}_i^a = \phi(\boldsymbol{a}_i) \in \mathbb{R}^{2 \times N_{a_i}}$ and $\boldsymbol{p}_i^b = \phi(\boldsymbol{b}_i) \in \mathbb{R}^{2 \times N_{b_i}}$ are two sets of 2D keypoint locations. Relative pose estimation seeks to jointly learn a correspondence prediction function $\mathcal{C}$ and estimate the relative poses $\boldsymbol{x}_i = (\boldsymbol{R}_i, \boldsymbol{t}_i) \in \mathrm{SO}(3) \times \mathbb{S}^2$ between images.*[2] *In particular, following [74], let $\mathcal{C}$ be a composition of a deep feature descriptor $\mathcal{F}(\cdot)$, a* softmax *function [36], and a weighted average:*

$$\boldsymbol{q}_{i,k}^b = \sum_{j=1}^{N_{b_i}} \boldsymbol{p}_{i,j}^b \frac{\exp\left(\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i)^{\mathsf{T}} \mathcal{F}(\boldsymbol{p}_{i,j}^b, \boldsymbol{b}_i)\right)}{\sum_{j=1}^{N_{b_i}} \exp\left(\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i)^{\mathsf{T}} \mathcal{F}(\boldsymbol{p}_{i,j}^b, \boldsymbol{b}_i)\right)}, \tag{4}$$

where the descriptor $\mathcal{F}$ takes the image and the keypoint location as input and outputs a high-dimensional feature vector for each keypoint, i.e., *$\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i) \in \mathbb{R}^{d_{\mathcal{F}}}$, where $d_{\mathcal{F}}$ denotes the dimension of the descriptor, the* softmax *function computes the probability of $\boldsymbol{p}_{i,j}^b$ being a match to $\boldsymbol{p}_{i,k}^a$ according to their inner product in the descriptor space, and the weighted average function returns the keypoint location as a weighted sum of all keypoint locations discounted by their matching probabilities.*

Example 2 (Point Cloud Registration)**.** *Consider a corpus of point cloud pairs $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^M$, where $\boldsymbol{a}_i, \boldsymbol{b}_i$ are 3D point clouds, let $\phi(\cdot)$ be a 3D keypoint detector,* e.g., ISS3D *[94],* USIP *[51], or a dense uniform voxel downsampler [22], such that $\boldsymbol{p}_i^a = \phi(\boldsymbol{a}_i) \in \mathbb{R}^{3 \times N_{a_i}}$, and $\phi(\boldsymbol{b}_i) \in \mathbb{R}^{3 \times N_{b_i}}$ are two sets of 3D keypoints. Point cloud registration seeks to jointly learn a correspondence function $\mathcal{C}$ and estimate the rigid transformation $\boldsymbol{x}_i = (\boldsymbol{R}_i, \boldsymbol{t}_i) \in \mathrm{SO}(3) \times \mathbb{R}^3$ between point clouds. In particular, following [22, 34], let $\mathcal{C}$ be a composition of a deep feature descriptor $\mathcal{F}(\cdot)$ and nearest neighbor search:*

$$\boldsymbol{q}_{i,k}^b = \arg\min_{\boldsymbol{p}_{i,j}^b \in \boldsymbol{p}_i^b} \left\| \mathcal{F}(\boldsymbol{p}_{i,j}^b, \boldsymbol{b}_i) - \mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i) \right\|, \tag{5}$$

where the descriptor $\mathcal{F}$ takes the point cloud and the keypoint location as input and outputs a high-dimensional feature vector for each keypoint, i.e., *$\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i) \in \mathbb{R}^{d_{\mathcal{F}}}$, with $d_{\mathcal{F}}$ denoting the descriptor dimension, and condition (5) asks that the corresponding keypoint $\boldsymbol{q}_{i,k}^b$ is the keypoint among $\boldsymbol{p}_i^b$ that achieves the shortest distance to $\boldsymbol{p}_{i,k}^a$ in descriptor space.*[3]

Examples 1-2 represent two key problems in vision that concern pose estimation from 2D-2D and 3D-3D measurements, all of which involve the coupling of correspondence matching (*a.k.a.* data association) and geometric model estimation. Interestingly, although little is known about how to solve Problem 1 directly, significant efforts have been made to solve its two subproblems.

3.2. Robust Estimation

Problem 2 (Robust Estimation)**.** *In Problem 1, assuming the correspondence matching function $\mathcal{C}$ is known, robust*

[2] The translation $\boldsymbol{t} \in \mathbb{S}^2 \doteq \{\boldsymbol{t} \in \mathbb{R}^3 | \, \|\boldsymbol{t}\| = 1\}$ is up to scale.

[3] Alternatively, one can establish correspondences through *cross check* [95] or *ratio test* [55]. In addition to (5), cross check asks $\boldsymbol{p}_{i,k}^a$ is also the closest keypoint to $\boldsymbol{q}_{i,k}^b$ among $\boldsymbol{p}_i^a$, while ratio test asks the ratio $\|\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i) - \mathcal{F}(\boldsymbol{q}_{i,k}^b, \boldsymbol{b}_i)\| / \|\mathcal{F}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i) - \mathcal{F}(\boldsymbol{p}_{i,j}^b, \boldsymbol{b}_i)\|$ is below a predefined threshold $\zeta < 1$ for all $\boldsymbol{p}_{i,j}^b \neq \boldsymbol{q}_{i,k}^b$.

estimation seeks to estimate the unknown parameters of the geometric models given putative correspondences (corrupted by outliers*), by optimizing the following objective:*

$$\min_{\{\boldsymbol{x}_i\}_{i=1}^{M} \in \mathcal{X}^M} \quad \sum_{i=1}^{M} \sum_{k=1}^{N_{a_i}} \rho\left(r\left(\boldsymbol{x}_i, \boldsymbol{p}_{i,k}^a, \boldsymbol{q}_{i,k}^b\right)\right). \tag{6}$$

Problem 2 shows that robust estimation is a subproblem of Problem 1 with a known and fixed correspondence function. Despite the nonconvexity of problem (6) (*e.g.*, due to a nonconvex $\mathcal{X}$ or a nonconvex ρ), research in robust estimation has focused on improving the robustness [85, 61], efficiency [5] and theoretical guarantees [84] of estimation algorithms to mitigate the adversarial effects of outliers on the estimated geometric models.

3.3. Supervised Feature Learning

Problem 3 (Supervised Feature Learning)**.** *In Problem 1, assuming the parameters of the geometric models are known and denoting them as $\boldsymbol{x}_i^\circ, i = 1, \ldots, M$, feature learning seeks to find the best correspondence matching function $\mathcal{C}_{\boldsymbol{\theta}}$ by solving the following optimization problem:*

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{N_C}} \quad \sum_{i=1}^{M} \sum_{k=1}^{N_{a_i}} \rho(r(\boldsymbol{x}_i^\circ, \boldsymbol{p}_{i,k}^a, \boldsymbol{q}_{i,k}^b)) \tag{7}$$

$$s.t. \quad \boldsymbol{q}_{i,k}^b = \mathcal{C}_{\boldsymbol{\theta}}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i, \boldsymbol{p}_i^b, \boldsymbol{b}_i), \tag{8}$$

where the correspondence function is parametrized by the weights $\boldsymbol{\theta} \in \mathbb{R}^{N_C}$ of a deep (descriptor) neural network and N_C is the number of weight parameters in the network.

At first glance, the optimization (7) is different from the loss functions designed in the supervised feature learning literature [22, 74, 91]. However, the next proposition states that, if we take $\rho(\cdot)$ to be the truncated least squares (TLS) cost function, then common loss functions can be designed using the *Augmented Lagrangian Method* (ALM) [8].

Proposition 1 (Feature Learning as ALM)**.** *Let $\rho(r) = \min\left\{r^2, \bar{c}^2\right\}$ be the* TLS *cost function [84], where $\bar{c} > 0$ sets the maximum allowed inlier residual, supervised feature learning [74, 22] in Examples 1-2 can solve the optimization* (7)*. In particular, the loss functions in [74, 22] can be interpreted as the Augmented Lagrangian of problem* (7)*.*

Proof. See the Supplementary Material. □

Proposition 1 states that, just as robust estimation algorithms optimize geometric models given a fixed correspondence matching function, supervised feature learning methods optimize the feature descriptor given known geometric models. In the next section, we show that this framework naturally allows us to solve Problem 1 by alternating the execution of robust estimation and feature learning.

4. The SGP Algorithm

We first give an overview of the SGP algorithm (Section 4.1), then discuss its applications (Section 4.2).

4.1. Overview

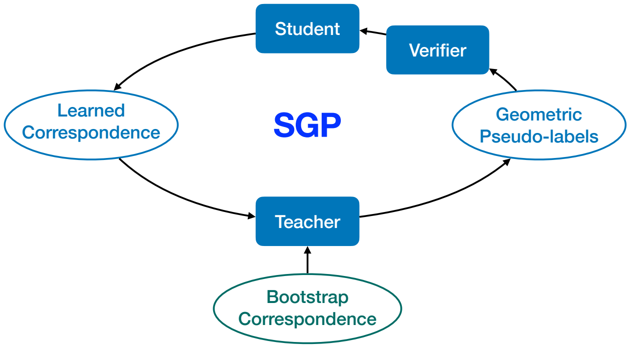


Figure 1. Algorithmic overview of SGP.

An overview of SGP is shown in Fig. 1, and details of SGP are summarized in Algorithm 1. SGP does not have access to the ground-truth geometric models and internally creates *geometric pseudo-labels*. SGP contains three key components: a *teacher*, a *student* and (optionally) a *verifier*.

Definition 1 (Teacher)**.** *An algorithm that estimates geometric pseudo-labels given a correspondence matcher.*

Definition 2 (Student)**.** *An algorithm that estimates the parameters of a correspondence matching function under the supervision of geometric models.*

Definition 3 (Verifier)**.** *An algorithm that verifies if a geometric model estimated by the teacher is correct.*

From the definitions above, one can see that a teacher is a solver for the robust estimation problem (6), while a student is a solver for the supervised feature learning problem (7). Because problems (6) and (7) are the two subproblems of the joint geometric perception problem (2), the SGP algorithm 1 alternates in executing the teacher and the student (*cf.* line 21-23), referred to as the *teacher-student loop*, to perform alternating minimization for the joint problem (2).

In particular, at the τ-th iteration of the teacher-student loop, the student initializes the network parameters at $\boldsymbol{\theta}$, and updates the parameters to $\boldsymbol{\theta}^{(\tau)}$, by minimizing problem (7) (using stochastic gradient descent) under the noisy "supervision" of the geometric pseudo-labels estimated from iteration $\tau - 1$ (line 21). The student either initializes $\boldsymbol{\theta}$ at random (line 17, referred to as retrain), or initializes $\boldsymbol{\theta}$ from the weights of the last iteration $\boldsymbol{\theta}^{(\tau-1)}$ (line 19, referred to as finetune). Then, using the correspondence function with updated parameters, denoted by $\mathcal{C}_{\boldsymbol{\theta}^{(\tau)}}$, the teacher solves robust estimation (6) to update the models (line 23).

Throughout the teach-student loop, neither the correspondence matcher nor the teacher are perfect, leading to

Algorithm 1: SGP

```
1  Input: A corpus of visual measurements:
       $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i=1}^M$; a preprocessing module: $\phi$; an initial
       correspondence matching method: $\mathcal{B}$; an
       architecture for a learned correspondence
       prediction function: $\mathcal{C}$, with initial weights: $\boldsymbol{\theta}^{(0)}$
       (default: randParam); Number of iterations: $T$;
       boolean: verifyLabel (default True); boolean: retrain
       (default False);
2  Output: final weights of $\mathcal{C}$: $\hat{\boldsymbol{\theta}}$; estimated geometric
       models: $\{\hat{\boldsymbol{x}}_i\}_{i=1}^M$;
3  % Compute keypoint locations
4  $\boldsymbol{p}_i^a = \phi(\boldsymbol{a}_i), \boldsymbol{p}_i^b = \phi(\boldsymbol{b}_i), \ \forall i \in [M]$;
5  % Bootstrap (Initialize pseudo-labels)
6  $\boldsymbol{x}_i^{(0)} = \mathsf{teach}(\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{p}_i^a, \boldsymbol{p}_i^b, \mathcal{B}), \ \ \forall i \in [M]$;
7  % Alternating minimization
8  for $\tau = 1 : T$ do
9      if verifyLabel = True then
10         % Verify correctness of labels
11         $\mathcal{S} = \mathsf{verify}(\{\boldsymbol{x}_i^{(\tau-1)}, \boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{p}_i^a, \boldsymbol{p}_i^b\}_{i=1}^M)$;
12     else
13         $\mathcal{S} = [M]$;
14     end
15     % Feature learning (problem (7))
16     if retrain = True then
17         $\boldsymbol{\theta} = \mathsf{randParam}$; % retrain
18     else
19         $\boldsymbol{\theta} = \boldsymbol{\theta}^{(\tau-1)}$; % finetune
20     end
21     $\boldsymbol{\theta}^{(\tau)} = \mathsf{learn}(\{\boldsymbol{x}_i^{(\tau-1)}, \boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{p}_i^a, \boldsymbol{p}_i^b\}_{i \in \mathcal{S}}, \boldsymbol{\theta})$;
22     % Robust estimation (problem (6))
23     $\boldsymbol{x}_i^{(\tau)} = \mathsf{teach}(\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{p}_i^a, \boldsymbol{p}_i^b, \mathcal{C}_{\boldsymbol{\theta}^{(\tau)}}), \ \ \forall i \in [M]$;
24 end
25 return: $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(T)}, \hat{\boldsymbol{x}}_i = \boldsymbol{x}_i^{(T)}, i = 1, \ldots, M$.
```

a significant fraction of the geometric pseudo-labels being incorrect, which can potentially bias the student. Therefore, SGP optionally uses a verifier to generate a verified set of pseudo-labels, denoted by $\mathcal{S}$, that are more likely to be correct (line 11). If the flag verifyLabel is False, then $\mathcal{S} = [M]$ is the full set of pseudo-labels (line 13). The verifier design is application dependent, as discussed in Section 4.2.

An *initialization* is required to start the iterative updates in alternating minimization. To do so, we initialize the geometric models by performing model estimation using a *bootstrap matcher* $\mathcal{B}$ (line 6). Based on the specific application, the bootstrap matcher can be designed from a hand-crafted feature descriptor that requires no learning, or a descriptor that is trained with a small amount of data, or a descriptor that is trained on synthetic datasets. On the other hand, since we typically do not have prior information about the weights of $\mathcal{C}$, $\boldsymbol{\theta}^{(0)}$ is initialized at random.

Remark 1 (Implementation Considerations)**.** *(i)* Convergence*: In the current* SGP *implementation, we execute the teacher-student loops for a fixed number of iterations T. However, one can stop* SGP *if the difference between $\boldsymbol{x}_i^{(\tau)}$ and $\boldsymbol{x}_i^{(\tau-1)}$, or between $\boldsymbol{\theta}^{(\tau)}$ and $\boldsymbol{\theta}^{(\tau-1)}$ is below some threshold. One can also choose the best $\mathcal{C}$ from* SGP *by using a validation dataset if available. (ii)* Speedup*: When running the teacher to generate pseudo-labels (line 23) at each iteration, one can skip the updates for some labels that are already "stable". For example, if a label $\boldsymbol{x}_i$ remains unchanged for consecutively 3 iterations, or the robust solver achieves high confidence about $\boldsymbol{x}_i$* (e.g., RANSAC *has inlier rate over* 80%*), then the teacher can skip the update for $\boldsymbol{x}_i$.*

4.2. Applications

We now discuss the application of SGP to Examples 1-2.

SGP **for Example 1**. The teacher performs robust relative pose estimation [39]. Therefore, a good candidate for a teacher is RANSAC [30] (with Nister's 5-point method [59]) and its variants, such as GCRANSAC [4] and MAGSAC [5]. The student performs descriptor learning using relative camera pose supervision. Recent work CAPS [74] is able to learn a descriptor under the supervision of fundamental matrices, which can be computed from relative pose and camera intrinsics [39]. Therefore, CAPS is the student network. The verifier can be designed based on the *inlier rate* estimated by RANSAC, *i.e*., the number of inlier matches divided by the total number of putative matches. Intuitively, the higher the inlier rate is, the more likely it is that RANSAC has found a correct solution. To initialize SGP, we use the hand-crafted SIFT descriptor (with ratio test) [55].

SGP **for Example 2**. The teacher performs robust registration. Many robust registration algorithms can serve as the teacher: RANSAC (with Horn's 3-point method [40]) and its variants, FGR [95], and TEASER++ [85]. As for the student, methods such as FCGF [22], 3DSmoothNet [34], and D3Feat [3] can learn point cloud descriptors under the supervision of rigid transformations. The verifier can be designed based on the *overlap ratio* computed from the estimated pose, *i.e*., the number of point pairs that are close to each other after transformation, divided by the total number of points in the point cloud. One can also use the certifier in TEASER++ [85]. To initialize SGP, we can use the hand-crafted FPFH descriptor (with cross check) [64].

Remark 2 (Novelty)**.** *Hand-crafted descriptors, robust estimation and feature learning are mature areas in computer vision. In this paper, instead of creating new techniques in each area, we show that combining existing techniques from each field in the* SGP *framework can tackle self-supervised geometric perception in full generality.*

Remark 3 (Generality)**.** *Although we only provide experimental results for relative pose estimation and point cloud registration, the joint optimization formulation in Problem 1 is general and the* SGP *algorithm 1 can be applied in any perception problem where a robust solver and a supervised feature learning method is available. For example, we also present the formulation for* object detection and pose estimation *[69, 62, 91, 15], and discuss the application of* SGP *in the Supplementary Material.*

5. Experiments

We first provide results demonstrating successful applications of SGP to relative pose estimation (Section 5.1) and point cloud registration (Section 5.2), then report ablation studies on point cloud registration where we vary the algorithmic settings of SGP (Section 5.3). *Detailed experimental data are tabulated in the Supplementary Material.*

5.1. Relative Pose Estimation

Setup. We first showcase SGP for Example 1 on the MegaDepth [53] benchmark containing a large collection of Internet images for the task of relative pose estimation. We adopted RANSAC10K (*i.e.*, RANSAC with maximum $10,000$ iterations) with 99.9% confidence and 0.001 inlier threshold as the teacher. We used the recently proposed CAPS [74] feature learning framework as the student.[4] To bootstrap SGP, we performed RANSAC10K with SIFT detector, SIFT descriptor, and 0.75 ratio test to initialize the geometric pseudo-labels (*i.e.*, relative poses).

To speed up the training of SGP, we sampled 10% of the original MegaDepth training set used in [74] uniformly at random, resulting in $78,836$ pairs of images *without* relative pose labels. To train CAPS, we modified the publicly available CAPS implementation[5], adopted a smaller batch size 5, and kept the Adam optimizer with initial learning rate 10^{-4}. We used finetune (*cf.* line 19) for the teacher-student loop, and in every iteration, we trained CAPS for $40,000$ steps. We trained SGP for a fixed number of $T = 10$ iterations.

In the teacher-student loop, we designed a verifier that prunes pseudo-labels according to the results of RANSAC10K – we only pass to the student pairs whose number of putative matches (either from SIFT with ratio test or CAPS with cross check) is above 100 and whose RANSAC estimated inlier rate is over 10%. Intuitively, pseudo-labels satisfying these two conditions are more likely to be correct.

We name the CAPS descriptor learned from SGP without ground-truth supervision as S-CAPS. We evaluated the performance of S-CAPS on (i) the MegaDepth test set, provided in [74], including $3,000$ image pairs equally divided into *easy*, *moderate*, and *hard* categories; (ii) the ScanNet [24] dataset to test cross-dataset generalization.

Results. Fig. 2 plots the dynamics of SGP on MegaDepth. PLSR stands for *Pseudo-Label Survival Rate* and is computed as $|\mathcal{S}|/M \times 100\%$, *i.e.*, the percentage of pseudo-labels that survived the verifier (*cf.* line 11). PLIR stands for *Pseudo-Label Inlier Rate* and denotes the percentage of correct labels in $\mathcal{S}$, a number that is not used by SGP but computed *a posteriori* using the ground-truth labels to show that SGP is robust to partially incorrect labels. Besides PLSR and PLIR, Fig. 2 plots the rotation recalls on both the training and the test sets (the translation recalls exhibit a similar trend and are shown in the Supplementary Material).[6] The BS (bootstrap) iteration plots the training and test recalls using SIFT. We make the following observations from Fig. 2: (i) PLSR gradually increases and approaches 90% w.r.t. iterations, indicating that the S-CAPS descriptor establishes dense correspondences with high inlier ratio, encouraged by the verifier; (ii) PLIR remains close to 90%, and is always higher than the recall, indicating that the verifier is effective in removing wrong labels; (iii) S-CAPS gradually improves itself on both the training and the test sets. (iv) While SIFT works better than S-CAPS on the training set, S-CAPS significantly outperforms SIFT on the test set.

Table 1 compares the performance of two versions of S-CAPS to other SOTA methods. S-CAPS^T is the S-CAPS descriptor at the last iteration, while S-CAPS* is the S-CAPS descriptor that performs best on the MegaDepth test set. We see that both versions of S-CAPS outperform the strong baseline using SIFT with ratio test and RANSAC10K, as well as the two SOTA results from the original CAPS [74] using both SIFT detector and SuperPoint detector [25].[7] Moreover, we report the performance of RANSAC10K plus the *supervised oracle*, CAPS$^\circ$, that is trained using full ground-truth supervision, on the test set. One can see that S-CAPS, trained using only 10% of the unlabeled training set, performs on par compared with the supervised oracle.

Fig. 3 provides qualitative examples of correspondence matching results on both MegaDepth and ScanNet. More examples are provided in the Supplementary Material.

5.2. Point Cloud Registration

Setup. To demonstrate SGP for Example 2, we conducted experiments on 3DMatch [92], a benchmark containing point clouds of real-world indoor scenes. We used RANSAC10K (with 7cm inlier threshold) plus ICP [9] as the teacher, FCGF [22] as the student, and FPFH [64] as the bootstrap descriptor to initialize transformation labels.

[4] We assumed known camera intrinsics so the fundamental matrix can be computed from the essential matrix to supervise CAPS.

[5] https://github.com/qianqianwang68/caps

[6] Recall is defined as the percentage of correctly estimated models divided by the total number of pairs. Following [74], we say a rotation or a translation is estimated correctly if it has angular error less than 10° w.r.t. to the groundtruth (note that translation is estimated up to scale).

[7] We suspect the RANSAC in [74] is not carefully tuned.

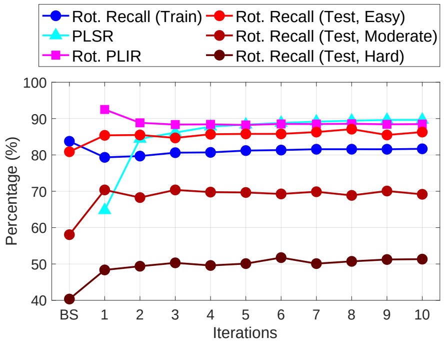


Figure 2. Dynamics of SGP on MegaDepth [53]. PLSR: *Pseudo-Label Survival Rate*. PLIR: *Pseudo-Label Inlier Rate*. BS: Boostrap.

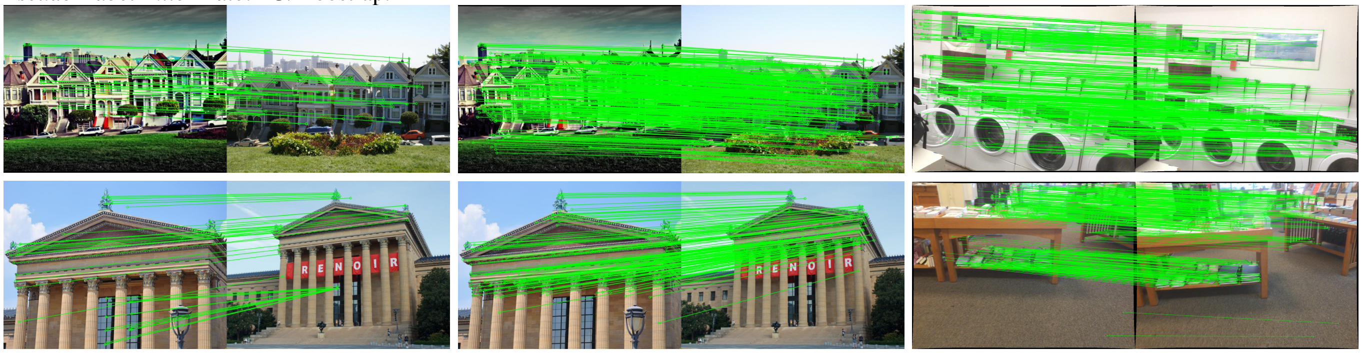
	Easy (%)		*Moderate* (%)		*Hard* (%)	
Methods	Rotation	Translation	Rotation	Translation	Rotation	Translation
SIFT+RANSAC10K [55][a]	80.9	48.8	58.1	43.5	40.4	34.0
SIFT+Wang-CAPS [74][b]	70.0	30.5	50.2	24.8	36.8	16.1
SuperPoint+Wang-CAPS [74][b]	72.9	30.5	53.5	27.9	38.1	19.2
SIFT+CAPS$^\circ$+RANSAC10K[c]	**87.1**	52.7	**72.5**	**53.8**	**52.7**	45.6
SIFT+S-CAPS^T+RANSAC10K	86.3	53.1	69.2	50.3	51.3	**47.1**
SIFT+S-CAPS*+RANSAC10K	**87.1**	**53.5**	70.4	53.3	51.8	**47.1**

Table 1. Rotation and translation recalls on the MegaDepth [53] test dataset using different methods. S-CAPS^T: last CAPS trained by SGP. S-CAPS*: best CAPS trained by SGP.

[a] SIFT and RANSAC implemented in OpenCV [12]. SIFT uses 0.75 ratio test. All RANSAC use 99.9% confidence.

[b] Recall statistics adapted from the original CAPS paper [74].

[c] Recall computed by using RANSAC10K with the pretrained CAPS$^\circ$ (*i.e.*, the supervised oracle).



(a) Success (top) and failure (bottom) by SIFT.

(b) Successes by S-CAPS^T.

(c) Cross-dataset generalization of S-CAPS^T.

Figure 3. Qualitative results showing the improved performance of (b) S-CAPS^T over the bootstrap descriptor (a) SIFT for relative pose estimation on MegaDepth [53], and (c) cross-dataset generalization of S-CAPS^T for relative pose estimation on the ScanNet dataset [24]. Green lines are inlier correspondences estimated by RANSAC10K. S-CAPS^T outputs reliable and dense matches. [Best viewed digitally]

SGP was trained on the training set provided by DGR [20] containing $9,856$ pairs of scans, *without* ground-truth transformation labels. Input point clouds were all voxelized with 5cm resolution before feature extraction (both FPFH and FCGF) and registration. To train FCGF, we followed the configuration of the original FCGF and used SGD with initial learning rate 0.1.[8] In the teacher-student loop, we used finetune, where we train FCGF for 100 epochs at iteration 1 and 50 epochs for the rest of the iterations. We designed a verifier based on estimated overlap ratio, *i.e.*, only pairs with estimated overlap ratio over η are passed to FCGF. We set $\eta = 30\%$ for the first two iterations and $\eta = 10\%$ for the rest. SGP is trained for $T = 10$ iterations.

We name the FCGF descriptor learned from SGP without ground-truth supervision as S-FCGF. We evaluated the performance of S-FCGF on (i) the 3DMatch test set including $1,623$ pairs; and (ii) the unseen Stanford RGBD dataset [18] for multi-way registration [96].

Results. Fig. 4 plots the dynamics of SGP on 3DMatch. We observe that: (i) PLSR increases and approaches 96%, indicating that more pairs enter the noisy student training; (ii) PLIR remains close to 93%, and is always higher than the recall, showing the effect of the verifier; (iii) S-FCGF gradually improves itself on both training and test sets.

Table 2 compares the performance of S-FCGF^{T}and S-FCGF* to other SOTA methods.[9] We see that S-FCGF* outperforms the baseline FPFH, FCGF [22], and the recently proposed DGR (even with RANSAC80K) [20]. We also provide results using RANSAC10K plus the *supervised oracle*, FCGF$^\circ$, that is trained using full ground-truth supervision. S-FCGF* outperforms the supervised oracle, while S-FCGF^T achieves similar performance.

Fig. 5 shows qualitative results using S-FCGF for pairwise registration on 3DMatch and for multi-way registration on Stanford RGBD. More qualitative results are shown in the Supplementary Material.

5.3. Ablation Study

We first study the effect of using retrain vs finetune in SGP for point cloud registration. We used the same setup as in Section 5.2, except that we changed from finetune to retrain, where in each iteration, we initialized the weights of FCGF at random and trained it for 100 epochs. We also set the verifier overlap ratio $\eta = 10\%$ for all iterations. Fig. 6(a) plots the corresponding dynamics, which overall looks similar to

[8] https://github.com/chrischoy/FCGF

[9] Following [20], we say a registration is successful if rotation error is below 15° and translation error is below 30cm.

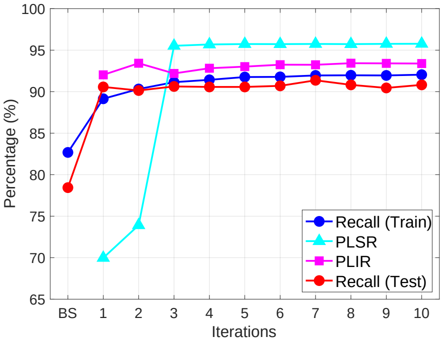
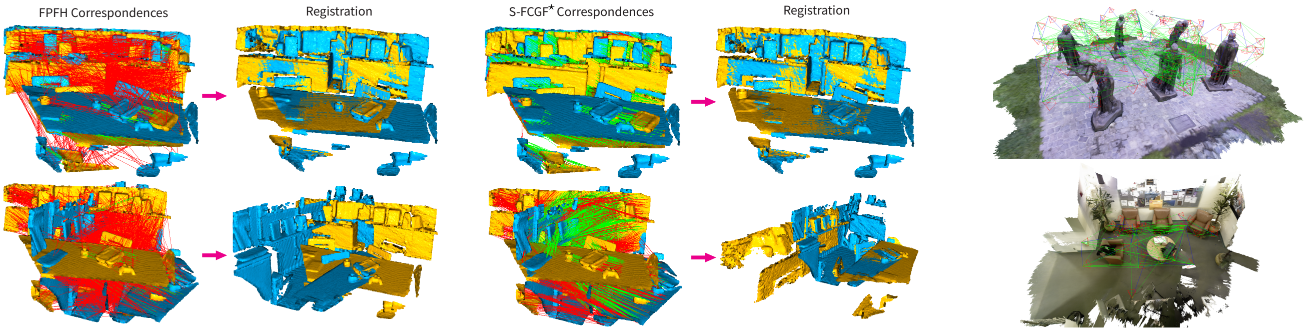


Figure 4. Dynamics of SGP on 3DMatch [92]. PLSR: *Pseudo-Label Survival Rate*. PLIR: *Pseudo-Label Inlier Rate*. BS: Boostrap.

Methods	Kitchen (%)	Home 1 (%)	Home 2 (%)	Hotel 1 (%)	Hotel 2 (%)	Hotel 3 (%)	Study (%)	MIT (%)	Overall (%)
FPFH+RANSAC10K [64][a]	80.6	84.6	69.2	88.1	76.9	**88.9**	71.2	70.1	78.4
FCGF [22][b]	93.0	91.0	71.0	91.0	87.0	69.0	75.0	80.0	82.0
DGR [20]	94.5	89.7	77.9	92.9	85.6	79.6	69.9	72.7	85.2
DGR+RANSAC80K [20]	**98.8**	96.2	**81.7**	97.3	91.2	87.0	81.9	79.2	91.3
FCGF$^\circ$+RANSAC10K[c]	97.2	**97.4**	77.9	97.8	**91.3**	83.3	86.3	76.6	91.1
S-FCGF^T+RANSAC10K	98.4	94.2	75.0	**98.7**	89.4	79.6	87.3	76.6	90.8
S-FCGF*+RANSAC10K	98.0	94.2	76.0	**98.7**	90.4	85.2	**88.0**	**80.5**	**91.4**

Table 2. Scene-wise and overall recalls on the 3DMatch [92] test dataset using different methods. S-FCGF^T: last FCGF trained by SGP. S-FCGF*: best FCGF trained by SGP.

[a] FPFH implemented in Open3D [96]. All RANSAC use 99.9% confidence.
[b] Recall statistics adapted from the original FCGF paper [22] evaluated with the criteria defined by 3DMatch.
[c] Recall computed by using RANSAC10K with the pretrained FCGF$^\circ$ (*i.e.*, the supervised oracle).

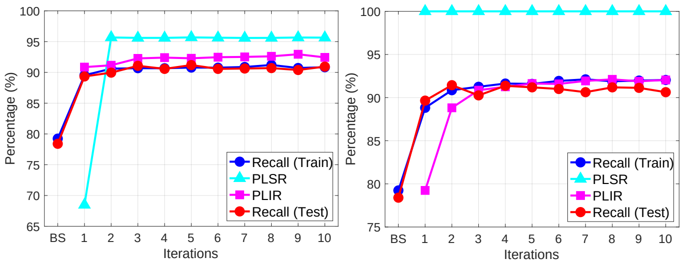


(a) Success (top) and failure (bottom) by FPFH.

(b) Successes by S-FCGF*.

(c) Multi-way registration using S-FCGF*.

Figure 5. Qualitative results showing the improved performance of (b) S-FCGF* over the bootstrap descriptor (a) FPFH for pairwise registration on 3DMatch [92], and (c) cross-dataset generalization of S-FCGF* for multi-way registration on the Stanford RGBD dataset [18]. In (a)-(b), the top pair has overlap ratio 89%, the bottom pair has overlap ratio 50%. Green lines: inlier correspondences. Red lines: outlier correspondences. In (c), top: *Lounge*, bottom: *Burghers*. Blue lines: odometry. Green lines: loop closures. [Best viewed digitally]



(a) retrain = True.

(b) verifyLabel = False.

Figure 6. Dynamics of SGP on 3DMatch [92] with (a) retrain instead of finetune (line 17); (b) the verify (line 13) turned off. SGP still achieves over 91% overall recall on the test set.

Fig. 4. The finetune train recall is slightly higher and more stable than the retrain train recall, due to the "continuous" weight update nature of finetune. SGP with retrain achieves similar performance on the test set: S-FCGF* has overall recall 91.2% and S-FCGF^T has overall recall 90.9%.

We then study the effect of the verifier by running SGP on 3DMatch without verification, *i.e.*, we set $\eta = 0$. As shown in Fig. 6(b), PLSR is always 100%. Despite higher noise in the pseudo-labels, the performance of SGP remains unaffected on the test set: S-FCGF* has overall recall 91.4% and S-FCGF^T has overall recall 90.6%.

In the Supplementary Material, we provide two more ablation studies on 3DMatch: (i) we trained SGP on the small test set and tested S-FCGF on the large training set, to show better generalization of a large training set; (ii) we replaced RANSAC10K with a non-robust registration solver as the teacher to show the importance of a robust solver.

6. Conclusion

We proposed SGP, the first general framework for feature learning in geometric perception without any supervision from ground-truth geometric labels. SGP iteratively performs robust estimation of the geometric models to generate pseudo-labels, and feature learning under the supervision of the noisy pseudo-labels. We applied SGP to camera pose estimation and point cloud registration, demonstrating performance that is on par or even superior to supervised oracles in large-scale real datasets.

Future research includes (i) increasing the training recall towards 100%; (ii) differentiating the robust estimation layer [37]; (iii) designing an optimality-based [84] and learnable verifier based on *cycle consistency* [41, 35, 56]; (iv) speeding up the teacher-student loop; (iv) forming image and point cloud pairs using *image retrieval* [72, 23].

References

[1] Pasquale Antonante, Vasileios Tzoumas, Heng Yang, and Luca Carlone. Outlier-robust estimation: Hardness, minimally-tuned algorithms, and applications. *arXiv preprint arXiv:2007.15109*, 2020. 2

[2] Erik Ask, Olof Enqvist, and Fredrik Kahl. Optimal geometric fitting under the truncated l2-norm. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1722–1729, 2013. 2

[3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5

[4] Daniel Barath and Jiří Matas. Graph-cut RANSAC. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6733–6741, 2018. 2, 5

[5] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1304–1312, 2020. 2, 4, 5

[6] Jean-Charles Bazin, Hongdong Li, In So Kweon, Cédric Demonceaux, Pascal Vasseur, and Katsushi Ikeuchi. A branch-and-bound approach to correspondence and grouping problems. *IEEE Trans. Pattern Anal. Machine Intell.*, 35(7):1565–1576, 2012. 2

[7] J. C. Bazin, Y. Seo, and M. Pollefeys. Globally optimal consensus set maximization through rotation search. In *Asian Conference on Computer Vision*, pages 539–551. Springer, 2012. 2

[8] Dimitri Bertsekas. *Nonlinear programming*. Athena Scientific, 1999. 4, 14

[9] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2), 1992. 6

[10] M. Bosse, G. Agamennoni, and I. Gilitschenski. Robust estimation and applications in robotics. *Foundations and Trends in Robotics*, 4(4):225–269, 2016. 2

[11] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European Conf. on Computer Vision (ECCV)*, pages 536–551. Springer, 2014. 1

[12] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008. 7

[13] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robotics*, 32(6):1309–1332, 2016. 1

[14] Zhipeng Cai, Tat-Jun Chin, and Vladlen Koltun. Consensus maximization tree search revisited. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1637–1645, 2019. 2

[15] Bo Chen, Jiewei Cao, Alvaro Parra, and Tat-Jun Chin. Satellite pose estimation with deep landmark regression and nonlinear pose refinement. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 6, 15

[16] Tat-Jun Chin, Pulak Purkait, Anders Eriksson, and David Suter. Efficient globally optimal consensus maximisation with tree search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2413–2421, 2015. 2

[17] T. J. Chin and D. Suter. The maximum consensus problem: recent algorithmic advances. *Synthesis Lectures on Computer Vision*, 7(2):1–194, 2017. 2

[18] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, 2015. 1, 7, 8, 15, 17

[19] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 15, 17

[20] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7, 8

[21] Christopher Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016. 2

[22] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Intl. Conf. on Computer Vision (ICCV)*, pages 8958–8966, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 14

[23] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *Intl. J. of Robotics Research*, 27(6):647–665, 2008. 8

[24] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017. 1, 6, 7

[25] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 3, 6

[26] Wei Dong, Jaesik Park, Yi Yang, and Michael Kaess. Gpu accelerated robust scene reconstruction. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 7863–7870. IEEE, 2019. 1

[27] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017. 2

[28] Sovann En, Alexis Lechervy, and Frédéric Jurie. Rpnet: An end-to-end network for relative camera pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[29] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *European Conf. on Computer Vision (ECCV)*, pages 738–751. Springer, 2012. 2

[30] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981. 2, 5

[31] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2018. 1

[32] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(8):930–943, 2003. 15

[33] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Intl. Conf. on Computer Vision (ICCV)*, pages 3828–3838, 2019. 2

[34] Zan Gojcic, Caifa Zhou, Jan Dirk Wegner, and Wieser Andreas. The perfect match: 3d point cloud matching with smoothed densities. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 5

[35] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1759–1769, 2020. 8

[36] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. 3

[37] Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks: A new hope. *arXiv preprint arXiv:1909.04866*, 2019. 8

[38] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 529–536, 2005. 2

[39] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 5, 13, 15

[40] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Amer.*, 4(4):629–642, Apr 1987. 5, 15, 18

[41] Xiangru Huang, Zhenxiao Liang, Xiaowei Zhou, Yao Xie, Leonidas J Guibas, and Qixing Huang. Learning transformation synchronization. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8082–8091, 2019. 8

[42] G. Izatt, H. Dai, and R. Tedrake. Globally optimal object pose estimation in point clouds with mixed-integer programming. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2017. 2

[43] Huaizu Jiang, Gustav Larsson, Michael Maire Greg Shakhnarovich, and Erik Learned-Miller. Self-supervised relative depth learning for urban scene understanding. In *European Conf. on Computer Vision (ECCV)*, pages 19–35, 2018. 2

[44] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Machine Intell.*, 2020. 2

[45] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Intl. Conf. on Computer Vision (ICCV)*, pages 153–161, 2017. 2

[46] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 1

[47] Laurent Kneip, Hongdong Li, and Yongduek Seo. UPnP: An optimal o(n) solution to the absolute pose problem with universal applicability. In *European Conf. on Computer Vision (ECCV)*, pages 127–142. Springer, 2014. 15

[48] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4681–4690, 2017. 2

[49] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013. 2

[50] Hongdong Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1074–1080. IEEE, 2009. 2

[51] Jiaxin Li and Gim Hee Lee. USIP: Unsupervised stable interest point detection from 3d point clouds. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 361–370, 2019. 3

[52] Yin Li, Manohar Paluri, James M Rehg, and Piotr Dollár. Unsupervised learning of edges. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1619–1627, 2016. 2

[53] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2041–2050, 2018. 1, 2, 6, 7, 15, 16

[54] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4571–4580, 2019. 2

[55] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. J. of Computer Vision*, 60(2):91–110, 2004. 1, 2, 3, 5, 7

[56] Joshua G Mangelson, Derrick Dominic, Ryan M Eustice, and Ram Vasudevan. Pairwise consistent measurement set maximization for robust multi-robot map merging. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2916–2923. IEEE, 2018. 8

[57] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 675–687. Springer, 2017. 1

[58] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Trans. Robotics*, 33(5):1255–1262, 2017. 2

[59] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(6):756–770, 2004. 5

[60] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *ICCV*, 2017. 15, 17

[61] Á. Parra Bustos and T. J. Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Trans. Pattern Anal. Machine Intell.*, 40(12):2868–2882, 2018. 1, 2, 4

[62] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise voting network for 6dof pose estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4561–4570, 2019. 1, 6, 15

[63] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2213–2222, 2017. 2

[64] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3212–3217. IEEE, 2009. 1, 2, 5, 6, 8

[65] Tanner Schmidt, Richard Newcombe, and Dieter Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2016. 1

[66] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1, 2

[67] Jingnan Shi, Heng Yang, and Luca Carlone. ROBIN: a Graph-Theoretic Approach to Reject Outliers in Robust Estimation using Invariants. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2021. 2

[68] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 15, 17

[69] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 292–301, 2018. 1, 6, 15

[70] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *Intl. Conf. on Computer Vision (ICCV)*, 2019. 2

[71] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 661–669, 2017. 2

[72] Carl Toft, Will Maddern, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, et al. Long-term visual localization revisited. *IEEE Trans. Pattern Anal. Machine Intell.*, 2020. 8

[73] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2022–2030, 2018. 2

[74] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *European Conf. on Computer Vision (ECCV)*, 2020. 1, 2, 3, 4, 5, 6, 7, 13

[75] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Intl. Conf. on Computer Vision (ICCV)*, pages 3523–3532, 2019. 1

[76] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*, 2020. 2

[77] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018. 1

[78] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10687–10698, 2020. 2

[79] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conf. on Computer Vision (ECCV)*, 2020. 1

[80] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 2020. 2

[81] Heng Yang and Luca Carlone. A quaternion-based certifiably optimal solution to the Wahba problem with outliers. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1665–1674, 2019. 2

[82] Heng Yang and Luca Carlone. A polynomial-time solution for robust registration with extreme outlier rates. In *Robotics: Science and Systems (RSS)*, 2019. 1, 2

[83] Heng Yang and Luca Carlone. In perfect shape: Certifiably optimal 3D shape reconstruction from 2D landmarks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[84] Heng Yang and Luca Carlone. One ring to rule them all: Certifiably robust geometric perception with outliers. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2, 4, 8

[85] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. Robotics*, 2020. 1, 2, 4, 5

[86] Jiaolong Yang, Hongdong Li, and Yunde Jia. Optimal essential matrix estimation via inlier-set maximization. In *European Conf. on Computer Vision (ECCV)*, pages 111–126. Springer, 2014. 2

[87] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1281–1292, 2020. 2

[88] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995. 2

[89] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *European Conf. on Computer Vision (ECCV)*, 2018. 2

[90] Wentao Yuan, Ben Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. DeepGMR: Learning Latent Gaussian Mixture Models for Registration. 2020. 1

[91] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. DPOD: 6d pose object detector and refiner. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1941–1950, 2019. 1, 4, 6, 15

[92] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and T Funkhouser. 3dmatch: Learning the matching of local 3d geometry in range scans. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 1, 2, 6, 8, 15, 17, 18, 19

[93] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conf. on Computer Vision (ECCV)*, pages 649–666. Springer, 2016. 2

[94] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 689–696. IEEE, 2009. 3

[95] Qian-Yi. Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conf. on Computer Vision (ECCV)*, pages 766–782. Springer, 2016. 2, 3, 5

[96] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 1, 2, 7, 8, 16

[97] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2017. 2

[98] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*, 2020. 2

Supplementary Material

A1. Proof of Proposition 1

Proof. We prove Proposition 1 for Examples 1-2 separately.

Example 1: Relative Pose Estimation. In relative pose estimation, the known geometric model for the i-th measurement pair is $\boldsymbol{R}_i^\circ \in \mathrm{SO}(3)$ and $\boldsymbol{t}_i^\circ \in \mathbb{S}^2$, where $\boldsymbol{R}_i^\circ$ is the relative rotation, and $\boldsymbol{t}_i^\circ$ is the up-to-scale relative translation between two images $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$. Using $(\boldsymbol{R}^\circ, \boldsymbol{t}^\circ)$, we can form the *essential matrix* $\boldsymbol{E}_i^\circ \doteq [\boldsymbol{t}_i^\circ]_\times \boldsymbol{R}_i^\circ$, from which we further compute the *fundamental matrix* $\boldsymbol{F}_i^\circ \doteq (\boldsymbol{K}_i^b)^{-T} \boldsymbol{E}_i^\circ (\boldsymbol{K}_i^a)^{-1}$, where $\boldsymbol{K}_i^a$, $\boldsymbol{K}_i^b$ are the camera intrinsics for the two images $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ [39]. Now we let the residual function $r(\cdot)$ be the algebraic error [39]:

$$r(\boldsymbol{F}_i^\circ, \tilde{\boldsymbol{p}}_{i,k}, \tilde{\boldsymbol{q}}_{i,k}^b) = (\tilde{\boldsymbol{q}}_{i,k}^b)^\mathsf{T} \boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k}, \tag{A1}$$

which should vanish if there is no measurement noise, and $\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}}^b \in \mathbb{R}^3$ denotes the homogeneous coordinates of the keypoint locations. In eq. (A1), $\boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k}$ is called the *epipolar line* (in fact, $\boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k}$ represents the normal vector of the plane formed by the epipolar line and the camera optical center).

Because we have adopted a TLS cost function, *i.e.* $\rho(r) = \min\left\{r^2, \bar{c}^2\right\}$ (and assume $\bar{c}^2$ is small), obviously, the global minimizer of problem (7) is the following:

$$\boldsymbol{q}_{i,k}^b = \mathcal{C}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i, \boldsymbol{p}_i^b, \boldsymbol{b}_i) \in \begin{cases} \text{the epipolar line } \boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k} & \text{if the epipolar line intersects } \boldsymbol{b}_i \\ \boldsymbol{b}_i & \text{otherwise} \end{cases}, \tag{A2}$$

which says that the predicted keypoint $\boldsymbol{q}_{i,k}^b$ should lie precisely on the epipolar line if the epipolar line has a nonempty intersection with the image $\boldsymbol{b}_i$ (so that the residual (A1) is zero and $\rho(r) = 0$), or it can be an arbitrary point on the image otherwise (so that the residual (A1) is nonzero and $\rho(r) = \bar{c}^2$ is very small). In [74], the authors designed another constraint that enforces cycle consistency, *i.e.*, the back-predicted keypoint of the predicted keypoint should be the original keypoint:

$$\mathcal{C}(\boldsymbol{q}_{i,k}^b, \boldsymbol{b}_i, \boldsymbol{p}_i^a, \boldsymbol{a}_i) = \boldsymbol{p}_{i,k}^a. \tag{A3}$$

Combining eq. A2 and (A3), we can reformulate the original feature learning problem (7) as:

$$\text{find} \quad \mathcal{C}_{\boldsymbol{\theta}} \tag{A4}$$

$$s.t. \quad \mathcal{C} \text{ satisfies (A2) and (A3)}, \tag{A5}$$

which enforces the correspondence function $\mathcal{C}$ (parametrized by $\boldsymbol{\theta} \in \mathbb{R}^{N_\mathcal{C}}$) to map keypoints in $\boldsymbol{a}_i$ to their corresponding epipolar lines (if the epipolar line exists) in $\boldsymbol{b}_i$, and to map the predicted keypoints in $\boldsymbol{b}_i$ back to their original keypoints, which is connected to the cross check criteria mentioned in the main text.

The reformulated problem (A4) is a constrained optimization problem that is not suitable for training neural networks. Therefore, the last step we do is to move the constraints to the cost function and penalize the *violation* of the constraints, which is commonly referred to as the *Augmented Lagrangian Method* (ALM), or the *penalty method*:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{N_\mathcal{C}}} \sum_{i=1}^{M} \sum_{k=1}^{N_{a_i}} \lambda_{\text{epipolar}} \cdot \textbf{dist}\left(\underbrace{\mathcal{C}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i, \boldsymbol{p}_i^b, \boldsymbol{b}_i)}_{\boldsymbol{q}_{i,k}^b}, \boldsymbol{F}_i^\circ \tilde{\boldsymbol{p}}_{i,k} \right)^2 + \lambda_{\text{cycle}} \cdot \textbf{dist}\left(\mathcal{C}\left(\underbrace{\mathcal{C}(\boldsymbol{p}_{i,k}^a, \boldsymbol{a}_i, \boldsymbol{p}_i^b, \boldsymbol{b}_i)}_{\boldsymbol{q}_{i,k}^b}, \boldsymbol{b}_i, \boldsymbol{p}_i^a, \boldsymbol{a}_i \right), \boldsymbol{p}_{i,k}^a \right)^2, \tag{A6}$$

where $\lambda_{\text{epipolar}}, \lambda_{\text{cycle}} > 0$ are constants chosen by the user. Finally, let the correspondence function be the form in (4), we recover the loss function in the CAPS paper [74].[10] Therefore, the CAPS neural network can be seen as a method to solve the feature learning problem (7) by solving its Augmented Lagrangian (A6).

Example 2: Point Cloud Registration. In point cloud registration, the known geometric model for the i-th measurement pairs is the rigid transformation $\boldsymbol{R}_i^\circ \in \mathrm{SO}(3)$ and $\boldsymbol{t}_i^\circ \in \mathbb{R}^3$ between the two point clouds $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$. Let the residual function $r(\cdot)$ be the Euclidean distance:

$$r(\boldsymbol{R}_i^\circ, \boldsymbol{t}_i^\circ, \boldsymbol{p}_{i,k}^a, \boldsymbol{q}_{i,k}^b) = \left\| \boldsymbol{q}_{i,k}^b - \boldsymbol{R}_i^\circ \boldsymbol{p}_{i,k}^a - \boldsymbol{t}_i^\circ \right\|, \tag{A7}$$

[10] The $\textbf{dist}\,(\cdot)$ function in (A6) is equivalent to the ℓ_2 norm $\|\cdot\|$. [74] used the $\textbf{dist}\,(\cdot)$ instead of $\textbf{dist}\,(\cdot)^2$. This can be easily seen as the Augmented Lagrangian if using the constraint $\sqrt{\textbf{dist}\,(\cdot)} = 0$, instead of $\textbf{dist}\,(\cdot) = 0$.

which should be zero without measurement noise. Under the TLS cost function $\rho(r) = \min\left\{r^2, \bar{c}^2\right\}$, the global minimizer of problem (7) is

$$\boldsymbol{q}^b_{i,k} = \mathcal{C}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i, \boldsymbol{p}^b_i, \boldsymbol{b}_i) = \begin{cases} \arg\min\limits_{\boldsymbol{p}^b_{i,j} \in \boldsymbol{p}^b_i} r(\boldsymbol{R}^\circ_i, \boldsymbol{t}^\circ_i, \boldsymbol{p}^a_{i,k}, \boldsymbol{p}^b_{i,j}) & \text{if } \min\limits_{\boldsymbol{p}^b_{i,j} \in \boldsymbol{p}^b_i} r(\boldsymbol{R}^\circ_i, \boldsymbol{t}^\circ_i, \boldsymbol{p}^a_{i,k}, \boldsymbol{p}^b_{i,j}) < \bar{c} \\ \emptyset & \text{otherwise} \end{cases}, \tag{A8}$$

which states that the correspondence function $\mathcal{C}$ should output the nearest neighbor of $(\boldsymbol{R}^\circ_i \boldsymbol{p}^a_{i,k} + \boldsymbol{t}^\circ_i)$ in $\boldsymbol{p}^b_i$ if the Euclidean distance between the nearest neighbor and $(\boldsymbol{R}^\circ_i \boldsymbol{p}^a_{i,k} + \boldsymbol{t}^\circ_i)$ is close enough to be considered as an inlier, and outputs nothing otherwise (*i.e.*, $\boldsymbol{p}^a_{i,k}$ does not have a corresponding point in $\boldsymbol{p}^b_i$). Therefore, we can reformulate problem (7) as:

$$\text{find} \quad \mathcal{C} \tag{A9}$$

$$s.t. \quad \mathcal{C} \text{ satisfies (A8)}. \tag{A10}$$

We then use the fact that $\mathcal{C}$ is a composition of a feature descriptor and nearest neighbor search in the feature space (*cf.* eq. (5) in Example 2), and hence, problem (A9) is further equivalent to finding a descriptor $\mathcal{F}$ such that:

$$\text{find} \quad \mathcal{F} \tag{A11}$$

$$s.t. \quad \textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{q}^b_{i,k}, \boldsymbol{b}_i)\right) \leq \textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{p}^b_{i,j}, \boldsymbol{b}_i)\right), \forall \boldsymbol{p}^b_{i,j} \neq \boldsymbol{q}^b_{i,k}, \tag{A12}$$

which precisely states that the distance in the feature space between $\boldsymbol{p}^a_{i,k}$ and the corresponding keypoint $\boldsymbol{q}^b_{i,k}$ is smaller than the distance between $\boldsymbol{p}^a_{i,k}$ and any other point in $\boldsymbol{p}^b_i$. In fact, we can ask for stronger conditions on the feature descriptor $\mathcal{F}$:

$$\text{find} \quad \mathcal{F} \tag{A13}$$

$$s.t. \quad \textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{q}^b_{i,k}, \boldsymbol{b}_i)\right) \leq m_p, \tag{A14}$$

$$\textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{p}^b_{i,j}, \boldsymbol{b}_i)\right) \geq m_n, \forall \boldsymbol{p}^b_{i,j} \neq \boldsymbol{q}^b_{i,k}, \tag{A15}$$

$$\textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{q}^b_{i,k}, \boldsymbol{b}_i)\right) \leq m + \textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{p}^b_{i,j}, \boldsymbol{b}_i)\right), \tag{A16}$$

that says: (i) the feature distance between the matched keypoint pair $\boldsymbol{p}^a_{i,k}$ and $\boldsymbol{q}^b_{i,k}$ has to be smaller than a predefined margin $m_p > 0$ (eq. (A14)); (ii) the feature distance between $\boldsymbol{p}^a_{i,k}$ and all the other non-matched keypoints has to be larger than a predefined margin $m_n > m_p$ (eq. (A15)); (iii) the feature distance between non-matched keypoint pairs has to be at least m larger than the feature distance between matched keypoint pairs (eq. (A16)). Obviously, conditions (A14)-(A16) are sufficient (but not necessary) for ensuring condition (A12).

Again, problem (A13) is a constrained optimization that is not suitable for neural network training. Therefore, we develop its Augmented Lagrangian (for the constraints related to the keypoint $\boldsymbol{p}^a_{i,k}$) to be:

$$\begin{aligned} \mathcal{L}(\boldsymbol{p}^a_{i,k}, s_p, s_n, s) = {} & \lambda_p \left(m_p - \textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{q}^b_{i,k}, \boldsymbol{b}_i)\right) - s_p\right)^2 + \\ & \sum_{\boldsymbol{p}^b_{i,j} \neq \boldsymbol{q}^b_{i,k}} \lambda_n \left(\textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{p}^b_{i,j}, \boldsymbol{b}_i)\right) - m_n - s_n\right)^2 + \\ & \sum_{\boldsymbol{p}^b_{i,j} \neq \boldsymbol{q}^b_{i,k}} \lambda \left(\textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{p}^b_{i,j}, \boldsymbol{b}_i)\right) - \textbf{dist}\left(\mathcal{F}(\boldsymbol{p}^a_{i,k}, \boldsymbol{a}_i), \mathcal{F}(\boldsymbol{q}^b_{i,k}, \boldsymbol{b}_i)\right) - m - s\right)^2, \end{aligned} \tag{A17}$$

where $s_p, s_n, s \geq 0$ are nonnegative slack variables. In eq. (A17), the first two terms denote the *contrastive loss*, while the last term denotes the *triplet loss*. The ALM [8] solves the following optimization:

$$\min_{\mathcal{F}, s_p \geq 0, s_n \geq 0, s \geq 0} \sum_{i=1}^{M} \sum_{k=1}^{N_{a_i}} \mathcal{L}(\boldsymbol{p}^a_{i,k}, s_p, s_n, s). \tag{A18}$$

Finally, by enforcing $s_p = s_n = s = 0$, problem (A18) recovers the metric learning problem in the FCGF paper [22]. Therefore, the FCGF neural network can be seen as a method to solve the feature learning problem (7) by solving the Augmented Lagrangian (A18). □

A2. Application of SGP on Object Detection and Pose Estimation

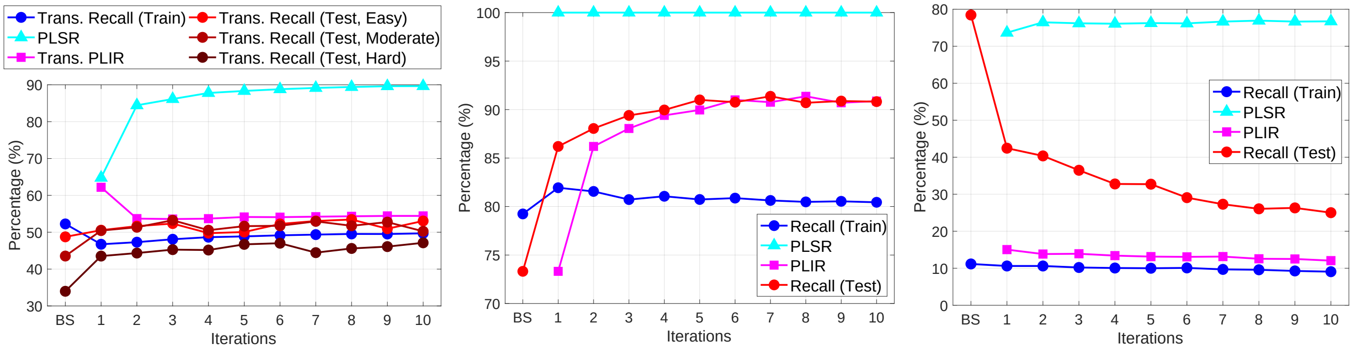
Example A1 (Object Detection and Pose Estimation)**.** *Given a collection of 3D models* $\{\boldsymbol{a}_i\}_{i=1}^{O}$*, where each model* $\boldsymbol{a}_i \in \mathbb{R}^{3\times N_{a_i}}$ *consists of a set of 3D keypoints, let* $\boldsymbol{a} \in \mathbb{R}^{3\times N_a}, N_a = \sum_{i=1}^{O} N_{a_i}$*, be the concatenation of all 3D keypoints. In addition, given a corpus of 2D images* $\{\mathbf{b}_i\}_{i=1}^{M}$*, where each* $\mathbf{b}_i$ *is an RGB image that contains the (partial, occluded) projections of the 3D models plus some background. Object detection and pose estimation seeks to jointly learn a keypoint prediction function* $\mathcal{C}$ *and estimate the poses of the 3D models* $\boldsymbol{x}_i = \{(\boldsymbol{R}_{i,j}, \boldsymbol{t}_{i,j})\}_{j\in\mathcal{S}\subset[O]} \in (\mathrm{SO}(3)\times\mathbb{R}^3)^{|\mathcal{S}|}$*, where* $\mathcal{S} \subset [O]$ *is the subset of 3D models observed by the* i*-th 2D image (*$|\mathcal{S}|$ *denotes the cardinality of the set). In particular, following [91], let* $\mathcal{C}$ *be a combination of UVW mapping and semantic ID masking,* i.e.*, for each pixel in* $\mathbf{b}_i$*,* $\mathcal{C}$ *predicts which 3D model it belongs to (from* 1 *to* O*, and* 0 *for background), and what is the corresponding 3D coordinates in the specific model, thus deciding which point in* $\boldsymbol{a}$ *is the corresponding 3D point.*[11]

SGP **for Example A1**. The teacher performs robust absolute pose estimation, *a.k.a. perspective-n-point* (PnP) [39, 47]. A good candidate for the teacher is RANSAC and its variants (*e.g.*, using P3P [32]). The student trains a 2D keypoint predictor under the supervision of camera poses. Recent works such as YOLO6D [69], PVNet [62], and DPOD [91] can all serve as the student network, despite using different methodologies. As for the verifier, similar to Example 1, it can be designed based on the estimated inlier rate by RANSAC. Alternatively, one can project the 3D models onto the 2D image using the estimated absolute poses and compute the overlap ratio (in terms of pixels) between the 2D projection and the estimated semantic ID mask. To initialize SGP, we can train a bootstrap predictor using synthetic datasets, *i.e.*, by rendering synthetic projections of the 3D models under different simulated poses, which is common in [91, 62, 69, 15].

A3. Detailed Experimental Data

A3.1. Relative Pose Estimation

In Section 5.1, Fig. 2 plots the rotation statistics for running SGP on the MegaDepth [53] dataset for relative pose estimation. Here in Fig. A1(a), we plot the translation statistics. In addition, the full statistics of SGP are tabulated in Table A1. Fig. A2 visualizes 9 qualitative examples of relative pose estimation using S-CAPS^{T}on the MegaDepth test set.



(a) SGP translation statistics on MegaDepth. (b) SGP on 3DMatch with exchanged train and test. (c) SGP on 3DMatch with Horn as teacher.

Figure A1. Supplementary statistics. (a) The translation statistics for using SGP on MegaDepth [53] (rotation statistics shown in Fig. 2 in the main text). (b) Dynamics of SGP on 3DMatch [92] with training and test sets exchanged, *i.e.*, we train SGP on the smaller test set ($1,623$ pairs), but test S-FCGF on the larger training set ($9,856$ pairs). (c) Dynamics of SGP on 3DMatch by replacing the original RANSAC10K teacher with a non-robust Horn's method [40] as the teacher.

A3.2. Point Cloud Registration

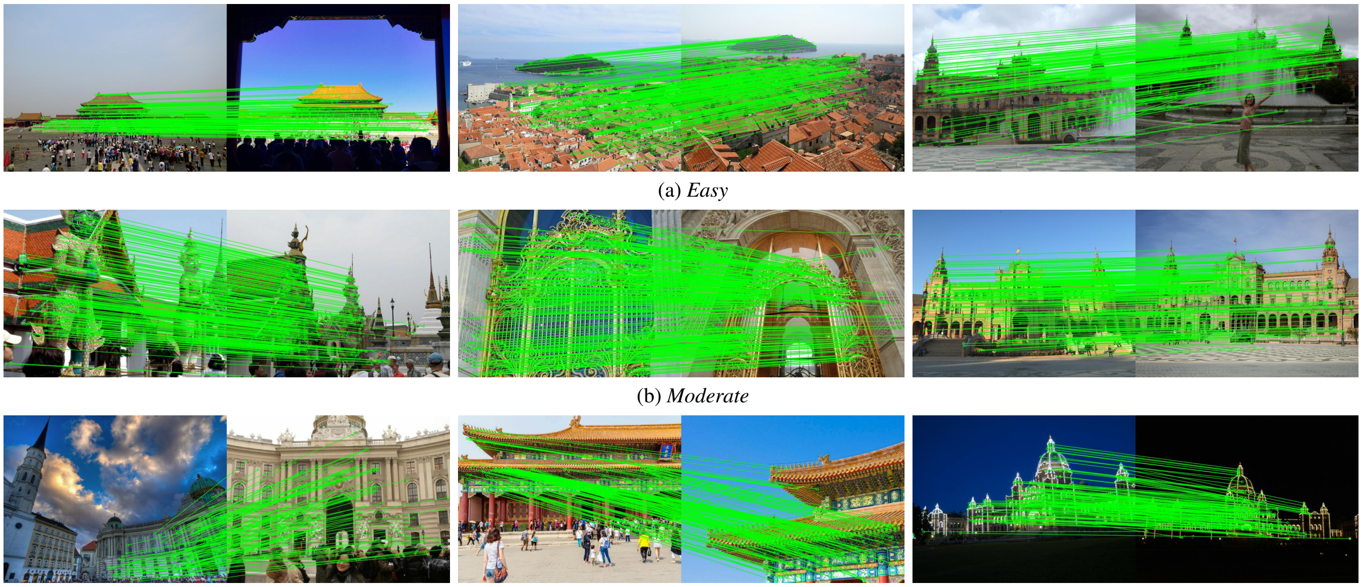
In Section 5.2, Fig. 4 plots the dynamics of runing SGP on the 3DMatch [92] dataset. Here we provide the full statistics in Table A2.

For qualitative results, in Fig. A3 we showcase multiway registration results on various RGB-D datasets [68, 18, 19, 60] in addition to Fig. 5. With S-FCGF, rich loop closures can be detected (in green lines), ensuring high-fidelity camera poses for dense reconstruction. It is worth noting that global registration with trained S-FCGF+RANSAC10K, unlike DGR, can easily

[11] There are many different ways to establish 2D-3D correspondences, see PVNet [62], YOLO6D [69] and references therein.

	Statistics (%)	SIFT Bootstrap	SGP trained CAPS (S-CAPS) 1	2	3	4	5	6	7	8	9	10
Train	PLSR	$**$	64.79	84.44	86.14	87.78	88.34	88.80	89.15	89.41	89.62	89.64
	Rot. PLIR	$**$	92.50	88.83	88.35	88.41	88.25	88.52	88.50	88.57	88.43	88.48
	Rot. Recall	87.75	79.33	79.69	80.62	80.70	81.20	81.34	81.57	81.57	81.56	81.68
	Trans. PLIR	$**$	62.20	53.70	53.58	53.68	54.13	54.09	54.22	54.33	54.43	54.43
	Trans. Recall	52.25	46.74	47.30	48.09	48.66	48.87	49.16	49.36	49.54	49.52	49.70
Test Recall	Rot., *Easy*	80.88	85.39	85.49	84.68	85.69	85.79	85.79	86.29	**87.09**	85.49	86.29
	Rot., *Moderate*	58.06	**70.37**	68.27	**70.37**	69.77	69.67	69.27	69.87	68.87	70.07	69.17
	Rot., *Hard*	40.35	48.36	49.38	50.31	49.59	50.10	**51.75**	50.10	50.72	51.23	51.33
	Trans., *Easy*	48.75	50.55	51.65	52.35	49.75	50.05	52.25	53.05	**53.45**	50.95	53.05
	Trans., *Moderate*	43.54	50.45	51.35	**53.25**	50.55	51.65	51.75	52.95	51.75	52.75	50.25
	Trans., *Hard*	33.98	43.53	44.35	45.28	45.17	46.71	47.02	44.46	45.60	46.10	**47.13**

Table A1. Train and test statistics of running SGP on MegaDepth [53]. SGP setting: retrain = False, verifyLabel = True, verifier criteria: number of matches larger than 100 and RANSAC estimated inlier rate larger than 10%. Rotation statistics plotted in Fig. 2 in the main text. Translation statistics plotted in Fig. A1(a).



(a) *Easy*

(b) *Moderate*

(c) *Hard*

Figure A2. Supplementary qualitative results for relative pose estimation on the MegaDepth dataset [53] using S-CAPS^T.

run in parallel on a single graphics card due to its inexpensive memory cost. This results in at least 4× speedup comparing to DGR in practice when multi-thread loop closure detection is enabled [96].

A3.3. Ablation Study

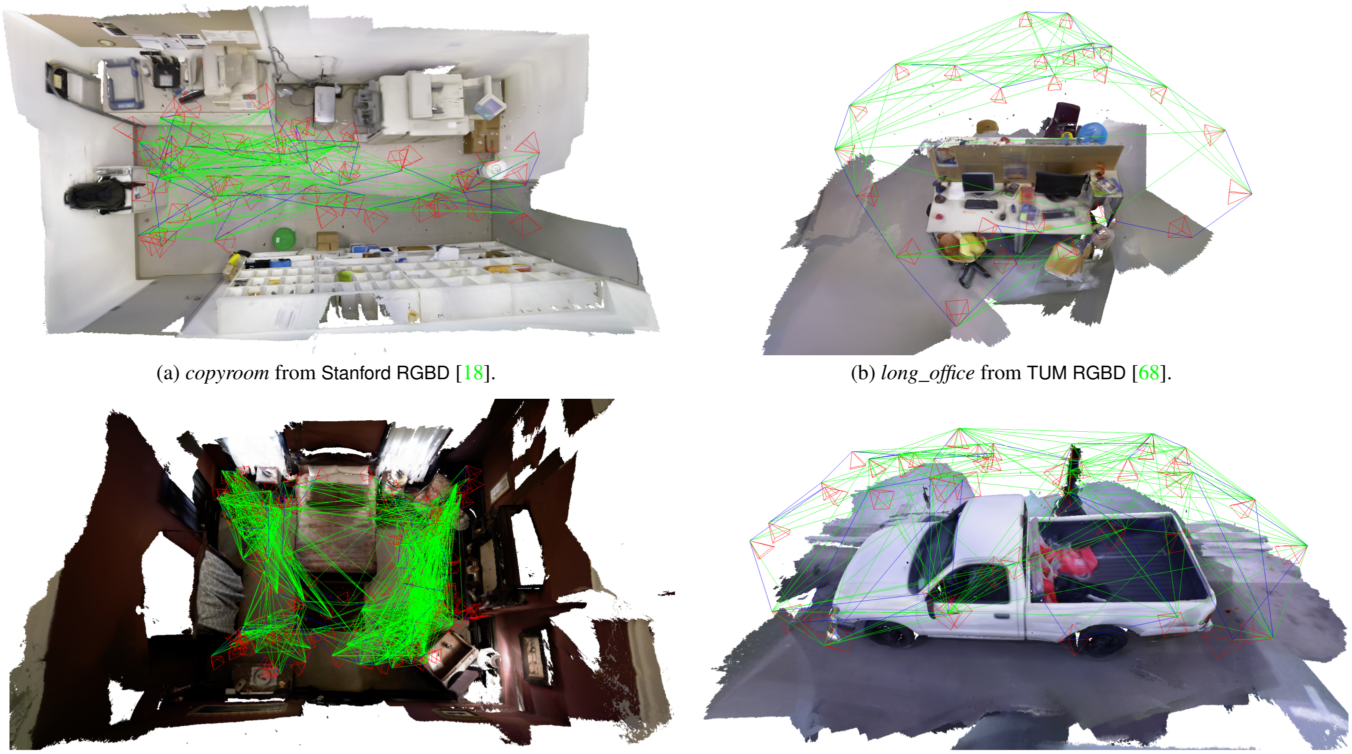
In Section 5.3, Fig. 6 plots the dynamics of running SGP on 3DMatch with two different algorithmic settings: (a) set retrain =True and use retrain instead of finetune; (b) set verifyLabel = False and turn off the verifier. Here we provide the full statistics for (a) and (b) in Table A3 and Table A4, respectively.

Additionally, we show results for two extra ablation experiments on the 3DMatch dataset for point cloud registration.

Exchange the training and test sets. Because SGP requires no ground-truth pose labels, there is no fundamental difference between the training and test set, except that the training set ($9,856$ pairs) is much larger than the test set ($1,623$ pairs). Therefore, we ask the question: *Can* SGP *learn an equally good feature representation from the much smaller test set?* Our answer is: *it depends on the purpose*. We performed an experiment where we trained SGP on the test set, and tested

	Statistics (%)	FPFH Bootstrap	SGP trained FCGF (S-FCGF) 1	 2	 3	 4	 5	 6	 7	 8	 9	 10
Train	PLSR	**	69.98	73.91	95.53	95.69	95.74	95.73	95.75	95.73	95.76	95.77
	PLIR	**	92.03	93.42	92.19	92.82	93.02	93.25	93.24	93.43	93.41	93.39
	Recall	82.68	89.14	90.92	91.14	91.43	91.76	91.78	91.95	91.97	91.95	92.05
Test Recall	Kitchen	80.63	98.42	98.02	98.22	98.02	98.22	98.42	98.02	97.83	98.62	98.42
	Home 1	84.62	92.31	93.59	91.03	93.59	92.95	94.23	94.23	94.23	94.23	94.23
	Home 2	69.23	77.88	74.04	75.48	75.00	75.96	73.08	75.96	76.92	73.08	75.00
	Hotel 1	88.05	96.90	97.35	98.23	97.79	98.23	99.12	98.67	98.67	98.23	98.67
	Hotel 2	76.92	87.50	85.58	86.54	90.38	89.42	90.38	90.38	89.42	89.42	89.42
	Hotel 3	88.89	85.19	83.33	83.33	79.63	81.48	79.63	85.19	79.63	77.78	79.63
	Study	71.23	85.27	86.30	87.67	86.99	85.96	86.99	88.01	86.99	86.30	87.33
	MIT	70.13	79.22	79.22	80.52	77.92	77.92	77.92	80.52	76.62	79.22	76.62
	Overall	78.44	90.57	90.14	90.63	90.57	90.57	90.70	**91.37**	90.82	90.45	90.82

Table A2. Train and test statistics of running SGP on 3DMatch [92]. SGP setting: retrain = False, verifyLabel = True, verifier overlap ratio threshold η: $\eta = 30\%$ for iterations $\tau = 1, 2$, $\eta = 10\%$ for iterations $\tau = 3, \dots, 10$. Statistics plotted in Fig. 4 in the main text.



(a) *copyroom* from Stanford RGBD [18].

(b) *long_office* from TUM RGBD [68].

(c) *bedroom* from Indoor LIDAR RGBD [60].

(d) *truck* from Redwood Objects [19].

Figure A3. Supplementary qualitative results for 3D registration. Multi-way reconstruction using S-FCGF+RANSAC10K as the global registration method succeeds on various unseen RGB-D datasets. Blue lines: odometry. Green lines: loop closures.

the learned S-FCGF representation on the much larger training set. For SGP we used retrain = False and verifyLabel = False. Fig. A1(b) plots the dynamics and Table A5 provides the full statistics. Two observations can be made: (i) Exchanging the training and test set has almost no effect on the recall of S-FCGF on the test set (*cf.* Table A5 vs Table A2-A4). This means that, if one only cares about the performance of the learned representation on the test set, then running SGP directly on the target test set is sufficient. (ii) Although exchanging the training and test set does not hurt the recall on the test set, it indeed decreases the recall on the training set by more than 10%. This suggests that a small training set has the shortcoming of

		FPFH	SGP trained FCGF (S-FCGF)									
Statistics (%)		Bootstrap	1	2	3	4	5	6	7	8	9	10
Train	PLSR	$**$	68.48	95.68	95.61	95.61	95.69	95.64	95.60	95.61	95.67	95.65
	PLIR	$**$	90.86	91.16	92.27	92.40	92.29	92.47	92.52	92.61	92.95	92.44
	Recall	79.24	89.53	90.60	90.69	90.68	90.79	90.77	90.88	91.20	90.72	90.84
Test Recall	Kitchen	$**$	97.23	97.63	98.22	97.83	98.42	97.83	97.83	97.23	98.42	98.22
	Home 1	$**$	91.67	93.59	94.23	95.51	94.87	93.59	95.51	95.51	91.03	93.59
	Home 2	$**$	73.56	71.63	76.92	73.56	75.00	74.04	72.60	76.44	75.00	75.00
	Hotel 1	$**$	96.90	96.90	96.90	96.46	96.90	96.46	96.90	98.23	97.35	96.90
	Hotel 2	$**$	85.58	89.42	92.31	88.46	87.50	90.38	88.46	88.46	86.54	91.35
	Hotel 3	$**$	85.19	88.89	83.33	81.48	83.33	83.33	83.33	85.19	85.19	83.33
	Study	$**$	82.88	84.59	86.64	88.36	88.70	87.67	87.67	86.30	87.33	86.64
	MIT	$**$	85.71	83.12	79.22	79.22	83.12	80.52	83.12	77.92	77.92	84.42
	Overall	$**$	89.34	89.96	91.07	90.57	**91.19**	90.57	90.63	90.70	90.39	90.94

Table A3. Train and test statistics of running SGP on 3DMatch [92]. SGP setting: retrain = True, verifyLabel = True, verifier overlap ratio threshold η: $\eta = 10\%$ for all iterations $\tau = 1, \ldots, 10$. Statistics plotted in Fig. 6(a) in the main text.

		FPFH	SGP trained FCGF (S-FCGF)									
Statistics (%)		Bootstrap	1	2	3	4	5	6	7	8	9	10
Train	PLSR	$**$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	PLIR	$**$	79.24	88.82	90.86	91.25	91.63	91.59	91.93	92.12	91.89	91.97
	Recall	79.24	88.82	90.86	91.25	91.63	91.59	91.93	92.12	91.89	91.97	92.05
Test Recall	Kitchen	$**$	97.43	98.22	98.62	97.83	98.62	98.81	98.22	98.62	98.22	98.22
	Home 1	$**$	92.31	94.23	91.67	94.23	94.23	92.95	93.59	93.59	94.87	92.95
	Home 2	$**$	74.04	75.00	72.12	77.40	74.04	74.04	73.56	74.04	73.56	73.08
	Hotel 1	$**$	95.58	98.23	97.35	97.79	99.12	98.23	98.67	96.90	97.79	97.35
	Hotel 2	$**$	90.38	93.27	88.46	90.38	88.46	87.50	86.54	88.46	88.46	89.42
	Hotel 3	$**$	88.89	85.19	83.33	87.04	85.19	85.19	81.48	85.19	83.33	81.48
	Study	$**$	84.59	87.33	87.67	86.64	88.01	88.01	87.67	88.70	88.70	87.67
	MIT	$**$	76.62	83.12	77.92	84.42	79.22	80.52	80.52	84.42	83.12	83.12
	Overall	$**$	89.65	**91.44**	90.26	91.37	91.19	91.00	90.63	91.19	91.13	90.63

Table A4. Train and test statistics of running SGP on 3DMatch [92]. SGP setting: retrain = False, verifyLabel = False. Statistics plotted in Fig. 6(b) in the main text.

overfitting and the learned representation fails to generalize to a larger dataset. Therefore, if one cares generalization of the learned representation, then a larger training set is still preferred. Nevertheless, this ablation study demonstrates the power of the alternating minimization nature of SGP, that is, SGP is able to find a sufficiently good local minimum.

Use a non-robust solver as the teacher. All the experiments so far showed successes of the teacher-student loop, and the robustness of the SGP algorithm to imperfections of both the student and the teacher (noisy geometric pseudo-labels). However, we ask another question: *Can we, intentionally, make* SGP *fail?* Our answer is: yes if we try badly. We performed an experiment running SGP on 3DMatch, this time replacing RANSAC10K with the non-robust Horn's method [40]. We remark that Horn's method is a subroutine of RANSAC and in practice nobody would use Horn's method alone in the presence of outlier correspondences. Nevertheless, for the purpose of ablation study, we adopted this pessimistic choice. Again, for SGP we used retrain = False, verifyLabel = True with a constant overlap ratio threshold $\eta = 10\%$. Fig. A1(c) shows the dynamics. We see that the PLIR is always below 20%, meaning that 8 out of 10 geometric labels passed to FCGF training are wrong. In this case, the learned S-FCGF representation keeps getting worse, as shown by the decreasing recalls on both the training and test set. Note that for testing, we actually used RANSAC10K as the registration solver to be consistent with other experiments we

		FPFH	SGP trained FCGF (S-FCGF)									
Statistics (%)		Bootstrap	1	2	3	4	5	6	7	8	9	10
Train	PLSR	∗∗	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	PLIR	∗∗	73.32	86.20	88.05	89.40	89.96	91.00	90.76	91.37	90.70	90.88
	Recall	73.32	86.20	88.05	89.40	89.96	91.00	90.76	**91.37**	90.70	90.88	90.82
	Kitchen	∗∗	94.66	96.84	98.42	98.81	99.21	99.60	99.41	99.01	99.21	99.21
	Home 1	∗∗	91.03	89.74	93.59	95.51	95.51	94.87	94.87	95.51	96.15	95.51
	Home 2	∗∗	70.67	70.67	71.63	69.23	71.15	70.19	74.04	72.60	72.12	72.60
	Hotel 1	∗∗	94.69	96.02	97.35	98.67	98.67	99.12	99.12	99.12	99.12	99.12
	Hotel 2	∗∗	77.88	79.81	77.88	80.77	84.62	83.65	86.54	83.65	84.62	83.65
	Hotel 3	∗∗	83.33	85.19	81.48	85.19	88.89	87.04	85.19	83.33	84.19	85.19
	Study	∗∗	79.79	84.93	87.33	86.64	88.36	87.33	87.67	87.33	87.67	86.99
	MIT	∗∗	75.32	75.32	75.32	79.22	79.22	80.52	80.52	77.92	76.62	79.22
Test on train set		79.24	**81.94**	81.56	80.72	81.06	80.73	80.87	80.63	80.48	80.54	80.44

Table A5. Train and test statistics of running SGP on 3DMatch [92] with **training and test sets exchanged**, *i.e.*, we train SGP on the smaller test set $(1,623$ pairs), but test S-FCGF on the larger training set $(9,856$ pairs). SGP setting: retrain = False, verifyLabel = False. Statistics plotted in Fig. A1(b). We see SGP demonstrates overfitting while training on the smaller test set: S-FCGF achieves equally good (91.37%) recall on the test set, but only achieves below 82% recall on the training set (while in Tables A2-A4 S-FCGF has over 92% recall on the training set). Statistics plotted in Fig. A1(b).

performed on 3DMatch. However, even with RANSAC10K, the test recall drops to below 30%. Therefore, this ablation study shows the necessity of a robust teacher for SGP to work. Fortunately, we have plenty of robust solvers, as discussed in the main text. So we think this is a strength of SGP, rather than a weakness.