F2M-REG: UNSUPERVISED RGB-D REGISTRATION WITH FRAME-TO-MODEL OPTIMIZATION

Anonymous authors
Paper under double-blind review

A APPENDIX

In the appendix, we organize the details as follows. First, in Sec. B, we present the specifics of our implementations. Second, Sec. C introduces additional experiments that focus on the effectiveness of loss functions, comparision with supervised learning based on SLAM, and the convergence of our framework. Finally, more qualitative results are presented in Sec. D.

B IMPLEMENTATION DETAILS OF F2M-REG

B.1 NEURAL IMPLICIT FIELD OPTIMIZATION

For each sequence, we initially conduct 200 iterations of mapping using the first frame to establish the initialization of the neural implicit field. Subsequently, during the training phase, we input both the current frame and the neural implicit field $\mathcal{M}$. We maintain $\mathcal{M}$ fixed while optimizing the camera parameters of the untracked pose. Specifically, we randomly select $N_t = 1024$ pixels from the current frame. For each ray, we uniformly sample $M_c = 32$ points between the near and far bounds. Additionally, we sample an extra $M_f = 21$ depth-guided points evenly within the range $[d - d_s, d + d_s]$, where d represents the depth and $d_s = 0.25$ denotes a small offset. During experimentation, it was noted that employing the same number of optimization rounds as in Co-SLAM (Wang et al., 2023) often yielded suboptimal untracked poses across most scenes. This challenge arises due to the larger inter-frame distances present in our data compared to scenes encountered in previous Neural SLAM tasks. Hence, we conduct 100 iterations for tracking to mitigate this issue.

Subsequently, the tracked pose is utilized in the mapping stage. Upon the initial addition of each frame to the mapping stage, 5% of its pixels are incorporated into the maintained pixel bank. During the mapping phase, 2048 pixels are randomly selected from the pixel bank, and rays are generated to participate in the training process. The subsequent procedure mirrors that of the tracking section, except that the optimized parameters $\Psi = \{\theta, \hat{T}_{i-2}, \hat{T}_{i-1}, \tilde{T}_i\}$ consist of the neural implicit field $\mathcal{M}$ and the camera poses in the batch.

Our sub-scene representation comprises a $L = 16$ level hash grid $\mathcal{V}_\alpha = \mathcal{V}_{\alpha l=1}^{l\ L}$, with 16 bins oneblob for each dimension. The color and Signed Distance Function (SDF) are encoded by two 2-layer MLPs with 32 hidden units and a 15-dimensional geometric feature. The boundaries of our sub-scene are confined within the following ranges along the xyz-axes: (-3, 7), (-5, 5), (-4, 4). Regarding the learning rates, we utilize $\eta_t = 0.002$ for tracking, and $\eta_f = 0.01$, $\eta_d = 0.01$, and $\eta_p = 0.0005$ for the feature grid, decoder, and pose optimizer, respectively.

In both of the above stages, we minimize four different losses introduced in Co-SLAM(Wang et al., 2023). They include (1) two rendering losses $\mathcal{L}_{RGB}$ and $\mathcal{L}_{depth}$ for minimizing errors between ground truth RGB/depth image $\hat{C}_p$/$\hat{D}_p$ and rendered RGB/depth image C_p/D_p:

$$\mathcal{L}_{RGB} = \frac{1}{|P|} \sum_{p \in P} (C_p - \hat{C}_p)^2, \mathcal{L}_{depth} = \frac{1}{|P|} \sum_{p \in P} (D_p - \hat{D}_p)^2, \tag{1}$$

where P represents sampled image pixels. (2) an SDF loss $\mathcal{L}_{sdf}$ to enhance the consistency of the SDF field:

$$\mathcal{L}_{sdf} = \frac{1}{|P|} \sum_{p \in P} \frac{1}{|S_p^{tr}|} \sum_{s \in S_p^{tr}} (D_s - \hat{D}_s)^2, \tag{2}$$

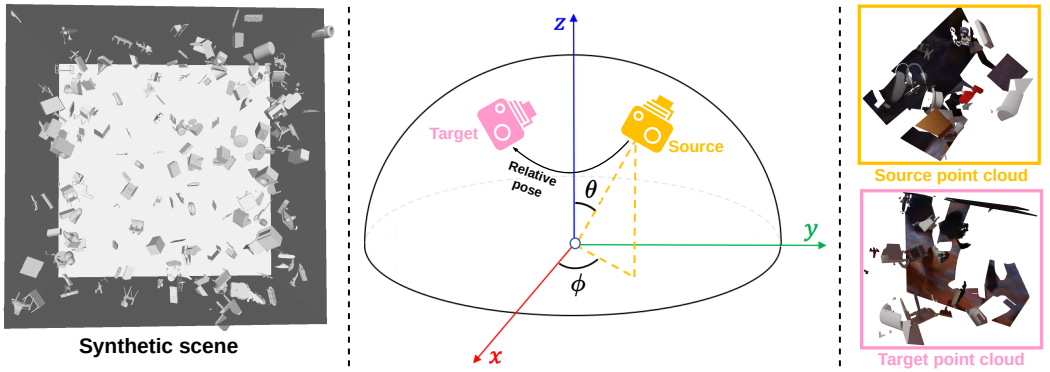

Figure 1: **Demonstration of Sim-RGBD dataset.** The entire scene is depicted in the left figure. Camera sampling is illustrated in the middle figure. Initially, we sample the position of the first camera based on a specified pitch angle θ and yaw angle ϕ, with (0, 0, 0) as the viewpoint, forming the camera's view direction. The position of the second camera is derived from the transformation of the first camera position, which is obtained from a Gaussian distribution. The right figure showcases the point cloud with color extracted from the scene.

S_p^{tr} represents whose signed distance function (SDF) is not truncated along the viewing ray of pixel p, and $D_s/\hat{D}_s$ denote their predicted/ground-truth SDF values. (4) For those sampled points distant from the observed surface, a free-space loss $\mathcal{L}_{fs}$ is applied to enforce their predicted SDF to be truncation distance d_{tr}:

$$\mathcal{L}_{fs} = \frac{1}{|P|} \sum_{p \in P} \frac{1}{|S_p^{fs}|} \sum_{s \in S_p^{fs}} (D_s - d_{tr})^2. \tag{3}$$

(5) An additional regularization on the interpolated features $\mathcal{V}_\alpha(x)$ in order to decrease the noisy in reconstruction.

$$\mathcal{L}_{smooth} = \frac{1}{\mathcal{V}} \sum_{x \in |\mathcal{V}|} (\Delta_x^2 + \Delta_y^2 + \Delta_z^2) \tag{4}$$

where $\mathcal{V}$ denotes the grid and $\Delta_{xyz} = \mathcal{V}_\alpha(x + \epsilon_{xyz}) - \mathcal{V}_\alpha(x)$ The weights of each loss are $\lambda_{RGD} = 5.0$, $\lambda_{depth} = 0.1$, $\lambda_{sdf} = 1000$, $\lambda_{fs} = 10$, $\lambda_{smooth} = 0.001$.

B.2 Registration Model Training

Supervisory signals for the feature extractor are generated when the current batch is fixed. These signals are generated by optimized pose $(\hat{T}_i, \hat{T}_j)$, corresponding point clouds (X_i, X_j), and corresponding features (F_i, F_j) within the current batch. Specifically, the process begins by deriving a relative pose using the global pose optimized within neural implicit representation. For one relative pose $\Delta T_{i-1,i}$, correspondences between two point clouds are identified using a specified threshold τ, which can be formulated as $\mathcal{C}^* = \{(p_{i-1}, p_i) \mid ||T_{i-1,i} p_{i-1} - p_i|| < \tau\}$. The correspondences and corresponding feature pairs $\{(F^{p_{i-1}}, F^{p_i}) \mid (p_{i-1}, p_i) \in \mathcal{C}^*\}$ are utilized to compute the loss.

B.3 Geometric Fitting

Given 512 input correspondences, $\mathcal{C} = \{(\mathbf{p}_i, \mathbf{q}_i) \mid \mathbf{p}_i \in \mathbf{X}, \mathbf{q}_i \in \mathbf{Y}\}$, we randomly sample $t = 10$ subsets, each containing $l = 20\%$ of the total correspondences. For each subset, a candidate transformation T is estimated by solving a Weighted Procrustes problem (Besl & McKay, 1992). The candidate transformation T^* that minimizes the error $E(C, T^*)$ is retained. Additionally, during the testing phase, we increase t to 100 and reduce l to 5% to achieve better RANSAC results while limiting computational costs.

B.4 SIM-RGBD

To clarify how we sample the appropriate camera poses, we visualize the process of sampling two camera poses in Fig. 1.

Table 1: Implementation details of our F2M-Reg

(a) Same setting as PointMBF

Momentum	0.9
Optimizer	Adam
Image size	128*128
Feature dimension	32
K_{v2g},K_{g2v} for training	$K_{v2g} = 16$,$K_{g2v} = 1$
K_{v2g},K_{g2v} for test	$K_{v2g} = 32$,$K_{g2v} = 1$

(b) Changes in F2M-Reg

Batch size	4
Normalization in ResNet	GroupNorm
Normalization in fusion	GroupNorm
Group of channels	32
low's ratio	False
Number of correspondence k	256

B.5 SETTINGS

We adopt PointMBF (Yuan et al., 2023) as our registration model and utilize several of its settings, such as data processing and learning rate. On the software side, our code is built using PyTorch and PyTorch3D (Ravi et al., 2020). On the hardware side, we train our network using an Nvidia GeForce RTX 3090Ti GPU with 24GB of memory, paired with an Intel® Core™ i9-12900K @ 3.9GHz × 16 and 32GB of RAM. To ensure a fair comparison, we adhere to the same training schemes as PointMBF, including data processing and other configurations. Table 1 provides further details on the similarities and differences between our approach and PointMBF. Specifically, Table 1a outlines the shared settings, while Table 1b highlights the differing configurations.

B.6 TIME EFFICIENCY

Table 2: Runtime analysis

	Time(ms)
Feature Extraction	79.87 ± 29.10
Correspondence Estimation	35.57 ± 10.36
Geometric Fitting	10.32 ± 9.20
Loss Computing	28.97 ± 10.38
Backward	202.34 ± 164.08
Tracking(Just for training)	20.10 ± 4.41
Mapping(Just for training)	25.06 ± 2.59

The time for our pipeline was reported in Table 2. Our method increases the training time, focusing on the tracking and mapping stage, i.e., we have to spend time on optimizing neural implicit field compared to rasterization of the point cloud. But considering the excellent performance of the frame-to-model set of optimization frameworks, we think these time overheads are meaningful.

Table 3: **Ablation on loss.** ***Corr*** denotes the correspondence loss. ***Circle*** denotes the circle loss. All the blank (1st row) means the registration model was only bootstrapped on the Sim-RGBD dataset without fintuning.

Corr	Circle	Train Set	Rotation(°)					Translation(cm)					Chamfer(mm)				
			Accuracy ↑			Error↓		Accuracy ↑			Error↓		Accuracy ↑			Error↓	
			5	10	45	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
			71.3	78.6	87.4	15.8	2.0	46.9	65.5	76.3	34.5	5.4	57.5	72.2	75.0	77.7	0.6
✓		ScanNet	71.0	77.7	86.7	18.8	2.0	48.7	66.0	75.8	34.7	5.2	58.1	72.2	75.0	83.8	0.6
	✓	ScanNet	77.1	84.3	**92.7**	**10.3**	**1.9**	49.1	70.2	81.6	**24.0**	5.1	61.1	77.3	80.4	**62.7**	**0.5**
✓	✓	ScanNet	**77.4**	**84.5**	92.5	15.5	**1.9**	**50.0**	**70.6**	**82.1**	30.1	**5.0**	**61.5**	**77.6**	**80.9**	73.8	**0.5**

C Additional Experiments

C.1 Effectiveness on loss

Our work incorporates the circle loss (Sun et al., 2020; Huang et al., 2021) and correspondence loss into the training process. We conduct a comprehensive Ablation study to elucidate the significance of these two losses within the entire pipeline.

The correspondence loss L_{corr} (1) is formalized in equation 5. In the context, $\mathcal{C} = \{(\mathbf{p}_i, \mathbf{q}_i) \mid \mathbf{p}_i \in \mathbf{X}, \mathbf{q}_i \in \mathbf{Y}\}$ denotes the correspondences selected based on the cosine similarity of the features of corresponding two points. We choose the top 256 pairs of correspondences and use the relative optimized pose $\Delta\hat{T} = [\hat{R} \mid \hat{t}]$ to calculate the loss. The weights w_i range from 0 to 1, and are derived from the cosine similarity values of the two point features.

$$L_{corr} = \sum_{(\mathbf{p}_i, \mathbf{q}_i) \in \mathcal{C}} w_i \| \hat{R}\mathbf{p}_i + \hat{t} - \mathbf{q}_i \| \tag{5}$$

To better supervise the point-wise descriptors, we also follow (Huang et al., 2021) and employ the circle loss. Considering the correspondence $\mathcal{C} = \{(\mathbf{p}_i, \mathbf{q}_i) \mid \mathbf{p}_i \in \mathbf{X}, \mathbf{q}_i \in \mathbf{Y}\}$ and the optimized pose $\hat{T}$. We compute, for each point in $\mathbf{X}$ the distance to all points in $\mathbf{Y}$ .Pairs of points with a distance less than r_p are treated as positive samples ϵ_{pos}, while those greater than r_s are treated as negative samples ϵ_{neg}. The circle loss from $\mathbf{X}$ is formalized in equation 6.

$$L^{\mathbf{X}}_{circle} = \frac{1}{n} \sum_{i=1}^{n} log[1 + \sum_{j \in \epsilon_{pos}} e^{\beta^j_{pos}(d^j_i - \Delta_{pos})} \cdot \sum_{k \in \epsilon_{neg}} e^{\beta^k_{neg}(\Delta_{neg} - d^k_i)}] \tag{6}$$

where n is the number of the points in $\mathbf{X}$, $d^j_i = \|f_{p_i} - f_{q_i}\|$ denotes the L2 distance of the corresponding point features and Δ_{pos}, Δ_{neg} are positive and negative margins. The weights $\beta^j_{pos} = \gamma(d^j_i - \Delta_{pos})$ and $\beta^k_{neg} = \gamma(\Delta_{neg} - d^k_i)$ are computed for each correspondence. The margin hyper-parameters are set to $\Delta_{pos} = 0.1$ and $\Delta_{neg} = 1.4$. For the circle loss $L^{\mathbf{Y}}_{circle}$ goes the same. The final circle loss $L_{circle} = (L^{\mathbf{X}}_{circle} + L^{\mathbf{Y}}_{circle})/2$.

The outcomes are presented in Table 3. It is evident from the results that both losses are instrumental in enhancing the performance of our registration framework during the finetuning phase. Regarding the nature of the losses, the circle loss facilitates the accurate recognition of correspondences by the registration model, whereas the correspondence loss aids in adjusting the weighting of identified correspondences. Furthermore, the experiment confirms that the concurrent utilization of these two losses contributes to further advancements in our registration model.

C.2 Comparison with supervised learning based on SLAM.

For sequential data, it is intuitive to obtain the pose of each frame quickly through reconstruction pipelines like SLAM. The pose reconstructed by SLAM can be used to supervise the training of a registration model. However, our approach can further enhance the performance of a registration model trained on SLAM-reconstructed poses. To demonstrate the effectiveness of our optimization framework, we designed this ablation experiment. We selected ROSEFusion (Zhang et al., 2021) as the SLAM algorithm for the experiment.

We randomly divide the 1,045 ScanNet training scenes into two groups: 300 scenes and 745 scenes. Using ROSEFusion, we perform the reconstruction on the 300 scenes, and the resulting poses were used to train a registration model from scratch. We then fine-tune the registration model using our frame-to-model optimization framework on the next 745 scenes. Finally, we test all the registation model on ScanNet testing scenes.

The results, as shown in Table 4, indicate that the performance of the registration model improves as the number of training scenes increases, demonstrating the optimization strength of our method. Notably, our bootstrapping module is flexible and can accommodate different datasets. When the quality of the synthetic dataset or the accuracy of the reconstruction from real-world scenes improves, these factors contribute to a better initialization of the registration model. Given a highly expressive model, our framework can leverage these improvements to provide a stronger initialization, further enhancing the registration model's performance.

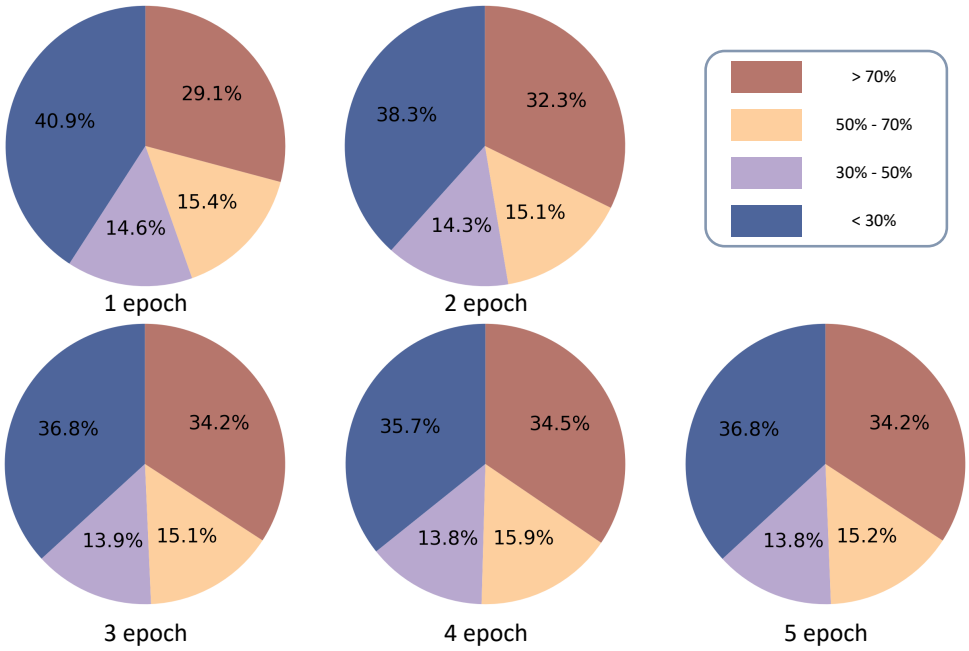

Figure 2: **Variation of Inlier Ratio Across Training Epochs.** The pie chart illustrates the distribution of inlier ratios across different ranges, with each range represented by distinct colored blocks. Larger blocks correspond to a higher number of frame pairs that fall within the respective inlier ratio range on the test dataset. This data was obtained by fine-tuning the registration model on the 3DMatch dataset Zeng et al. (2017) and testing it on the ScanNet test set Dai et al. (2017), using frame pairs spaced 50 frames apart.

C.3 CONVERGENCE

We conduct quantitative experiments to assess the convergence of our framework on the ScanNet (Dai et al., 2017), with the registration model fine-tuned on the 3DMatch (Zeng et al., 2017).

Figure 2 and table 5 depict the evolution on the ScanNet test set as the number of epochs progresses. We observe an initial increase in the inlier ratio until epoch 4, followed by oscillations. Again this phenomenon appears in the performance of the pose. These findings underscore the convergence of our framework. Our framework comprises two integral components: the registration model and the neural implicit field. They synergistically reinforce each other, wherein the enhanced performance of the registration model contributes to improved quality of the global pose input for neural field. Consequently, this enhances the reconstruction quality of neural field. The improved neural field quality facilitates more accurate and efficient optimization of the untracked poses, thereby providing a more effective gradient to refine the preceding registration model.

Table 4: **Comparison with supervised learning based on SLAM.** N_s indicates the number of scenes used for training. The first row shows the results of using the reconstructed poses from ROSEFusion on 300 scenes to supervise the registration model, which was then tested on ScanNet. The subsequent rows display the outcomes of increasing the number of training scenes from the first row and further optimizing the results using our pipeline.

N_s	Rotation(°)					Translation(cm)					Chamfer(mm)				
	Accuracy ↑			Error↓		Accuracy ↑			Error↓		Accuracy ↑			Error↓	
	5	10	45	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
300 (ROSEFusion)	70.5	79.6	90.4	12.5	2.3	42.9	64.5	77.4	28.7	5.9	55.0	72.2	75.7	73.1	0.7
+200 (ROSEFusion+Ours)	71.1	80.6	91.2	12.0	2.3	42.5	64.4	77.8	27.5	6.0	54.7	72.5	76.1	70.8	0.7
+400 (ROSEFusion+Ours)	71.6	80.6	90.9	11.4	2.2	43.3	64.7	77.7	27.0	5.9	55.2	72.7	76.2	69.2	0.7
+600 (ROSEFusion+Ours)	**73.2**	81.9	91.6	11.1	**2.1**	**45.8**	66.5	**79.7**	25.6	**5.5**	**57.4**	74.4	**78.1**	65.9	0.6
+745 (ROSEFusion+Ours)	**73.2**	**82.1**	**91.7**	**10.7**	**2.1**	45.3	**66.6**	79.6	**24.7**	**5.5**	57.1	**74.5**	78.0	**63.0**	**0.6**

Table 5: **Table of performance effects of training different epoch registation.** ***Epoch*** indicates the number of epochs that the registration model has been trained on 3DMatchZeng et al. (2017).

Epoch	Train Set	Rotation(°)					Translation(cm)					Chamfer(mm)				
		Accuracy ↑			Error↓		Accuracy ↑			Error↓		Accuracy ↑			Error↓	
		5	10	45	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
1	3DMatch	71.2	79.9	90.3	13.6	2.3	42.4	63.9	77.3	30.8	6.2	54.5	72.3	75.8	75.1	0.8
2	3DMatch	72.6	81.1	91.1	12.5	**2.2**	44.6	65.9	78.5	28.5	**5.8**	56.1	73.6	77.1	72.0	**0.7**
3	3DMatch	73.0	81.9	91.4	12.4	2.3	44.0	65.7	79.2	29.0	6.2	56.2	73.9	77.8	71.8	0.8
4	3DMatch	**73.5**	**82.2**	**91.6**	**11.9**	**2.2**	**44.7**	**66.4**	**79.7**	**26.8**	**5.8**	**56.9**	**74.5**	**78.2**	**69.0**	**0.7**
5	3DMatch	73.0	81.7	91.4	12.2	2.4	44.5	65.9	79.4	28.2	6.4	56.6	74.0	77.7	71.5	0.8

C.4 Ablation on Different Module Combinations

Table 6: **Ablation on different module combination.**

Warm up	F2F	F2M	Rotation(°)					Translation(cm)					Chamfer(mm)				
			Accuracy ↑			Error↓		Accuracy ↑			Error↓		Accuracy ↑			Error↓	
			5	10	45	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
	✓		60.4	68.2	79.9	19.2	2.3	40.0	54.3	66.9	38.1	6.0	48.9	61.5	65.8	85.8	0.7
✓			71.3	78.6	87.4	15.8	2.0	46.9	65.5	76.3	34.5	5.4	57.5	72.2	75.0	77.7	0.6
		✓	74.4	82.8	92.3	**10.8**	2.1	46.8	67.9	80.4	**25.4**	5.5	58.5	75.5	79.0	**67.1**	0.6
✓	✓		75.2	82.5	90.3	14.0	2.0	47.4	68.3	80.5	30.0	5.4	58.8	76.5	79.4	69.2	0.6
✓		✓	**77.4**	**84.5**	**92.5**	15.5	**1.9**	**50.0**	**70.6**	**82.1**	30.1	**5.0**	**61.5**	**77.6**	**80.9**	73.8	**0.5**

We have separately studied the improvements from each component in Tab.2, Tab.3, Tab.4 and Tab.5 in the main paper. Here, we reorganize the results in Tab 6 below for a better understanding. Warm up refers to the synthetic warm-up, F2M refers to frame-to-model optimization, and F2F refers to the frame-to-frame optimization. Applying the synthetic warm-up mechanism (line 2) and frame-to-model optimization (line 3) independently both result in significant improvements over the frame-to-frame baseline (line 1), highlighting the strong effectiveness of these designs. Moreover, combining both mechanisms further enhances performance, achieving an improvement of 6 percentage points over the baseline-only model and 3 percentage points over the frame-to-model-only model across most metrics.

C.5 Comparison on TUM RGB-D

We conduct a comparison with PointMBFYuan et al. (2023) and F2M-Reg on TUM RGB-D, which has more accurate and high-resolution RGB-D streams. The two registration models in Tab. 7 are trained on ScanNet and tested on TUM RGB-D with a 50 frames apart setting. We find that our method surpasses PointMBF by 9.4 percent point on Rotation Accuracy@5°, 11.5 percent point on Translation Accuracy@5cm, and 9.5 percent point on Chamfer Accuracy@1cm. These results have proven the strong generality of method to new datasets.

D Qualitative Visualization

In this section, we present more detailed visualization results in Fig 3 for both our method and PointMBF(Yuan et al., 2023). We visualize the inputs and the final alignment outcomes. In dataset selection, we deliberately choose scenes with minimal overlap and significant lighting variations. From the visualization results, our method exhibits several advantages. This observation further supports the superiority of the frame-to-model approach proposed in this paper over the frame-to-frame approach, such as PointMBF. Leveraging our neural implicit field constructed on RGB-D sequences, our approach excels in handling multi-view inconsistency. Consequently, the rerendering of neural implicit field can effectively leverage both photometric and geometric consistency to optimize the estimated pose, surpassing the capabilities of frame-to-frame methods.

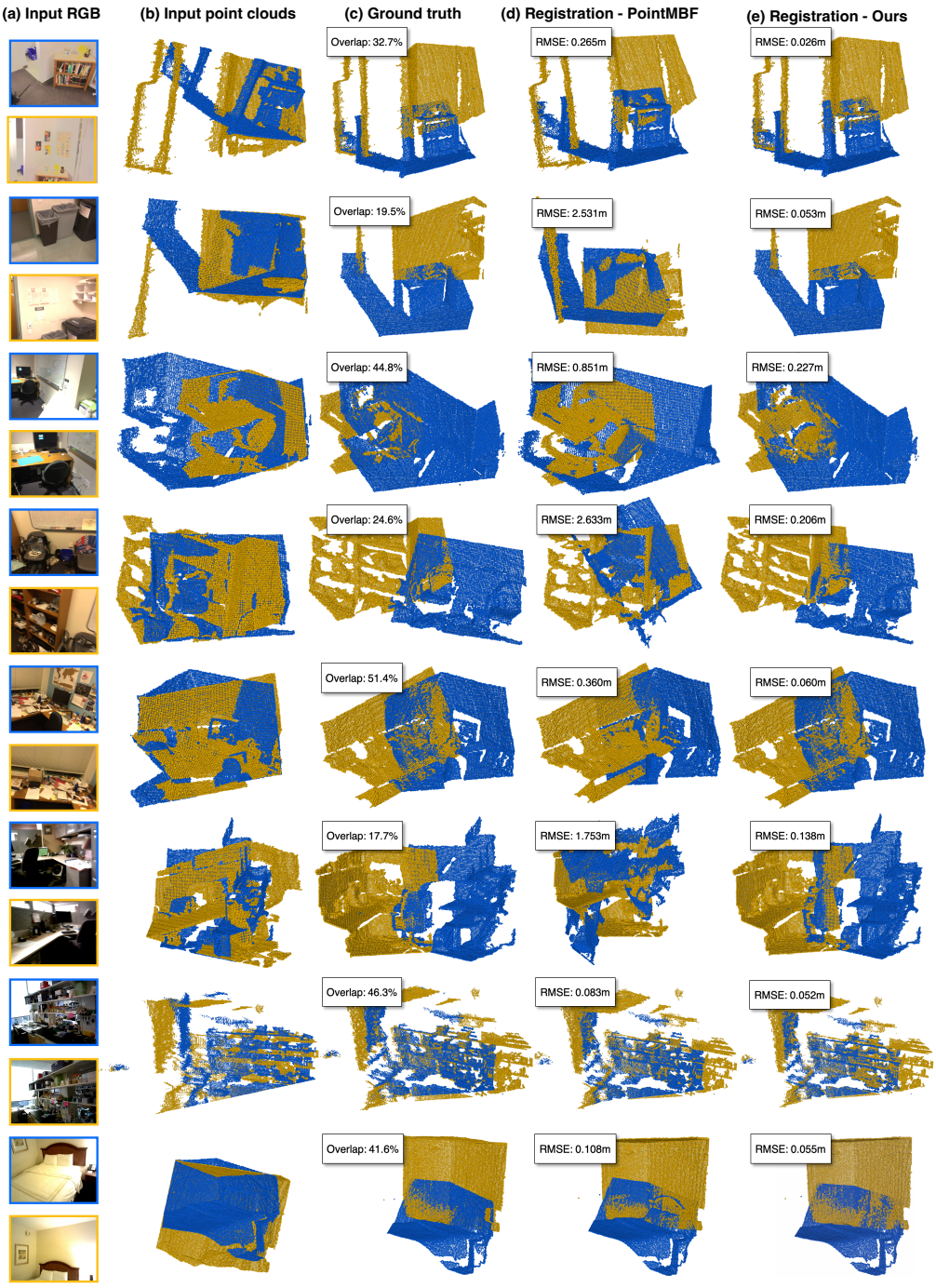

Figure 3: **Visual Comparison between PointMBF and F2M-Reg.** The registration model of PointMBF is trained on the ScanNet dataset Dai et al. (2017). Similarly, our registration model is fine-tuned on the ScanNet dataset, consistent with the experiments detailed in the paper.

Table 7: **Pairwise registration on TUM RGB-D with a 50 frames apart setting.**

	Rotation(°)					Translation(cm)					Chamfer(mm)				
	Accuracy ↑			Error↓		Accuracy ↑			Error↓		Accuracy ↑			Error↓	
	5	10	45	Mean	Med.	5	10	25	Mean	Med.	1	5	10	Mean	Med.
PointMBF	85.9	**97.9**	**100.0**	2.5	1.5	66.5	84.3	98.4	5.1	3.1	69.6	86.4	91.6	2.5	0.4
F2M-Reg	**95.3**	96.9	**100.0**	**1.8**	**1.1**	**78.0**	**94.8**	**99.5**	**3.7**	**2.6**	**79.1**	**95.3**	**96.9**	**1.1**	**0.3**

REFERENCES

Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pp. 586–606. Spie, 1992.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 4267–4276, 2021.

Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6398–6407, 2020.

Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13293–13302, 2023.

Mingzhi Yuan, Kexue Fu, Zhihao Li, Yucong Meng, and Manning Wang. Pointmbf: A multi-scale bidirectional fusion network for unsupervised rgb-d point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17694–17705, 2023.

Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1802–1811, 2017.

Jiazhao Zhang, Chenyang Zhu, Lintao Zheng, and Kai Xu. Rosefusion: random optimization for online dense reconstruction under fast camera motion. *ACM Transactions on Graphics (TOG)*, 40(4):1–17, 2021.