# IL-NeRF: Incremental Learning for Neural Radiance Fields with Camera Pose Alignment

## Supplementary Material

## 7. Finding Previous Optimal Camera Poses

To find previous $D$ optimal camera poses, we formulate a reward-collection optimization problem on a graph. In this graph, the nodes represent camera positions (i.e., the translations in the camera poses) with each node assigned a reward corresponding to the negative value of the preceding training loss. The edges represent Euclidean distances between each camera pair's positions. The goal is to find a path that collects as much reward as possible, subject to constraints on the total number of visited nodes and camera view coverage. Concretely, the objective optimization problem can be formulated as

$$\max \sum_{k=1}^{|\mathcal{P}^p|} x_k R_k \tag{15}$$

$$s.t. \sum_{k=1}^{|\mathcal{P}^p|} x_k = D \tag{16}$$

$$S(K) \geq S_{th}, K = \{k | x_k = 1, k = 1, \ldots, |\mathcal{P}^p|\} \tag{17}$$

$$E(x_k) \leq 1, \forall k \in \{1, \ldots, |\mathcal{P}^p|\} \tag{18}$$

where $x_k$ is the binary decision variable: $x_k = 1$ if node $k$ is visited otherwise $x_k = 0$. $S(K)$ is the shortest path that connects all the selected nodes. $E(x_k)$ is the number of incoming edge of each selected node. The first constraint (16) makes sure that only $D$ previous camera poses are selected. The second constraint (17) means that the view coverage of the selected cameras is larger than a threshold. This is because a large field of view coverage of the selected cameras improves the accuracy of the camera pose estimation. The third constraint (18) guarantees that every node only has one incoming edge. In other words, every node is visited at most once. Consequently, this reward-collection optimization problem can be viewed as a hybrid of the Knapsack Problem and the Travelling Salesperson Problem, which is an NP-hard problem.

**Related Work for Reward-Collection Optimization Problem.** The reward collection problem, also named orienteering problem, is an optimization issue that aims to determine the most efficient route for visiting multiple locations while maximizing the value or score of each place seen, all within a specified time frame and beginning and ending at a particular point [14]. This problem is widely utilized in the tourism sector [50], robot routing [38], food delivery [49] and transportation control [31]. As the orienteering problem belongs to the NP-hard class of problems, no algorithm can solve it optimally within a reasonable amount of time [17]. Different from the traditional orienteering problem, the optimization problem in this paper is more complex. Firstly, we do not limit the start and end points, and at the same time, we have a limitation on the number of accessible points, which makes it impossible to apply the existing proposed approaches to our method.

**Algorithm 1** Brute-Force for Selecting Cameras

```
Generate all possible D camera combinations 𝒦.
Initialize ℬ = ∅ and b = −∞.
for K ∈ 𝒦 do
    Use Breath-First-Search to find the shortest path S(K) that visits all the nodes in K.
    if S(K) ≥ S_th then
        Sum the total reward ℛ of K nodes.
        if ℛ ≥ b then
            ℬ = K
            b = ℛ
        end if
    end if
end for
Return ℬ
```

**Brute-Force Method.** The straightforward approach to address this problem is the Brute-Force method, demonstrated in Algorithm 1. This method involves: (1) Determining the shortest path for visiting all nodes for each $D$ camera combination. (2) Selecting the camera combination with the highest total rewards, while ensuring compliance with all constraints. However, the time complexity of this approach is $O((2^D \times D) \times \binom{N}{D})$, where $\binom{N}{D}$ represents the number of $D$-combinations derived from a given set of $N$ previous camera poses. $2^D \times D$ is the time complexity of finding the shortest path for each D camera combination. While this method guarantees an optimal solution, it becomes impractical for large numbers of nodes due to its time complexity.

**Proposed Greedy Algorithm.** Let $\mathbb{G}(V, E)$ be the graph of the previous camera poses, where $V$ denotes the set of cameras (nodes) and $E$ denotes the set of edges connecting each pair of two cameras. Let $e_{i,j} = e_{j,i}, e_{i,j} \in E, e_{j,i} \in E$ denote the edge between camera $i$ and $j$. Note that $\mathbb{G}(V, E)$ is an undirected weighted complete graph. The core concept of our greedy algorithm is to traverse all unvisited nodes starting from a specific node. During this process, the algorithm calculates the approximate edges between each pair

of the current node and its connected unvisited node, subsequently selecting the node with the maximum approximation edge as the next starting node. This process is repeated until a total of $D$ nodes have been selected. The complete description of the greedy algorithm is outlined in Algorithm 2.

**Step 1 (line 1 to line 2):** Introducing an auxiliary starting node $V_0$ into the graph, which establishes connections to all nodes with an edge length of 0. We define a set $\mathcal{B}$ to keep track of the visited nodes during traversal and initialize it as $\mathcal{B} = V_0$. Additionally, the current node index is initialized as $k = 0$.

**Step 2 (line 4 to line 10):** Identifying all the connected nodes of the current node $V_k$ that have not been visited yet (i.e., $V_i \notin \mathcal{B}$). Based on the reward $R_i$ and the edge length $e_{k,i}$ for each unvisited node, we compute the approximation edge length $\hat{e}_{k,i}$ and insert it into a temporary set $\hat{E}$. Specifically, $\hat{e}_{k,i} = R_i + \lambda(\frac{S_{th}}{D} - e_{k,i})$, where $\lambda$ is a parameter for adjusting the units of $R_i$ and $\frac{Sth}{D} - e_{k,i}$. This approximation edge is similar to the Lagrange multiplier [4] for handling the constraint (17).

**Step 3 (line 11 to line 12):** Selecting the node with the maximum $\hat{e}_{k,i}$ as the next visited node, updating the current index as $k = \arg\max \hat{E}$, and inserting the next visited node into the set of visited nodes $\mathcal{B}$.

**Step 4:** Repeating Step 2 and Step 3 until the greedy algorithm has visited a total of $D$ nodes.

**Algorithm 2** Proposed Greedy Algorithm

1: Add an auxiliary starting node $V_0$ linking all the nodes.
2: Initialize the visited set $\mathcal{B} = \{V_0\}$ and $k = 0$.
3: **while** $|\mathcal{B}| < D + 1$ **do**
4: $\quad \hat{E} = \{\}$
5: $\quad$ **for** $V_i \in V$ **do**
6: $\quad\quad$ **if** $V_i \notin \mathcal{B}$ **then**
7: $\quad\quad\quad \hat{e}_{k,i} = R_i + \lambda(\frac{S_{th}}{D} - e_{k,i})$
8: $\quad\quad\quad \hat{E}.append(\hat{e}_{k,i})$
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad k = \arg\max \hat{E}$
12: $\quad \mathcal{B}.append(V_k)$
13: **end while**
14: Return $\mathcal{B}.remove(V_0)$

The time complexity of our greedy algorithm is $O(D \times N \log N)$, which reduces computation time by several orders of magnitude.

**Comparison of Brute-Force Method with Ours.** Here, we show the performance comparison of Brute-Force Method and our greedy algorithm in terms of PSNR and computation time. As the results in Table 3 and 4, IL-NeRF can achieve comparable PSNR with Brute-Force method, however, the runtime of the Brute-Force method is several orders of magnitude larger than that of our proposed greedy algorithm, which makes the Brute-Force method impractical for incremental training scenarios.

Table 3. Performance comparison of Brute-Force method and our IL-NeRF on the Mip-NeRF360. IL-NeRF can achieve comparable PSNR with Brute-Force method, however, the runtime of the Brute-Force method is several orders of magnitude larger than our proposed greedy algorithm.

| Scene | Method | PSNR | running time |
|---|---|---|---|
| Bicycle | Brute-Force | 22.36 | 5 days 6 hours |
| | Greedy (Ours) | 22.34 | 10.92 ms |
| Bonsai | Brute-Force | 28.96 | 10 days 8 hours |
| | Greedy (Ours) | 28.96 | 25.57 ms |
| Counter | Brute-Force | 27.86 | 10 days 2 hours |
| | Greedy (Ours) | 27.82 | 23.78 ms |
| Garden | Brute-Force | 24.83 | 4 days 18 hours |
| | Greedy (Ours) | 24.82 | 9.87 ms |
| Kitchen | Brute-Force | 29.34 | 11 days 6 hours |
| | Greedy (Ours) | 29.34 | 28.39 ms |
| Room | Brute-Force | 31.49 | 12 days 10 hours |
| | Greedy (Ours) | 31.45 | 37.58 ms |
| Stump | Brute-Force | 24.91 | 4 days 12 hours |
| | Greedy (Ours) | 24.89 | 8.73 ms |

Table 4. Performance comparison of Brute-Force method and our IL-NeRF on the NeRF-real360 and Block-NeRF. IL-NeRF can achieve comparable PSNR with Brute-Force method, however, the runtime of the Brute-Force method is several orders of magnitude larger than our proposed greedy algorithm.

| Scene | Method | PSNR | running time |
|---|---|---|---|
| Pinecone | Brute-Force | 22.96 | 4 days 10 hours |
| | Greedy (Ours) | 22.93 | 9.58 ms |
| Vasedeck | Brute-Force | 26.24 | 5 days 2 hours |
| | Greedy (Ours) | 26.15 | 10.61 ms |
| Block-NeRF | Brute-Force | 26.63 | 20 days 5 hours |
| | Greedy (Ours) | 26.58 | 45.89 ms |

### 7.1. Implementation Details

The pipeline of IL-NeRF is summarized in Algorithm 3. We implement our framework following the architecture of Instant-NeRF [23, 34]. We use two separate Adam optimizers for NeRF and camera poses refinement respectively, with an initial learning rate of 0.01 for NeRF and an initial learning rate of 0.005 for pose refinement. The learning rate of the NeRF model decays every iteration by multiplying with 0.9954 (exponential decay), and the learning rate of the pose refinement decays every 100 iterations with a

**Algorithm 3** IL-NeRF Pseudo Code

Initialize $\mathcal{P}^p = \emptyset$.
**for** $t = 0$ **do**
    Estimate camera poses $\mathcal{P}_0^c$ from $G^0$
    Jointly train NeRF network $\Theta_0$ with camera poses $\mathcal{P}^p = \mathcal{P}^p \bigcup \mathcal{P}_0^c$
**end for**
**for** $t = 1$ to $t = T$ **do**
    Copy and freeze as $\Theta_{t-1}^*$
    Obtain past training data $C^p$ by $\mathcal{F}(\mathcal{P}^p, \Theta_{t-1}^*)$
    Align the camera pose $\mathcal{P}_t^c$ based on $\mathcal{P}^p$
    Jointly train NeRF network $\Theta_t$ with camera poses $\mathcal{P}^p = \mathcal{P}^p \bigcup \mathcal{P}_t^c$
**end for**

multiplier of 0.9. We train the network in each incremental step for Mip-NeRF360 with 30k iterations and $D = 10$, NeRF-real360 with 20k iterations and $D = 10$ and Block-NeRF with 40k iterations and $D = 50$.

## 8. More Comparisons for Results

In the main text, we only show PSNR, SSIM and LPIPS for some scenes of the datasets, and here we give the full results. Table 5 shows the results obtained by IL-NeRF and baseline methods on the Mip-NeRF360 dataset with seven real-world indoor and outdoor scenes. Additionally, Table 6 also shows the results obtained on the NeRF-real360 dataset with two real-world object-orientation scenes. From the results, we can see that IL-NeRF outperforms the original NeRF and achieves comparable results with CLNeRF.

Furthermore, we compare the original NeRF and IL-NeRF on two scenes, the 'Kitchen' and 'Counter' scenes in the Mip-NeRF360 dataset. Here, we give more visualization results of original NeRF and IL-NeRF on all scenes of three datasets.

Figure 10 to Figure 14 provide additional insight by presenting a qualitative comparison of the performance of the original NeRF and IL-NeRF. Specifically, we demonstrate the rendering results on the first task after each incremental training. It is evident that the original NeRF suffers from the catastrophic forgetting problem, resulting in images with significant distortions such as noise and blur, whereas IL-NeRF generates highly realistic images with quality comparable to the ground truth. This observation indicates that IL-NeRF is highly effective in mitigating the forgetting problem and addressing the coordinate shifting issue.

## 9. Ablation Study

In the main paper, we analyze the effectiveness of the camera coordinate alignment and the pose refinement that has been added to IL-NeRF on the scene 'Garden'. Here, we give more numerical results for the ablation study.

**Effect of Pose Selection.** Table 7 shows the performance comparison of our camera pose selection method with random selection and myopic selection on PSNR. IL-NeRF surpasses the other two methods, because it ensures the quality of rendered images used as references while providing a broader camera view coverage. Table 8 shows the PSNR of IL-NeRF across varying values of D on the 'Bicycle' scene from the Mip-NeRF360 dataset. As depicted, when D is a small value, the lack of adequate reference images results in the estimated camera poses of the incoming image data deviating from the original camera pose coordinate system, thereby leading to considerably poor rendering. As the value of D increases, the estimated camera poses of the incoming image data become increasingly precise, and thus the PSNR increases. Note that an excessively large value of D introduces poorly rendered cameras, subsequently leading to a decrease in PSNR. To identify the optimal D camera poses, we address a reward collection optimization problem on a graph.

**Effect of Transfer Matrices and Effect of Pose Refinement.** Table 9 to Table 10 show the performance comparison of the original NeRF, CLNeRF, IL-NeRF, IL-NeRF w/o TM, and IL-NeRF w/o PR on all three datasets. As we can see, IL-NeRF outperforms the original NeRF and achieves comparable results with CLNeRF. The results reveal a significant decline in performance on all test data without the transfer matrices (i.e., IL-NeRF w/o TM). This decline can be attributed to separate camera pose estimation for two tasks resulting in camera poses in two independent coordinate systems, which could mislead the model during training, leading to decreased performance. The results of IL-NeRF w/o PR, indicate that IL-NeRF with pose refinement outperforms IL-NeRF without it as the aligned camera poses by the transfer matrices may still contain noise and inaccuracies.

## 10. Camera Pose Alignment

Figure 15 shows the camera pose trajectories of GT and IL-NeRF. We treat the COLMAP [45] estimation from all training images as ground-truth (GT) camera poses. As we can see, IL-NeRF recovers accurate camera poses due to the help of incremental camera pose alignment. Table 11 shows the quantitative comparison of camera pose estimation accuracy of NeRF-SLAM, NoPe-NeRF and our IL-NeRF. Among our baselines, only NeRF-SLAM independently estimates camera poses, while the others need the camera poses estimated by COLMAP from all training images. Thus, we compare the camera pose estimation accuracy of IL-NeRF, NeRF-SLAM, and NoPe-NeRF. We use mean square error as the metric. IL-NeRF recovers more accurate camera poses with the help of the proposed camera pose alignment.

Table 5. Performance comparison on the Mip-NeRF360 dataset with the baselines: PSNR, SSIM, and LPIPS. IL-NeRF outperforms the original NeRF, EWC, NeRF-SLAM and achieves comparable results with CLNeRF.

| Scene | Method | Pose | **PSNR ⇑ / SSIM ⇑ / LPIPS ⇓** | | | |
|---|---|---|---|---|---|---|
| | | | $G^0$ | $G^1$ | $G^2$ | $G^3$ |
| Bicycle | NeRF | Yes | 22.76 / 0.61 / 0.33 | 18.58 / 0.47 / 0.46 | 20.03 / 0.52 / 0.43 | 20.03 / 0.52 / 0.44 |
| | EWC | Yes | 22.76 / 0.61 / 0.33 | 18.80 / 0.47 / 0.45 | 19.41 / 0.51 / 0.43 | 19.89 / 0.52 / 0.43 |
| | CLNeRF | Yes | 22.88 / 0.62 / 0.33 | **20.23 / 0.49 / 0.43** | 22.03 / 0.53 / **0.39** | 22.18 / 0.54 / 0.41 |
| | NeRF-SLAM | No | 22.78 / 0.61 / 0.33 | 19.67 / 0.48 / 0.45 | 21.37 / 0.53 / 0.41 | 21.61 / 0.53 / 0.42 |
| | IL-NeRF | No | **22.90 / 0.62 / 0.33** | 19.84 / 0.48 / 0.44 | **22.05 / 0.54** / 0.40 | **22.34 / 0.55 / 0.40** |
| Bonsai | NeRF | Yes | 33.30 / 0.93 / 0.07 | 25.47 / 0.75 / 0.25 | 23.53 / 0.66 / 0.34 | 22.12 / 0.68 / 0.35 |
| | EWC | Yes | 33.30 / 0.93 / 0.07 | 25.62 / 0.75 / 0.25 | 22.35 / 0.66 / 0.33 | 21.51 / 0.68 / 0.34 |
| | CLNeRF | Yes | 33.48 / 0.93 / 0.07 | 29.93 / 0.88 / 0.15 | 28.03 / 0.84 / 0.18 | 28.18 / 0.84 / 0.21 |
| | NeRF-SLAM | No | 33.32 / 0.93 / 0.07 | 29.13 / 0.84 / 0.21 | 28.01 / 0.79 / 0.29 | 26.85 / 0.80 / 0.29 |
| | IL-NeRF | No | **33.54 / 0.93 / 0.07** | **30.73 / 0.89 / 0.12** | **29.77 / 0.86 / 0.16** | **28.96 / 0.85 / 0.18** |
| Counter | NeRF | Yes | 32.12 / 0.91 / 0.07 | 24.62 / 0.72 / 0.25 | 21.94 / 0.65 / 0.34 | 20.30 / 0.62 / 0.37 |
| | EWC | Yes | 32.12 / 0.91 / 0.07 | 23.83 / 0.72 / 0.25 | 22.56 / 0.65 / 0.33 | 21.11 / 0.61 / 0.36 |
| | CLNeRF | Yes | **32.17 / 0.92** / 0.07 | 29.58 / 0.86 / 0.14 | 28.03 / 0.82 / 0.18 | **28.28 / 0.85** / 0.18 |
| | NeRF-SLAM | No | 31.75 / 0.91 / 0.07 | 28.30 / 0.83 / 0.21 | 26.84 / 0.79 / 0.28 | 25.30 / 0.77 / 0.31 |
| | IL-NeRF | No | 32.13 / 0.91 / **0.07** | **29.63 / 0.87 / 0.12** | **28.56 / 0.85 / 0.15** | 27.82 / 0.83 / **0.17** |
| Garden | NeRF | Yes | 24.70 / 0.71 / 0.20 | 22.34 / 0.64 / 0.25 | 20.17 / 0.59 / 0.31 | 19.42 / 0.54 / 0.38 |
| | EWC | Yes | 24.70 / 0.71 / 0.20 | 23.38 / 0.63 / 0.24 | 20.09 / 0.58 / 0.31 | 19.81 / 0.54 / 0.37 |
| | CLNeRF | Yes | 24.72 / 0.73 / 0.19 | **24.93 / 0.72 / 0.18** | 24.68 / 0.69 / 0.22 | 24.48 / 0.67 / **0.21** |
| | NeRF-SLAM | No | 24.72 / 0.71 / 0.20 | 24.03 / 0.69 / 0.23 | 23.50 / 0.65 / 0.28 | 23.37 / 0.61 / 0.33 |
| | IL-NeRF | No | **24.73 / 0.73 / 0.19** | 24.80 / 0.70 / 0.22 | **24.86 / 0.69 / 0.23** | **24.82 / 0.67** / 0.23 |
| Kitchen | NeRF | Yes | 31.17 / 0.91 / 0.08 | 27.01 / 0.75 / 0.25 | 21.42 / 0.70 / 0.31 | 23.69 / 0.75 / 0.24 |
| | EWC | Yes | 31.17 / 0.91 / 0.08 | 26.76 / 0.74 / 0.25 | 22.09 / 0.70 / 0.31 | 23.39 / 0.74 / 0.23 |
| | CLNeRF | Yes | 31.05 / 0.91 / 0.07 | 29.72 / 0.88 / 0.13 | 29.33 / 0.85 / 0.15 | 29.18 / 0.84 / 0.14 |
| | NeRF-SLAM | No | 30.87 / 0.90 / 0.09 | 29.63 / 0.85 / 0.20 | 27.65 / 0.81 / 0.24 | 27.71 / 0.82 / 0.20 |
| | IL-NeRF | No | **31.27 / 0.92 / 0.07** | **30.66 / 0.89 / 0.10** | **29.84 / 0.87 / 0.12** | **29.34 / 0.86 / 0.13** |
| Room | NeRF | Yes | 35.98 / 0.96 / 0.04 | 30.78 / 0.91 / 0.09 | 26.34 / 0.80 / 0.21 | 27.44 / 0.86 / 0.16 |
| | EWC | Yes | 35.98 / 0.96 / 0.04 | 31.84 / 0.90 / 0.09 | 27.38 / 0.79 / 0.20 | 28.08 / 0.86 / 0.16 |
| | CLNeRF | Yes | **36.18** / 0.96 / **0.03** | 33.93 / **0.95** / 0.05 | 32.03 / 0.92 / 0.08 | **31.99 / 0.93 / 0.06** |
| | NeRF-SLAM | No | 35.74 / 0.94 / 0.08 | 33.20 / 0.93 / 0.07 | 30.36 / 0.88 / 0.17 | 30.73 / 0.89 / 0.13 |
| | IL-NeRF | No | 36.04 / **0.96** / 0.04 | **34.02** / 0.94 / **0.04** | **32.35 / 0.92 / 0.07** | 31.45 / 0.91 / 0.09 |
| Stump | NeRF | Yes | 25.62 / 0.77 / 0.28 | 22.30 / 0.52 / 0.38 | 21.25 / 0.46 / 0.42 | 20.55 / 0.44 / 0.47 |
| | EWC | Yes | 25.62 / 0.77 / 0.28 | 22.55 / 0.51 / 0.37 | 21.09 / 0.45 / 0.42 | 21.48 / 0.44 / 0.46 |
| | CLNeRF | Yes | **26.18 / 0.79 / 0.27** | **25.93** / 0.64 / 0.37 | **25.12 / 0.62** / 0.38 | **25.18 / 0.64** / 0.39 |
| | NeRF-SLAM | No | 24.98 / 0.74 / 0.31 | 24.76 / 0.61 / 0.36 | 24.05 / 0.56 / 0.40 | 23.93 / 0.57 / 0.43 |
| | IL-NeRF | No | 25.96 / 0.77 / 0.28 | 25.75 / **0.66 / 0.32** | 25.09 / 0.60 / **0.37** | 24.89 / 0.58 / **0.37** |

**Combining CLNeRF with Our Camera Pose Alignment.** Table 12 shows the results of CLNeRF with provided camera poses (CLNeRF), CLNeRF with our camera pose alignment (CLNeRF-CPA). Comparison with CLNeRF and CLNeRF-CPA, our camera pose alignment is not only limited to IL-NeRF, but also can be combined with existing methods to solve more practical incremental learning scenarios.

**Combining Gaussian Splatting with Our Camera Pose Alignment.** Table 13 compares Gaussian Splatting with provided camera poses (GS) with Gaussian Splatting with our camera pose alignment (GS-CPA). GS-CPA employs the same camera pose selection and camera pose alignment, but for pose refinement, we refer to the pose refinement method outlined in [13]. As we can see, our camera pose alignment technique is effective not only for NeRF but also for Gaussian Splatting in incremental learning scenarios.

**Comparing with Existing Pose Optimization in NeRF.** We use NoPe-NeRF [5] as an example and replace our pose

Table 6. Performance comparison on the NeRF-real360 dataset with the baselines: PSNR, SSIM, and LPIPS. IL-NeRF outperforms the original NeRF, EWC, NeRF-SLAM and achieves comparable results with CLNeRF.

| Scene | Method | Pose | **PSNR ⇑ / SSIM ⇑ / LPIPS ⇓** | | | |
|---|---|---|---|---|---|---|
| | | | $G^0$ | $G^1$ | $G^2$ | $G^3$ |
| Pinecone | NeRF | Yes | 26.22 / 0.84 / 0.16 | 22.90 / 0.64 / 0.24 | 21.15 / 0.58 / 0.33 | 18.94 / 0.49 / 0.41 |
| | EWC | Yes | 26.22 / 0.84 / 0.16 | 22.70 / 0.63 / 0.24 | 21.42 / 0.57 / 0.32 | 18.81 / 0.48 / 0.41 |
| | CLNeRF | Yes | **26.88** / **0.89** / 0.12 | 24.23 / **0.79** / **0.16** | **24.03** / 0.73 / **0.19** | **23.18** / **0.74** / **0.21** |
| | NeRF-SLAM | No | 25.63 / 0.81 / 0.18 | 24.09 / 0.73 / 0.22 | 23.01 / 0.68 / 0.29 | 21.79 / 0.65 / 0.34 |
| | IL-NeRF | No | 26.31 / 0.87 / **0.10** | **24.56** / 0.78 / 0.17 | 23.78 / **0.74** / 0.20 | 22.93 / 0.72 / 0.23 |
| Vasedeck | NeRF | Yes | 29.03 / 0.85 / 0.07 | 23.99 / 0.70 / 0.26 | 22.73 / 0.69 / 0.24 | 21.57 / 0.64 / 0.31 |
| | EWC | Yes | 29.03 / 0.85 / 0.07 | 24.36 / 0.69 / 0.25 | 22.25 / 0.68 / 0.24 | 20.52 / 0.64 / 0.30 |
| | CLNeRF | Yes | 29.27 / 0.86 / 0.07 | **27.93** / **0.85** / 0.12 | 26.03 / 0.74 / 0.16 | **26.18** / 0.74 / 0.18 |
| | NeRF-SLAM | No | 27.98 / 0.79 / 0.11 | 26.41 / 0.77 / 0.21 | 25.10 / 0.72 / 0.21 | 24.62 / 0.71 / 0.26 |
| | IL-NeRF | No | **29.48** / **0.86** / **0.07** | 27.38 / 0.82 / **0.10** | **26.11** / **0.76** / **0.14** | 26.15 / **0.75** / **0.17** |


Figure 10. Qualitative comparison of the original NeRF and IL-NeRF on the rendering images in the first image data after each incremental training. GT means the ground truth of the training image. The original NeRF demonstrates severe catastrophic forgetting, leading to the loss of early-task scene information. In contrast, IL-NeRF is able to preserve the scene of interest throughout the training process. Testsets are the scenes 'Counter' and 'Bonsai' in the Mip-NeRF36 dataset.

camera alignment with NoPe-NeRF. We keep camera poses of previous data estimated by NoPe-NeRF, and when new incremental data arrives, we use the previous camera poses as the initial values, so that we can ensure that the camera poses of the new incremental data can be consistent with the coordinate system of the previous camera poses. Table 14 shows PSNR of NoPe-NeRF and IL-NeRF. IL-NeRF outperforms NoPe-NeRF. This is because NoPe-NeRF needs the previous data rendered from NeRF for depth estimation. Some poor quality replay data brings poor depth estimation results, which affects the performance of NoPe-NeRF.

## 11. Limitation

For large-scale scenes with limited overlap between views in the training dataset, the performance of IL-NeRF may be suboptimal because the limited overlap between views can result in significant errors or even the inability to calculate the transfer matrices during the camera coordinate alignment.

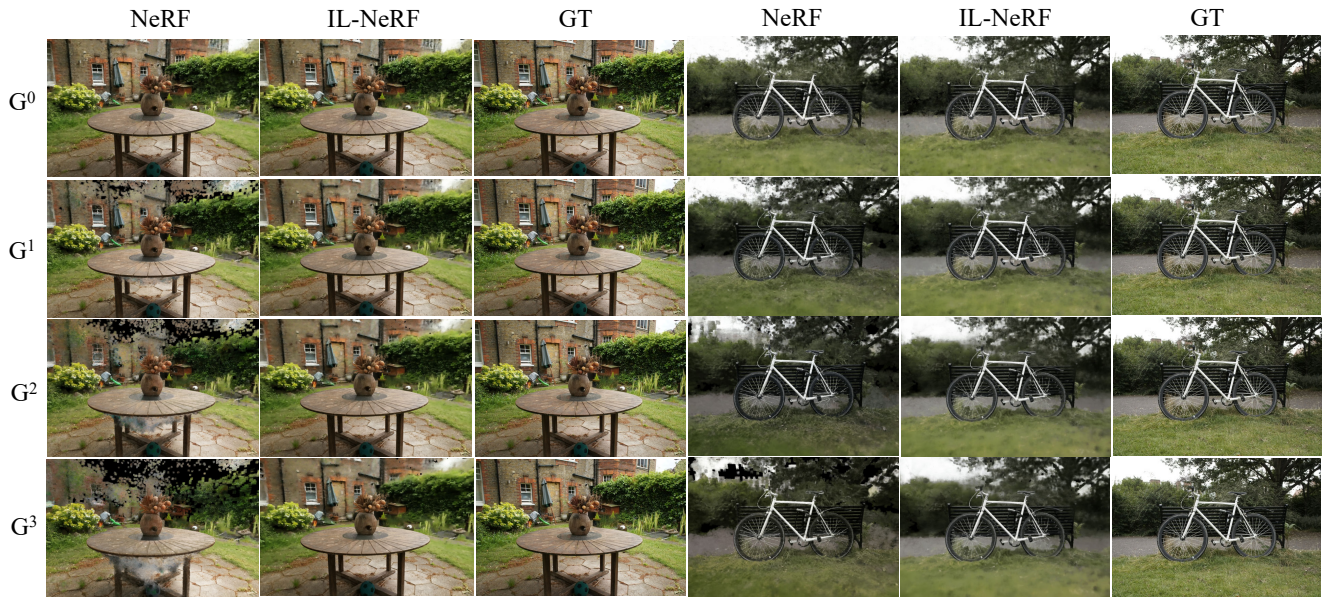



Figure 11. Qualitative comparison of the original NeRF and IL-NeRF on the rendering images in the first image data after each incremental training. GT means the ground truth of the training image. The original NeRF demonstrates severe catastrophic forgetting, leading to the loss of early-task scene information. In contrast, IL-NeRF is able to preserve the scene of interest throughout the training process. Testsets are the scenes 'Garden' and 'Bicycle' in the Mip-NeRF36 dataset.

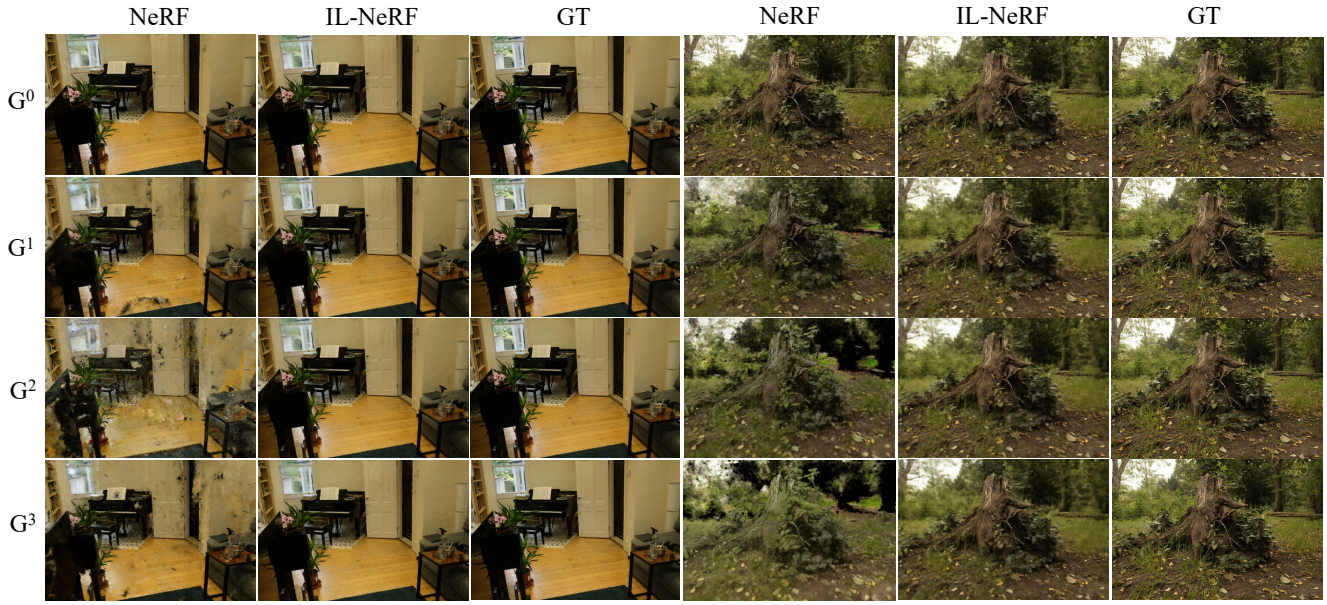



Figure 12. Qualitative comparison of the original NeRF and IL-NeRF on the rendering images in the first image data after each incremental training. GT means the ground truth of the training image. The original NeRF demonstrates severe catastrophic forgetting, leading to the loss of early-task scene information. In contrast, IL-NeRF is able to preserve the scene of interest throughout the training process. Testsets are the scenes 'Room' and 'Stump' in the Mip-NeRF36 dataset.

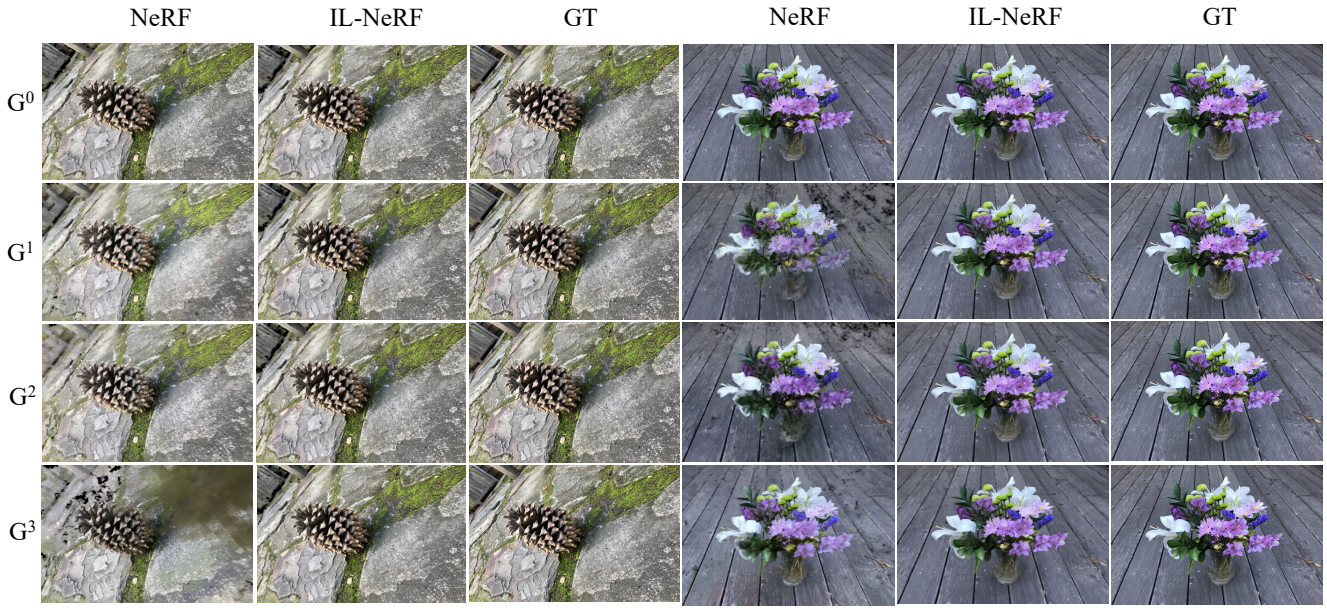



Figure 13. Qualitative comparison of the original NeRF and IL-NeRF on the rendering images in the first image data after each incremental training. GT means the ground truth of the training image. The original NeRF demonstrates severe catastrophic forgetting, leading to the loss of early-task scene information. In contrast, IL-NeRF is able to preserve the scene of interest throughout the training process. Testset is the scenes 'Pinecone' and 'Vasedesk' in the NeRF-real360 dataset.

Table 7. Comparison of camera pose selection method with random selection and myopic selection on PSNR. The higher the better.

| Method | **Mip-NeRF360** | | | | | | | **NeRF-real360** | | **Block-NeRF** |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bicycle | Bonsai | Counter | Garden | Kitchen | Room | Stump | Pinecone | Vasedeck | |
| Random | 17.76 | 23.41 | 23.41 | 20.49 | 23.12 | 23.82 | 19.98 | 18.23 | 20.21 | 19,38 |
| Myoptic | 21.06 | 28.15 | 27.06 | 24.28 | 28.44 | 29.96 | 24.29 | 22.02 | 25.51 | 24.86 |
| IL-NeRF | **22.34** | **28.96** | **27.82** | **24.82** | **29.34** | **31.45** | **24.89** | **22.93** | **26.15** | **26.58** |

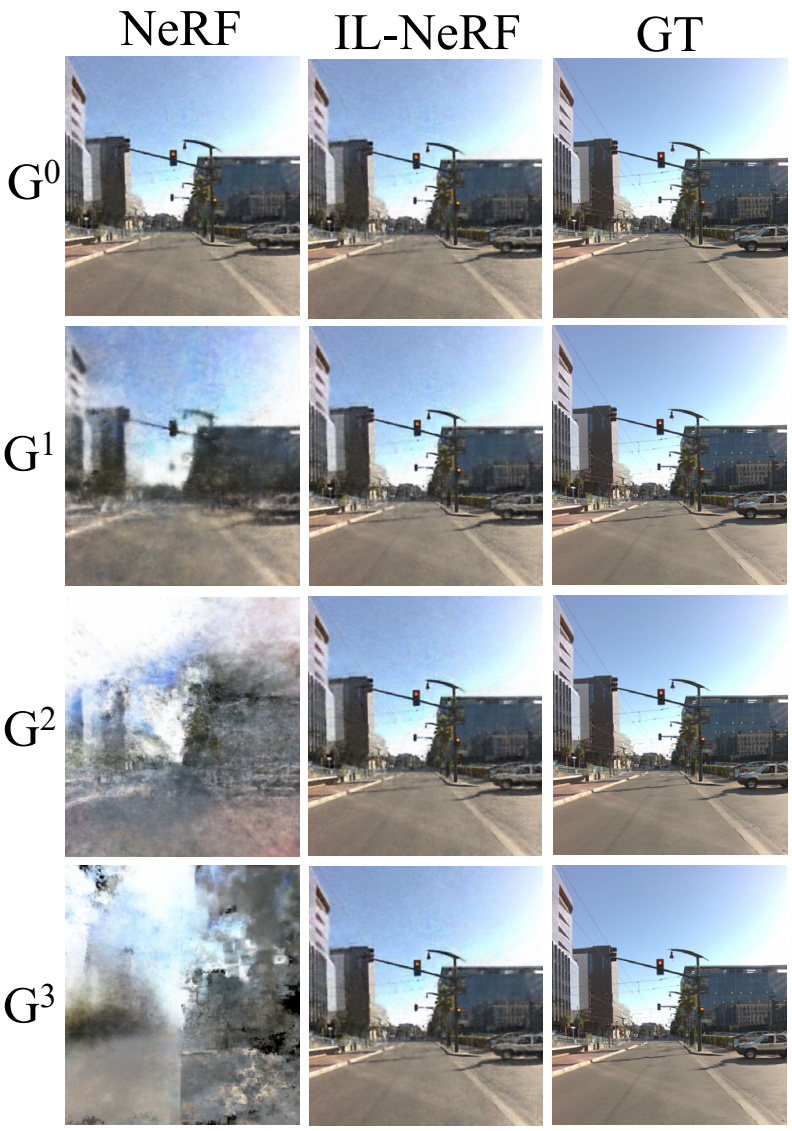



Figure 14. Qualitative comparison of the original NeRF and IL-NeRF on the rendering images in the first image data after each incremental training. GT means the ground truth of the training image. The original NeRF demonstrates severe catastrophic forgetting, leading to the loss of early-task scene information. In contrast, IL-NeRF is able to preserve the scene of interest throughout the training process. Testset is the scenes in the Block-NeRF dataset.

Table 8. Influence of optimal pose count D on IL-NeRF.

| Scene | Method | **PSNR** ⇑ | | | | |
|---|---|---|---|---|---|---|
| | | $D = 1$ | $D = 5$ | $D = 10$ | $D = 20$ | $D = All$ |
| Bicycle | Random | 15.35 | 16.27 | 17.76 | 18.41 | 18.69 |
| | Myoptic | 19.14 | 20.82 | 21.06 | 20.64 | 19.96 |
| | IL-NeRF | **19.14** | **21.13** | **22.34** | **21.85** | **20.57** |
| Bonsai | Random | 19.89 | 21.09 | 23.02 | 23.86 | 24.22 |
| | Myoptic | 24.81 | 26.98 | 27.30 | 26.75 | 25.87 |
| | IL-NeRF | **24.81** | **27.39** | **28.96** | **28.32** | **26.66** |
| Counter | Random | 19.11 | 20.26 | 22.11 | 22.92 | 23.27 |
| | Myoptic | 23.83 | 25.92 | 26.22 | 25.70 | 24.85 |
| | IL-NeRF | **23.83** | **26.31** | **27.82** | **27.20** | **25.61** |
| Garden | Random | 17.07 | 18.09 | 19.75 | 20.47 | 20.78 |
| | Myoptic | 21.29 | 22.75 | 23.42 | 22.95 | 22.20 |
| | IL-NeRF | **21.29** | **23.15** | **24.82** | **24.30** | **22.88** |
| Kitchen | Random | 20.15 | 21.36 | 23.32 | 24.17 | 24.54 |
| | Myoptic | 25.13 | 27.34 | 27.65 | 27.10 | 26.21 |
| | IL-NeRF | **25.13** | **27.75** | **29.34** | **28.69** | **27.01** |
| Room | Random | 21.60 | 22.90 | 25.00 | 25.91 | 26.31 |
| | Myoptic | 26.94 | 29.31 | 29.64 | 29.05 | 28.09 |
| | IL-NeRF | **26.94** | **29.74** | **31.45** | **30.76** | **28.95** |
| Stump | Random | 17.10 | 18.12 | 19.78 | 20.51 | 20.82 |
| | Myoptic | 21.32 | 23.19 | 23.46 | 22.91 | 22.23 |
| | IL-NeRF | **21.32** | **23.54** | **24.89** | **24.34** | **22.91** |
| Pinecone | Random | 15.75 | 16.69 | 18.22 | 18.89 | 19.18 |
| | Myoptic | 19.64 | 21.36 | 21.61 | 21.18 | 20.48 |
| | IL-NeRF | **19.64** | **21.68** | **22.93** | **22.42** | **21.11** |
| Vasedeck | Random | 17.96 | 19.04 | 20.78 | 21.54 | 21.87 |
| | Myoptic | 22.40 | 24.37 | 24.65 | 24.16 | 23.36 |
| | IL-NeRF | **22.40** | **24.73** | **26.15** | **25.57** | **24.07** |

Table 9. Comparison of IL- NeRF w/o TM, IL-NeRF w/o PR and IL-NeRF. IL-NeRF outperforms these two cases.

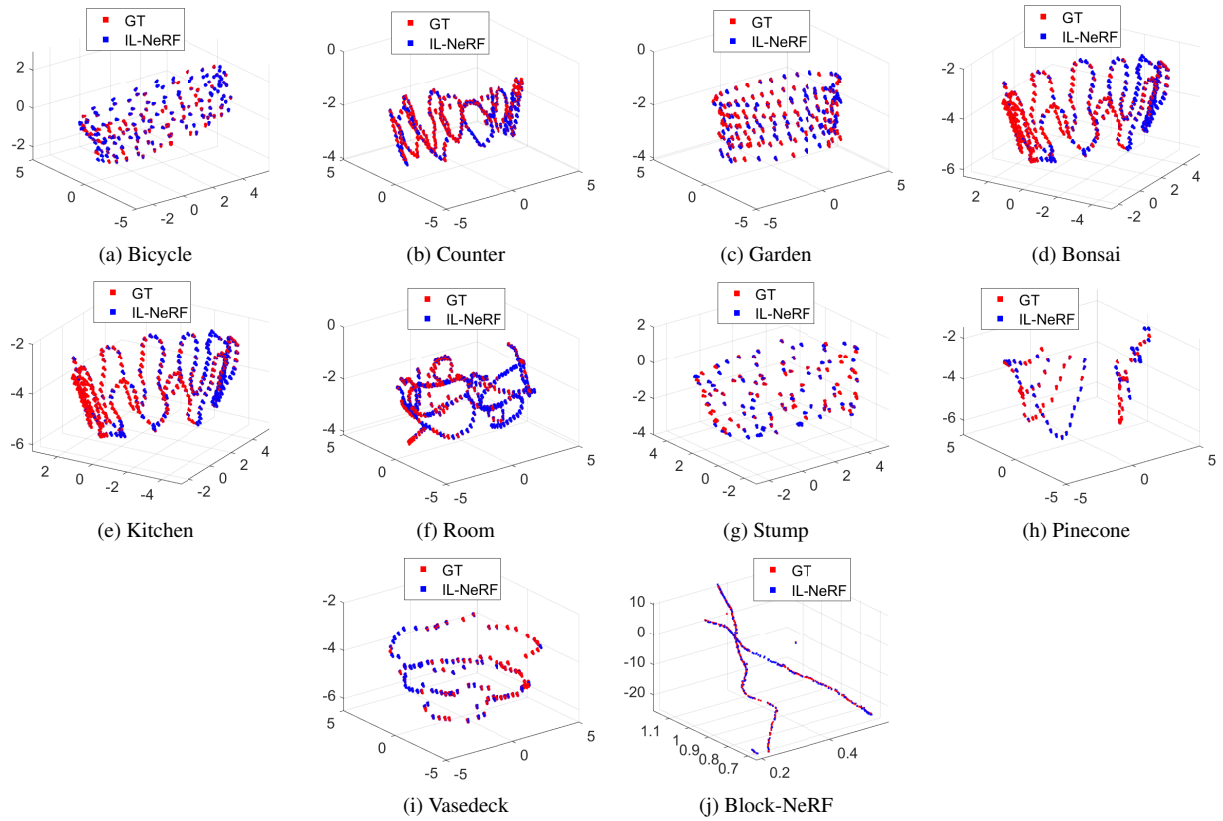
| Scene | Method | **PSNR ⇑ / SSIM ⇑ / LPIPS ⇓** | | | |
|---|---|---|---|---|---|
| | | $G^0$ | $G^1$ | $G^2$ | $G^3$ |
| Bicycle | w/o TM | 22.90 / 0.62 / 0.33 | 13.64 / 0.32 / 0.66 | 11.86 / 0.29 / 0.79 | 11.14 / 0.26 / 0.83 |
| | w/o PR | 22.76 / 0.61 / 0.33 | 18.67 / 0.41 / 0.52 | 20.74 / 0.46 / 0.50 | 21.06 / 0.46 / 0.52 |
| | IL-NeRF | **22.90 / 0.62 / 0.33** | **19.84 / 0.48 / 0.44** | **22.05 / 0.54 / 0.40** | **22.34 / 0.55 / 0.40** |
| Bonsai | w/o TM | 33.54 / 0.93 / 0.07 | 21.13 / 0.59 / 0.18 | 19.48 / 0.46 / 0.31 | 18.40 / 0.40 / 0.37 |
| | w/o PR | 33.30 / 0.93 / 0.07 | 28.30 / 0.84 / 0.18 | 27.80 / 0.81 / 0.21 | 27.33 / 0.80 / 0.22 |
| | IL-NeRF | **33.54 / 0.93 / 0.07** | **30.73 / 0.89 / 0.12** | **29.77 / 0.86 / 0.16** | **28.96 / 0.85 / 0.18** |
| Counter | w/o TM | 32.13 / 0.91 / 0.07 | 23.91 / 0.58 / 0.18 | 19.02 / 0.45 / 0.29 | 13.87 / 0.39 / 0.35 |
| | w/o PR | 32.12 / 0.91 / 0.07 | 27.89 / 0.83 / 0.13 | 27.05 / 0.81 / 0.16 | 26.47 / 0.79 / 0.17 |
| | IL-NeRF | **32.13 / 0.91 / 0.07** | **29.63 / 0.87 / 0.12** | **28.56 / 0.85 / 0.15** | **27.82 / 0.83 / 0.17** |
| Garden | w/o TM | 24.73 / 0.73 / 0.19 | 17.05 / 0.46 / 0.33 | 13.37 / 0.37 / 0.45 | 15.76 / 0.31 / 0.47 |
| | w/o PR | 24.70 / 0.71 / 0.20 | 23.34 / 0.67 / 0.20 | 23.17 / 0.69 / 0.21 | 22.42 / 0.67 / 0.23 |
| | IL-NeRF | **24.73 / 0.73 / 0.19** | **24.80 / 0.70 / 0.22** | **24.86 / 0.69 / 0.23** | **24.82 / 0.67 / 0.23** |
| Kitchen | w/o TM | 31.27 / 0.92 / 0.07 | 21.08 / 0.59 / 0.15 | 16.05 / 0.46 / 0.23 | 14.63 / 0.40 / 0.26 |
| | w/o PR | 31.17 / 0.91 / 0.08 | 28.54 / 0.84 / 0.12 | 27.86 / 0.82 / 0.15 | 27.48 / 0.78 / 0.18 |
| | IL-NeRF | **31.27 / 0.92 / 0.07** | **30.66 / 0.89 / 0.10** | **29.84 / 0.87 / 0.12** | **29.34 / 0.86 / 0.13** |
| Room | w/o TM | 36.04 / 0.96 / 0.04 | 27.45 / 0.62 / 0.06 | 17.40 / 0.49 / 0.13 | 19.98 / 0.43 / 0.18 |
| | w/o PR | 35.98 / 0.96 / 0.04 | 32.67 / 0.92 / 0.07 | 31.12 / 0.86 / 0.17 | 30.35 / 0.84 / 0.13 |
| | IL-NeRF | **36.04 / 0.96 / 0.04** | **34.02 / 0.94 / 0.04** | **32.35 / 0.92 / 0.07** | **31.45 / 0.91 / 0.09** |
| Stump | w/o TM | 25.96 / 0.77 / 0.28 | 20.78 / 0.44 / 0.48 | 16.71 / 0.32 / 0.73 | 15.81 / 0.27 / 0.80 |
| | w/o PR | 25.62 / 0.77 / 0.28 | 24.25 / 0.58 / 0.37 | 23.77 / 0.53 / 0.43 | 22.43 / 0.50 / 0.46 |
| | IL-NeRF | **25.96 / 0.77 / 0.28** | **25.75 / 0.66 / 0.32** | **25.09 / 0.60 / 0.37** | **24.89 / 0.58 / 0.39** |

Table 10. Comparison of IL- NeRF w/o TM, IL-NeRF w/o PR and IL-NeRF. IL-NeRF outperforms these two cases.

| Scene | Method | **PSNR ⇑ / SSIM ⇑ / LPIPS ⇓** | | | |
|---|---|---|---|---|---|
| | | $G^0$ | $G^1$ | $G^2$ | $G^3$ |
| Pinecone | w/o TM | 26.31 / 0.87 / 0.10 | 19.82 / 0.52 / 0.25 | 15.84 / 0.39 / 0.39 | 14.56 / 0.34 / 0.47 |
| | w/o PR | 26.22 / 0.84 / 0.16 | 23.87 / 0.61 / 0.22 | 22.24 / 0.66 / 0.28 | 21.13 / 0.66 / 0.32 |
| | IL-NeRF | **26.31 / 0.87 / 0.10** | **24.56 / 0.78 / 0.17** | **23.78 / 0.74 / 0.20** | **22.93 / 0.72 / 0.23** |
| Vasedeck | w/o TM | 29.48 / 0.86 / 0.07 | 22.09 / 0.54 / 0.25 | 16.05 / 0.40 / 0.35 | 13.04 / 0.35 / 0.37 |
| | w/o PR | 29.03 / 0.85 / 0.07 | 25.30 / 0.74 / 0.18 | 24.80 / 0.68 / 0.21 | 24.33 / 0.63 / 0.23 |
| | IL-NeRF | **29.48 / 0.86 / 0.07** | **27.38 / 0.82 / 0.10** | **26.11 / 0.76 / 0.14** | **26.15 / 0.75 / 0.17** |

Table 11. Quantitative comparison of camera pose estimation accuracy (Metric: Mean Square Error). The lower the better.

| Method | **Mip-NeRF360** | | | | | | | **NeRF-real360** | | **Block-NeRF** |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bicycle | Bonsai | Counter | Garden | Kitchen | Room | Stump | Pinecone | Vasedeck | |
| NeRF-SLAM | 0.857 | 0.469 | 0.556 | 0.749 | 0.282 | 0.267 | 0.774 | 1.146 | 0.537 | 1.674 |
| NoPe-NeRF | 1.853 | 0.967 | 1.191 | 1.664 | 0.631 | 0.524 | 1.649 | 2.248 | 1.087 | 2.983 |
| IL-NeRF | **0.752** | **0.387** | **0.499** | **0.668** | **0.257** | **0.212** | **0.693** | **0.971** | **0.468** | **0.972** |




(a) Bicycle (b) Counter (c) Garden (d) Bonsai

(e) Kitchen (f) Room (g) Stump (h) Pinecone

(i) Vasedeck (j) Block-NeRF

Figure 15. Camera pose estimation comparison. GT means the camera poses estimated by COLMAP from all the training images. IL-NeRF recovers accurate camera poses due to the help of incremental camera pose alignment.

Table 12. Comparison of CLNeRF, CLNeRF-CPA on PSNR. The higher the better.

| Method | **Mip-NeRF360** | | | | | | | **NeRF-real360** | | **Block-NeRF** |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bicycle | Bonsai | Counter | Garden | Kitchen | Room | Stump | Pinecone | Vasedeck | |
| CLNeRF | 22.18 | 28.18 | 28.28 | 24.48 | 29.18 | 31.99 | 25.18 | 23.18 | 26.18 | 25.93 |
| CLNeRF-CPA | 21.97 | 29.17 | 27.96 | 25.18 | 28.87 | 30.84 | 23.96 | 23.15 | 25.23 | 25.79 |

Table 13. Comparison of GS, GS-CPA on PSNR. The higher the better.

| Method | **Mip-NeRF360** | | | | | | | **NeRF-real360** | | **Block-NeRF** |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bicycle | Bonsai | Counter | Garden | Kitchen | Room | Stump | Pinecone | Vasedeck | |
| GS | 23.78 | 29.86 | 29.15 | 26.11 | 31.64 | 32.43 | 25.18 | 24.88 | 27.73 | 26.91 |
| GS-CPA | 23.91 | 29.83 | 29.72 | 25.97 | 30.96 | 32.27 | 25.37 | 24.27 | 28.02 | 26.89 |

Table 14. Comparison of NoPe-NeRF and IL-NeRF on PSNR. The higher the better.

| Method | **Mip-NeRF360** | | | | | | | **NeRF-real360** | | **Block-NeRF** |
|---|---|---|---|---|---|---|---|---|---|---|
| | Bicycle | Bonsai | Counter | Garden | Kitchen | Room | Stump | Pinecone | Vasedeck | |
| NoPe-NeRF | 18.35 | 20.76 | 21.38 | 17.54 | 19.81 | 22.74 | 18.91 | 17.65 | 19.36 | 18.97 |
| ILNeRF | 22.34 | 28.96 | 27.82 | 24.82 | 29.34 | 31.45 | 24.89 | 22.93 | 26.15 | 26.58 |