A. Appendix

In this supplementary material, we first provide rigorous definitions of evaluation metrics (Sec. A.1), then describe the data pre-processing step (Sec. A.2), network architectures (Sec. A.4) and training on individual datasets (Sec. A.3) in more detail. We further provide additional results (Sec. A.5), ablation studies (Sec. A.6) as well as (Sec. A.3) in more detail. We further provide additional architectures (Sec. A.4) and training on individual datasets.

A.1. Evaluation metrics

The evaluation metrics, which we use to assess model performance in Sec. 4 of the main paper and Sec. A.5 of this supplementary material, are formally defined as follows:

**Inlier ratio** looks at the set of putative correspondences \((p, q) \in K_{ij}\) found by reciprocal matching in feature space, and measures what fraction of them is "correct", in the sense that they lie within a threshold \(\tau_1 = 10\) cm after registering the two scans with the ground truth transformation \(\hat{T}_{ij}\):

\[
IR = \frac{1}{|K_{ij}|} \sum_{(p, q) \in K_{ij}} \left[ ||T_{ij}^p(p) - q||_2 < \tau_1 \right],
\]

with \([\cdot]\) the Iverson bracket.

**Feature Match recall (FMR)** [5] measures the fraction of point cloud pairs for which, based on the number of inlier correspondences, it is likely that accurate transformation parameters can be recovered with a robust estimator such as RANSAC. Note that FMR only checks whether the inlier ratio is above a threshold \(\tau_2 = 0.05\). It does not test if the transformation can actually be determined from those correspondences, which in practice is not always the case, since their geometric configuration may be (nearly) degenerate, e.g., they might lie very close together or along a straight edge. A single pair of point clouds counts as suitable for registration if

\[ IR > \tau_2 \]

Registration recall [2] is the most reliable metric, as it measures end-to-end performance on the actual task of point cloud registration. Specifically, it looks at the set of ground truth correspondences \(H^t_{ij}\) after applying the estimated transformation \(T_{ij}^Q\), computes their root mean square error,

\[
RMSE = \left( \frac{1}{|H^t_{ij}|} \sum_{(p, q) \in H^t_{ij}} ||T_{ij}^Q(p) - q||^2_2 \right)^{1/2},
\]

and checks for what fraction of all point pairs \(RMSE < 0.2\).

**Chamfer distance** measures the quality of registration on synthetic data. We follow [12] and use the modified Chamfer distance metric:

\[
\tilde{CD}(P, Q) = \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q_{raw}} ||T_{ij}^Q(p) - q||^2_2 + \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P_{raw}} ||q - T_{ij}^Q(p)||^2_2
\]

where \(P_{raw} \in \mathbb{R}^{2048 \times 3}\) and \(Q_{raw} \in \mathbb{R}^{2048 \times 3}\) are raw source and target point clouds, \(P \in \mathbb{R}^{717 \times 3}\) and \(Q \in \mathbb{R}^{717 \times 3}\) are input source and target point clouds.

**Relative translation and rotation errors** (RTE/RRE) measures the deviations from the ground truth pose as:

\[
\text{RTE} = ||t - \tilde{t}||_2 \\
\text{RRE} = \arccos \left( \frac{\text{trace}(R^t R) - 1}{2} \right)
\]

where \(R\) and \(t\) denote the estimated rotation matrix and translation vector, respectively.

The **Empirical Cumulative Distribution Function (ECDF)** measures the distribution of a set of values:

\[
\text{ECDF}(x) = \frac{|\{o_i < x\}|}{|O|}
\]

*First two authors contributed equally to this work.*
where $O$ is a set of values (overlap ratios in our case) and $x \in [\min\{O\}, \max\{O\}]$.

A.2. Dataset preprocessing

**3DMatch**: [13] is a collection of 62 scenes, combining earlier data from Analysis-by-Synthesis [10], 7Scenes [9], SUN3D [11], RGB-D Scenes v.2 [8], and Halber et al. [7]. The official benchmark splits the data into 54 scenes for training and 8 for testing. Individual scenes are not only captured in different indoor spaces (e.g., bedrooms, offices, living rooms, restrooms) but also with different depth sensors (e.g., Microsoft Kinect, Structure Sensor, Asus Xtion Pro Live, and Intel RealSense). 3DMatch provides great diversity and allows our model to generalize across different indoor spaces. Individual scenes of 3DMatch are split into point cloud fragments, which are generated by fusing 50 consecutive depth frames using TSDF volumetric fusion [4]. As a preprocessing step, we apply voxel-grid downsampling to all point clouds, and if multiple points fall into the same voxel, we randomly pick one.

**ModelNet40**: For each CAD model of ModelNet40, 2048 points are first generated by uniform sampling and scaled to fit into a unit sphere. Then we follow [12] to produce partial scans: for source partial point cloud, we uniformly sample a plane through the origin that splits the unit sphere into two half-spaces, shift that plane along its normal until $[2048 \cdot p_v]$ points are on one side, and discard the points on the other side; the target point cloud is generated in the same manner; then the two resulting, partial point clouds are randomly rotated, translated and jittered with Gaussian noise. For the rotation, we sample a random axis and a random angle $<45^\circ$. The translation is sampled in the range $[-0.5, 0.5]$. Gaussian noise is applied per coordinate with $\sigma = 0.05$. Finally, 717 points are randomly sampled from the $[2048 \cdot p_v]$ points.

**odometryKITTI**: The dataset was captured using a Velodyne HDL-64 3D laser scanner by driving around the mid-size city of Karlsruhe, in rural areas and on highways. The ground truth poses are provided by GPS/IMU system. We follow [1] to use ICP to reduce the noise in the ground truth poses.

A.3. Implementation and training

For 3DMatch/Modelnet/KITTI, we train PREDATOR using Stochastic Gradient Descent for 30/200/150 epochs, with initial learning rate 0.005/0.01/0.05, momentum 0.98, and weight decay $10^{-6}$. The learning rate is exponentially decayed by 0.05 after each epoch. Due to memory constraints we use batch size 1 in all experiments. The dataset-dependent hyper-parameters which include number of negative pairs in circle loss $n_p$, temperature factor $\gamma$, voxel size $V$, search radius for positive pair $r_p$, safe radius $r_s$, overlap and matchability radius $r_o$ and $r_m$ are given in Tab. 1. For more details please see our code.

A.4. Network architecture

The detailed network architecture of PREDATOR is depicted in Fig. 2. Our model is built on the KPConv implementation from the D3Feat repository. We complement each KPConv layer with instance normalisation Leaky ReLU activations. The $l$-th strided convolution is applied to a point cloud downsampled with voxel size $2^l \cdot V$. Upsampling in the decoder is performed by querying the associated feature of the closest point from the previous layer.

With $\approx$20k points after voxel-grid downsampling, the point clouds in 3DMatch are much denser than those of ModelNet40 with only 717 points. Moreover, they also have larger spatial extent with bounding boxes up to $3 \times 3 \times 3$ m$^3$, while ModelNet40 point clouds are normalised to fit into a unit sphere. To account for these large differences, we slightly adapt the encoder and decoder per dataset, but keep the same overlap attention model. Differences in network hyper-parameters are shown in Tab. 2.

A.5. Additional results

**Detailed registration results**: We report detailed per-scene Registration Recall (RR), Relative Rotation Error (RRE) and Relative Translation Error (RTE) in Tab. 3. RRE and RTE are only averaged over successfully registered pairs for each scene, such that the numbers are not dominated by gross errors from complete registration failures.

We get the highest RR and lowest or second lowest RTE and RRE for almost all scenes, this further shows that our overlap attention module together with probabilistic sampling supports not only robust, but also accurate registration.

**Feature match recall**: Finally, Fig. 1 shows that our descriptors are robust and perform well over a wide range of

\[
\begin{array}{cccccc}
\text{Dataset} & n_p & \gamma & V & r_p & r_s & r_o & r_m \\
3DMatch & 256 & 24 & 0.025 & 0.0375 & 0.1 & 0.0375 & 0.05 \\
ModelNet & 384 & 64 & 0.06 & 0.018 & 0.06 & 0.04 & 0.04 \\
odometryKITTI & 512 & 40 & 0.3 & 0.21 & 0.75 & 0.45 & 0.3 \\
\end{array}
\]

Table 1: Hyper-parameters configurations for different datasets.

\[
\begin{array}{cccccc}
\text{Dataset} & \# \text{strided} & \text{convolutions} & \text{radius} & \text{feature dim.} & \text{final feature dim.} \\
3DMatch (Predator) & 3 & 2.5 & 64 & 32 \\
3DMatch (bigPredator) & 3 & 2.5 & 128 & 32 \\
ModelNet & 2 & 2.75 & 256 & 96 \\
odometryKITTI & 3 & 4.25 & 128 & 32 \\
\end{array}
\]

Table 2: Different network configurations for 3DMatch, ModelNet and odometryKITTI datasets.
analyse the impact of each individual component. We first matchability scores attain the highest RR. Here we further probabilistic sampling guided by the product of the overlap and Ablations of matchability score

A.6. Additional ablation studies

Ablations of matchability score: We find that probabilistic sampling guided by the product of the overlap and matchability scores attains the highest RR. Here we further analyse the impact of each individual component. We first construct a baseline which applies random sampling (rand) over conditioned features, then we sample points with probability proportional to overlap scores (prob. (o)), to matchability scores (prob. (m)), and to the combination of the two scores (prob. (om)). As shown in Tab. 4, rand fares clearly worse, in all metrics. Compared to prob. (om), either prob. (o) or prob. (m) can achieve comparable results on 3DMatch; the performance gap becomes big on the more challenging 3DLoMatch dataset, where our prob. (om) is around 4 pp better in terms of RR.

Ablations of overlap attention module with FCGF: To demonstrate the flexibility of our model, we additionally add proposed overlap attention module to FCGF model. We

| matchability | overlap | 3DMatch | 3DLoMatch |
|--------------|---------|---------|-----------|
|              | FMR     | IR      | RR        | FMR     | IR      | RR        |
| ✓            | 96.0    | 43.6    | 83.9      | 69.3    | 15.7    | 39.3      |
| ✓            | 96.3    | 48.4    | 87.8      | 72.2    | 19.4    | 50.8      |
| ✓            | 96.1    | 46.2    | 88.0      | 71.3    | 16.9    | 49.3      |
| ✓ ✓           | 96.6    | 49.9    | 88.3      | 71.7    | 20.0    | 54.2      |

Table 4: Different combinations of scores used for probabilistic sampling.

thresholds for the allowable inlier distance and the minimum inlier ratio. Notably, REDATOR consistently outperforms D3Feat that uses a similar KPConv backbone.

Table 3: Detailed results on the 3DMatch and 3DLoMatch datasets.

| # Samples | 3DMatch | 3DLoMatch |
|-----------|---------|-----------|
| 5000      | 2500    | 1000      | 500 250  |
| 5000      | 2500    | 1000      | 500 250  |
| Registration Recall (%) | 3DMatch | 3DLoMatch |
| FCGF [3]   | 85.1 84.7 83.3 81.6 71.4 | 40.1 41.7 38.2 35.4 26.8 |
| FCGF+OA    | 88.4 87.7 87.8 87.0 84.3 | 53.4 55.8 57.5 56.4 50.9 |

| data loader | encoder | overlap attention | decoder | overall |
|-------------|---------|-------------------|---------|---------|
| FCGF [3]    | 6 414   | —                 | 25 445  |
| D3Feat [1]  | 200 11  | —                 | 63 274  |
| Ours        | 191 9   | 70                 | 1 271   |

Table 6: Runtime per fragment pair in milliseconds, averaged over 1623 test pairs of 3DMatch.
Figure 1: Feature matching recall in relation to inlier distance threshold $\tau_1$ (left) and inlier ratio threshold $\tau_2$ (right)

consuming step of our model, and also of D3Feat, is the data loader, as we have to pre-compute the neighborhood indices before the forward pass. With its smaller encoder and decoder, but the additional overlap attention module, PREDATOR is still marginally faster than D3Feat. FCGF has a more efficient data loader that relies on sparse convolution and queries neighbors during the forward pass. See Tab. 6.

A.8. Qualitative visualization

We show more qualitative results in Fig. 3 and Fig. 4 for 3DLoMatch and ModelLoNet respectively. The input point clouds are rotated and translated here for better visualization of overlap and matchability scores.

References

[1] 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 CVPR, 2020. 2, 3
[2] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In CVPR, 2015. 1
[3] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In ICCV, 2019. 3
[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In ACM SIGGRAPH, 1996. 2
[5] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3D point matching. In CVPR, 2018. 1
[6] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3D point cloud matching with smoothed densities. In CVPR, 2019. 3
[7] Maciej Halber and Thomas A. Funkhouser. Structured global registration of RGB-D scans in indoor environments. arXiv preprint arXiv:1607.08539, 2016. 2
[8] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised feature learning for 3D scene labeling. In ICRA, 2014. 2
[9] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In CVPR, 2013. 2
[10] Julien Valentin, Angela Dai, Matthias Nießner, Pushmeet Kohli, Philip Torr, Shahram Izadi, and Cem Keskin. Learning to navigate the energy landscape. In 3DV, 2016. 2
[11] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In ICCV, 2013. 2
[12] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust point matching using learned features. In CVPR, 2020. 1, 2
[13] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: learning local geometric descriptors from RGB-D reconstructions. In CVPR, 2017. 2
Figure 2: Network architecture of PREDATOR for 3DMatch (middle) and ModelNet (bottom). In the cross attention module, for each (query $s_i \in \mathbb{R}^{b \times 1}$, key $k_i \in \mathbb{R}^{b \times 1}$, value $v_i \in \mathbb{R}^{b \times 1}$), $\bigodot$ denotes first reshape them into shape $(4, \frac{b}{4})(4$ heads), then compute scores matrix $S$ from $s_i$ and $k_i$, finally get message update from $v_i$ and reshape back to $(b, 1)$. 

Source: [Original Paper]
Figure 3: Example results on 3DLoMatch.
Figure 4: Example results on \textit{ModelLoNet}. 

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example_results}
\caption{Example results on \textit{ModelLoNet}.}
\end{figure}