PRINCETON UNIVERSITY

# *Reconstruction*, *Optimization* *and* *Simulation* of Dynamic LiDAR Point Clouds

Zehan Zheng

Dec 6, 2024

# About Me - Zehan Zheng

- Master’s Student @ Tongji University
- Research Interest: **3D Computer Vision**

  *Point Clouds*

  *Neural Rendering*

  *Dynamic Reconstruction*

  *Generative Models*

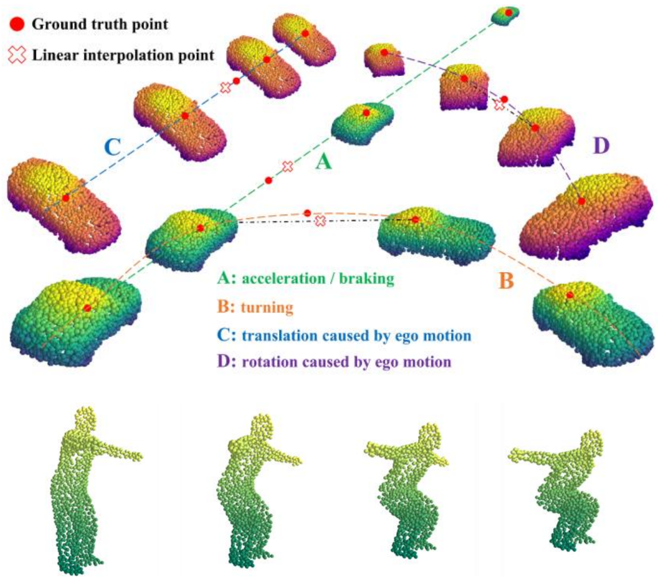
  *Autonomous Driving*


Home Page: https://dyfcalid.github.io

PRINCETON UNIVERSITY

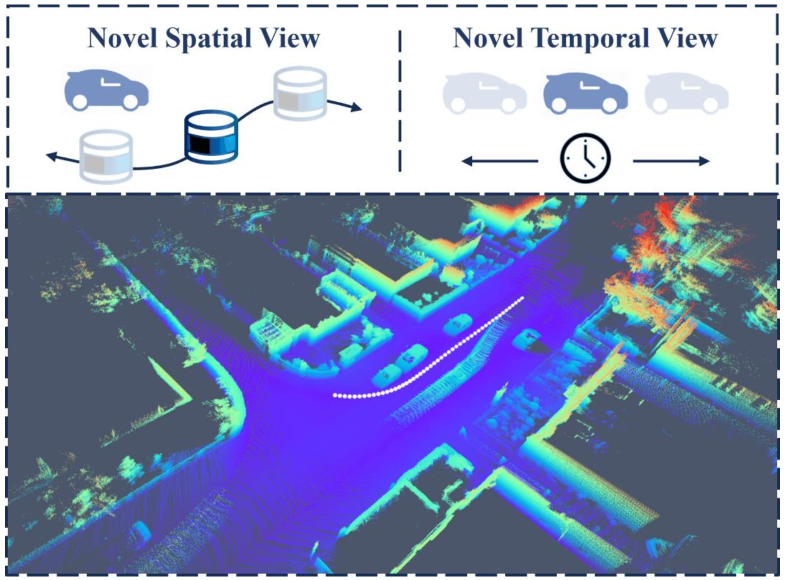
# Dynamic Reconstruction

## Flow Optimization



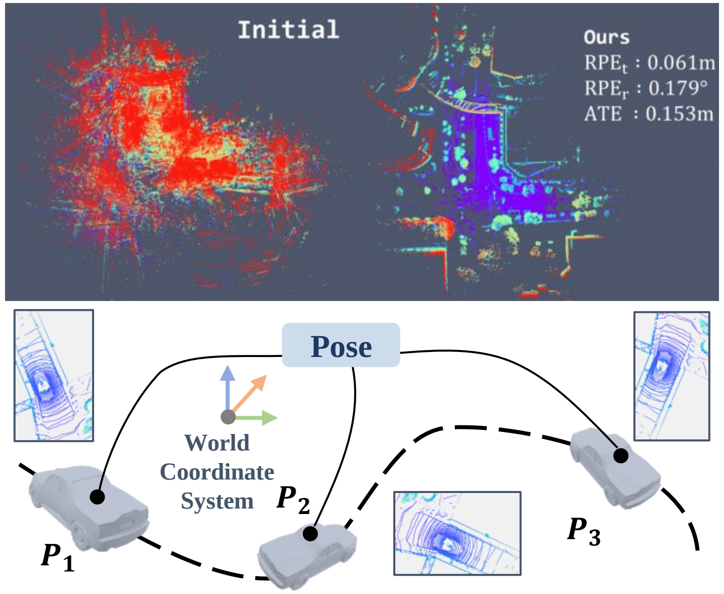

NeuralPCI
[CVPR'23]

## Novel View Synthesis & Simulation




LiDAR4D
[CVPR'24]

## Pose Optimization



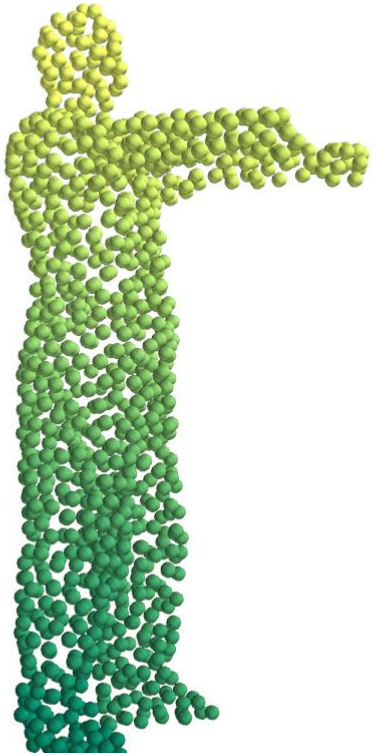
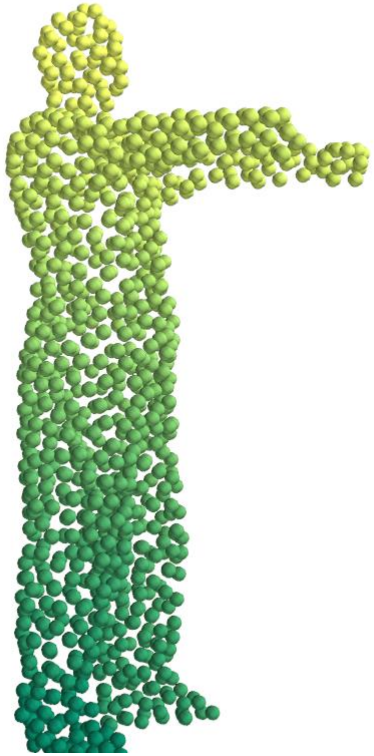
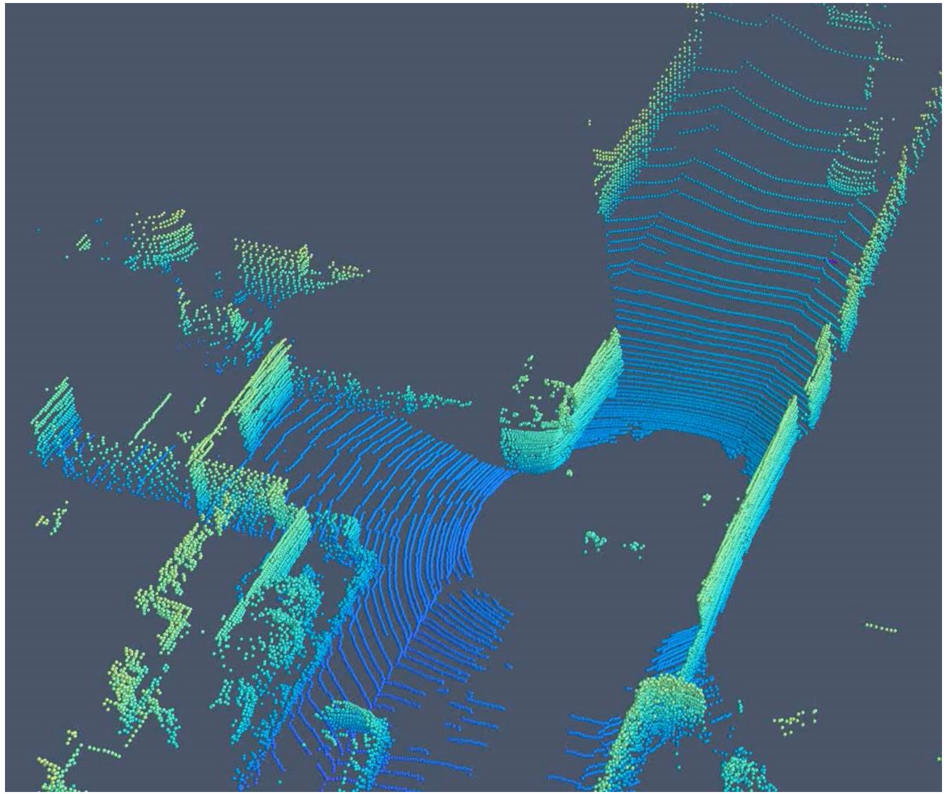

GeoNLF
[NeurIPS'24]

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

SEATTLE, WA CVPR JUNE 17-21, 2024

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS 2024

## Dynamic Point Clouds

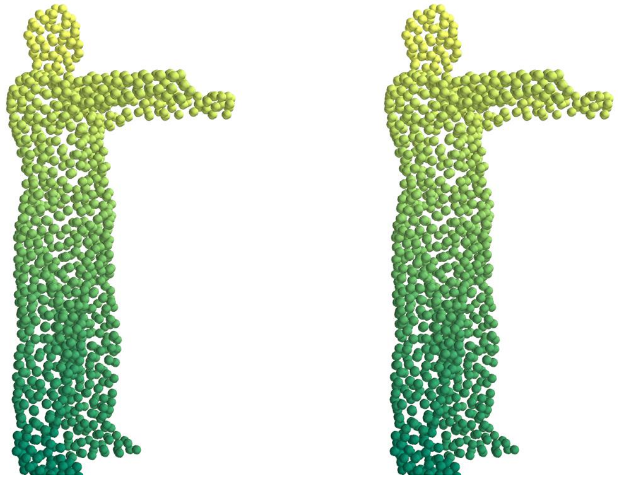




*Some Examples!*

PRINCETON UNIVERSITY

## ■ Dynamic Point Clouds

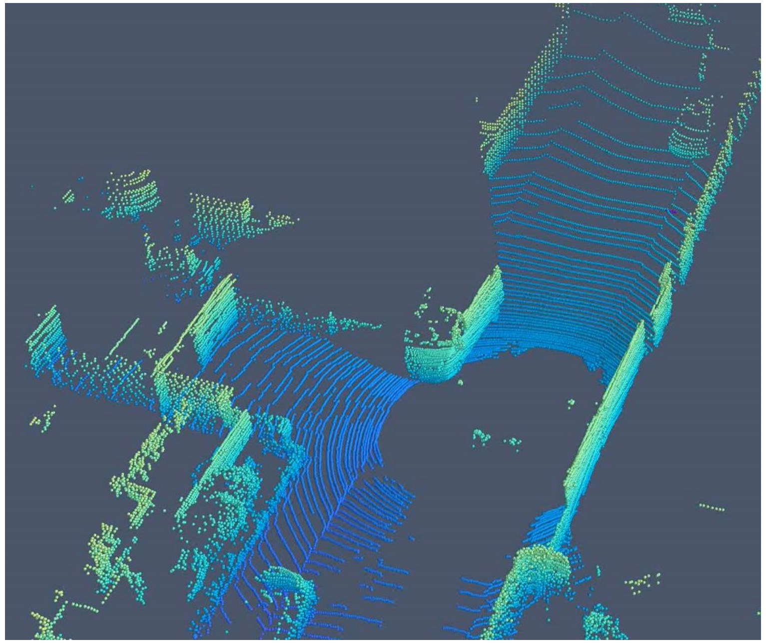
point cloud sequence $\{P_0, P_1, \ldots, P_M\}$, $P_i \epsilon \mathbb{R}^{N\times 3}$



*We humans can understand it easily, but computers are not*

**3D Point Cloud** – *Simple and Effective*

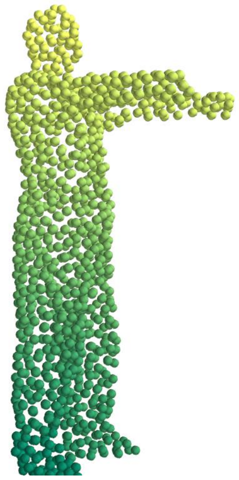
- Discrete
- Irregular
- Unordered
- No correspondence



*Frame1: [[0.44,0.13,0.28], [0.97,0.62,0.15], [0.51,0.79,0.47], ...]*
*Frame2: [[0.12,0.75,0.47], [0.01,0.71,0.33], [0.82,0.19,0.05], ...]*
...

## ■ Point Cloud Interpolation

### Challenges



- **Sparsity** both in spatial and temporal domain

  (limited to the sensors)

- **Point Cloud Structure**

  (Irregular, unordered, and hard to find correspondences between frames)

  cannot interpolate **directly**

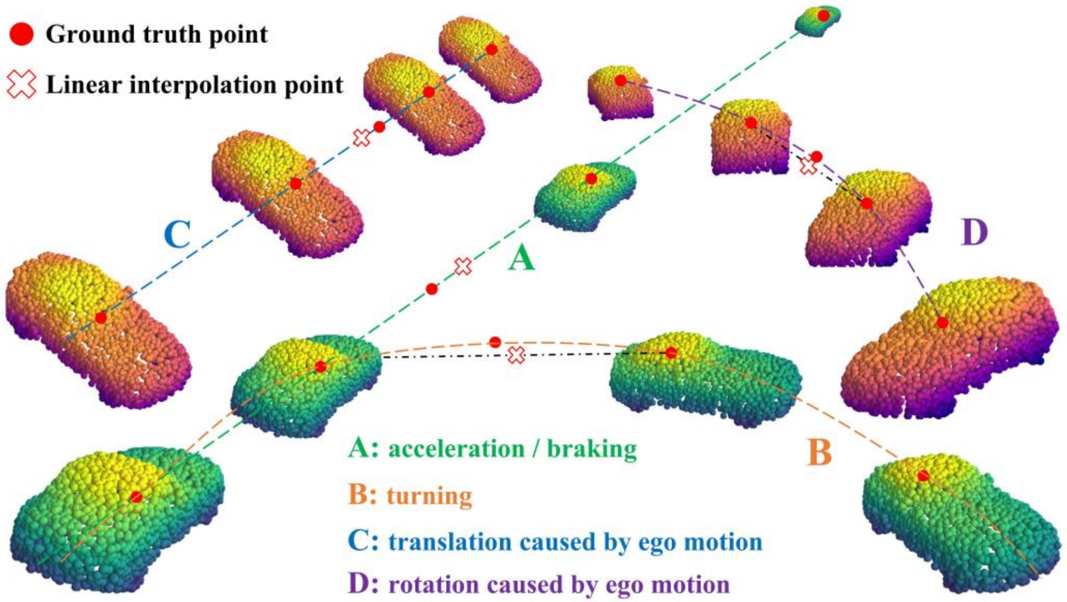
- **Nonlinear Motion**

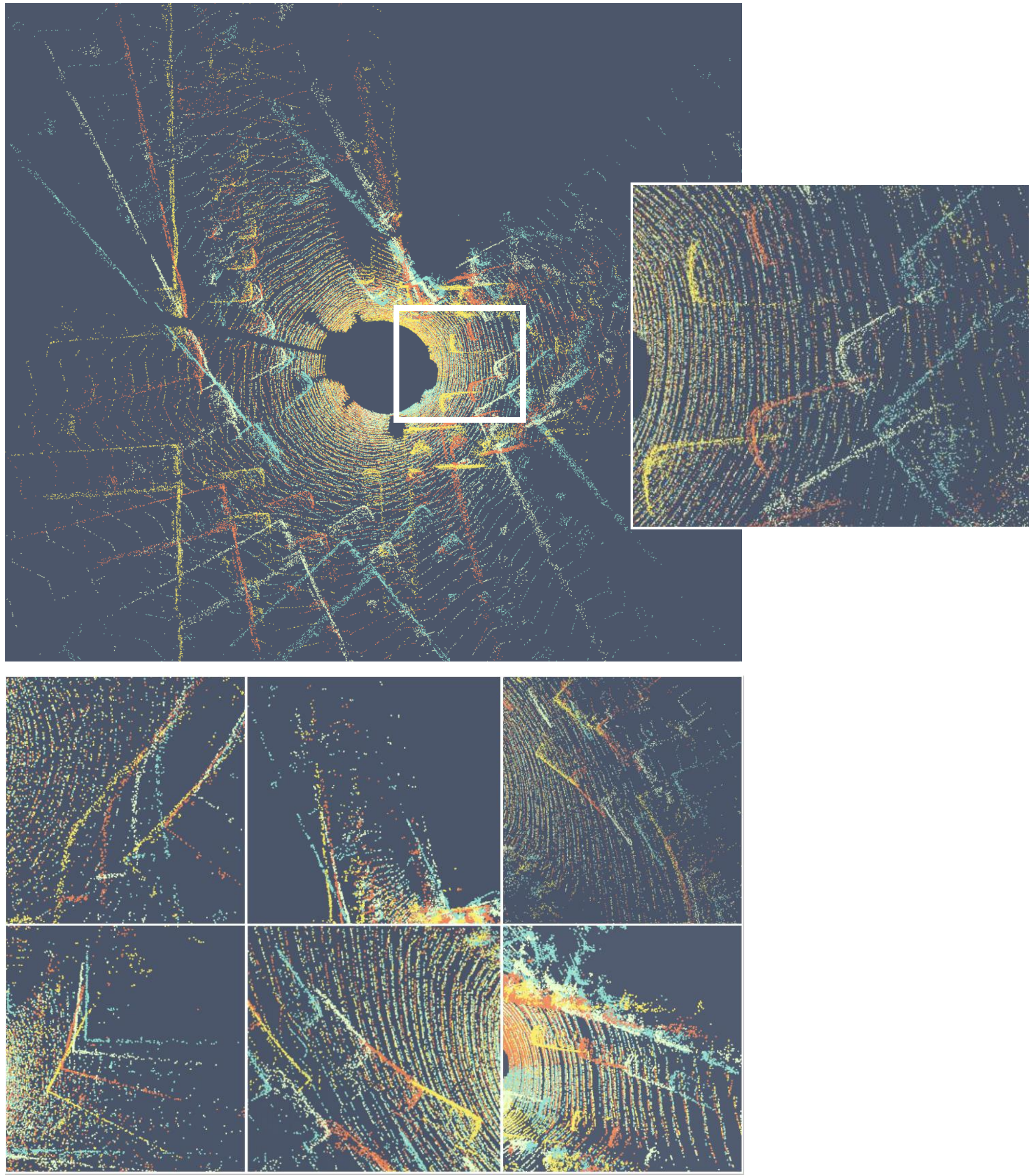
  (i.e. dynamic human / vehicle motions)

  cannot use one simple formula

## Point Cloud Interpolation

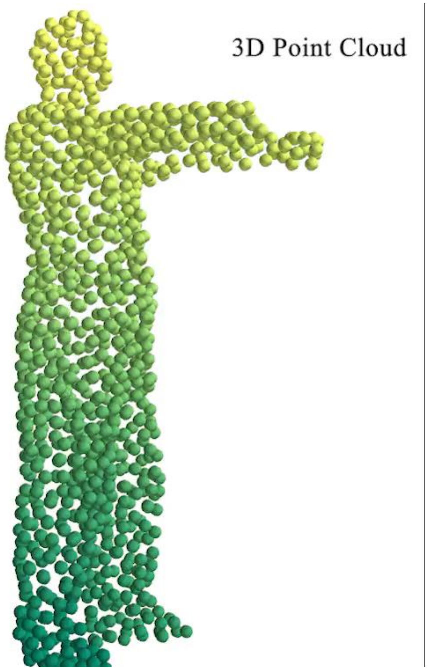
Large amount of nonlinear complex motion in the real-world scenarios



## Point Cloud Interpolation

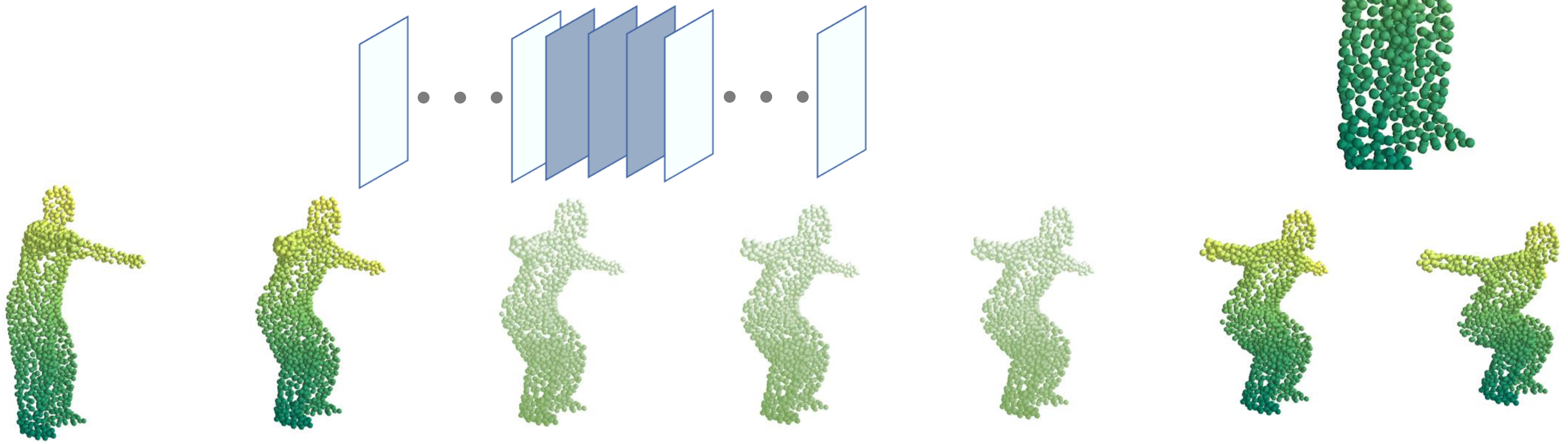
Given the point cloud sequence $\{P_0, P_1, \ \dots, P_M\}$, $P_i \epsilon \mathbb{R}^{N\times 3}$




Interpolate $\boldsymbol{k}$ frames

**Low** Frame Rate → **High** Frame Rate

between every two frames



We can understand because we have the **prior**:

The shape and motion are **continuous**

**Explicit**

**3D Point Cloud**

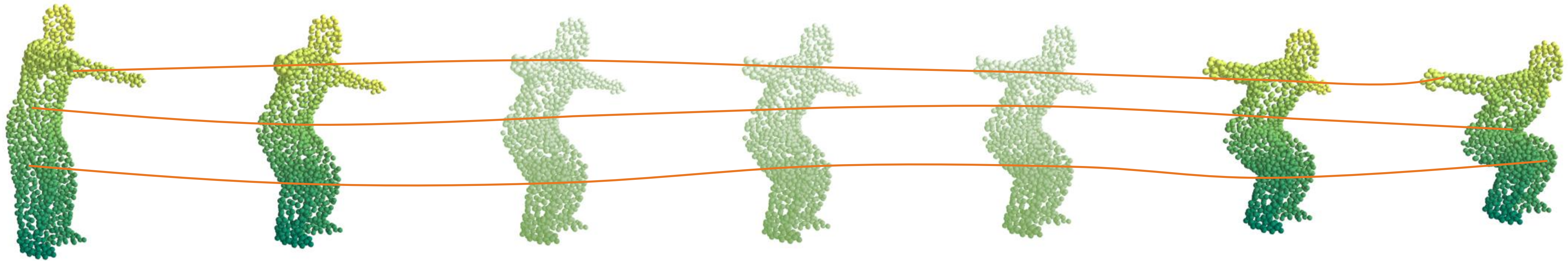
- Discrete
- Irregular
- Unordered
- No correspondence

Continuous

Representation

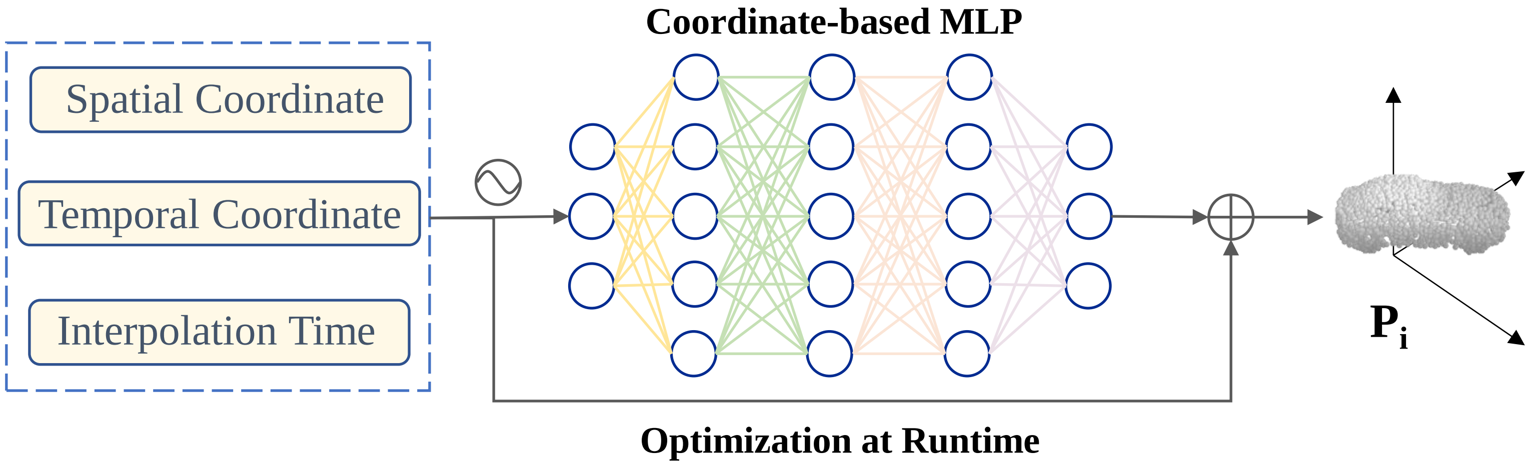
**Implicit**

**Neural Fields**



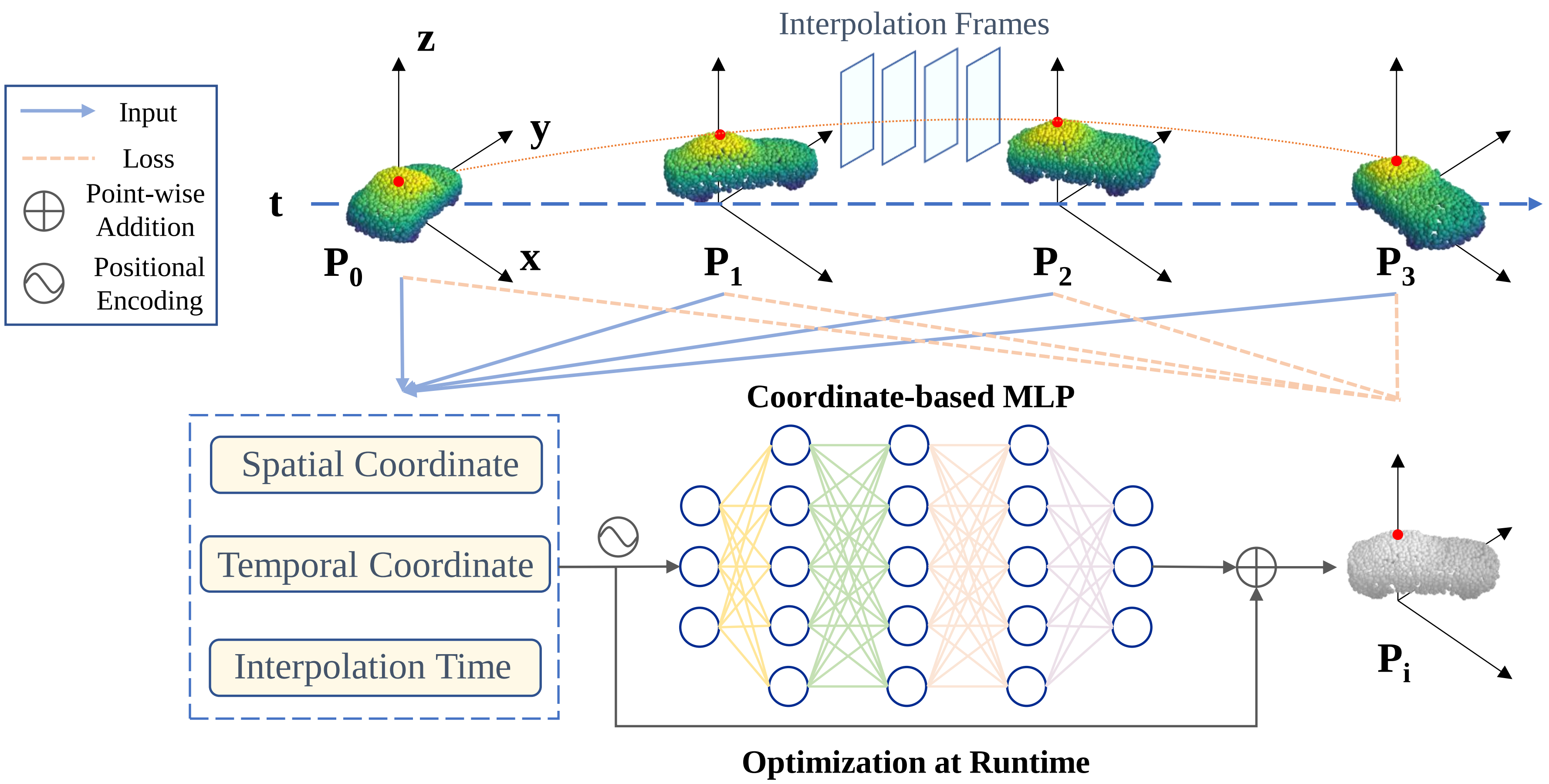
# Method

- **4D** $(x, y, z, t)$ **Spatio-temporal Neural Field**
  - Establish mapping: *Coordinate Field* $\longrightarrow$ *Motion Field*
    $\mathbb{R}^4$ $\mathbb{R}^3$
  - Use *Interpolation Time* to control the output



# Method

- **Multi-frame Integration**



# Method

- **Self-supervised Losses**
  - Chamfer Distance Loss $$\mathcal{L}_{CD} = \frac{1}{N}\sum_{\hat{p}_i \in \hat{P}} \min_{p_i \in P} \|\hat{p}_i - p_i\|_2 + \frac{1}{N}\sum_{p_i \in P} \min_{\hat{p}_i \in \hat{P}} \|p_i - \hat{p}_i\|_2$$
  - Earth Mover's Distance Loss $$\mathcal{L}_{EMD} = \min_{\phi:\hat{P}\to P} \frac{1}{N}\sum_{\hat{p}\in\hat{P}} \|\hat{p} - \phi(\hat{p})\|_2$$
  - Smoothness Loss $$\mathcal{L}_S = \sum_{p_i \in P} \frac{1}{|N(p_i)|} \sum_{p_j \in N(p_i)} \|\Delta x(p_j) - \Delta x(p_i)\|_2^2$$

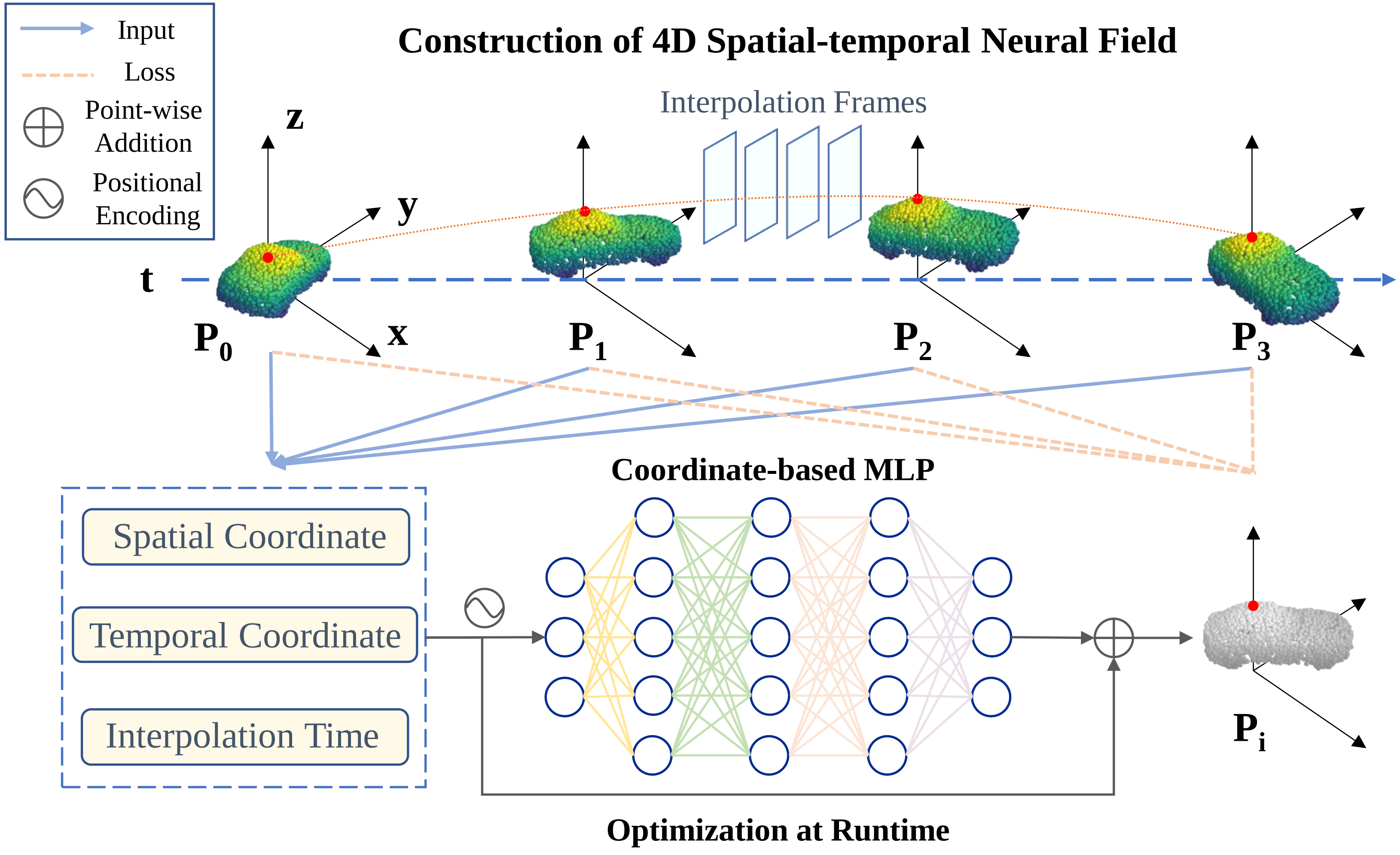
Overall Loss

$$\Psi = \alpha\mathcal{L}_{CD} + \beta\mathcal{L}_{EMD} + \gamma\mathcal{L}_S$$

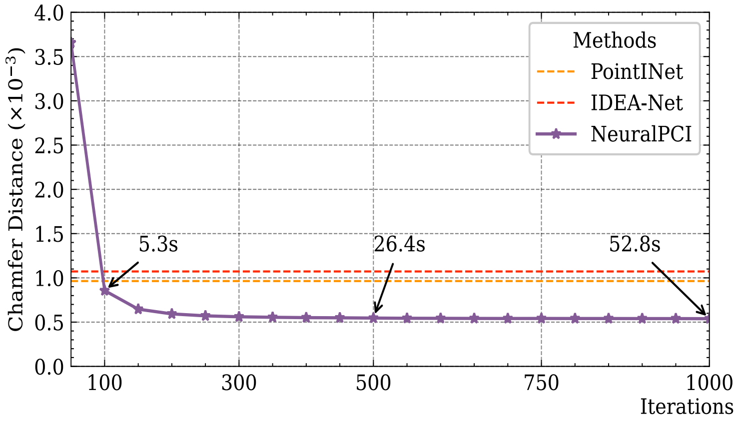
$$\mathcal{L} = \sum_{P_i \in S} \sum_{t_j \in T} \Psi\left(P_{t_j}, \hat{P}_i^{t_j}\right)$$

# Method

- **Runtime optimization**

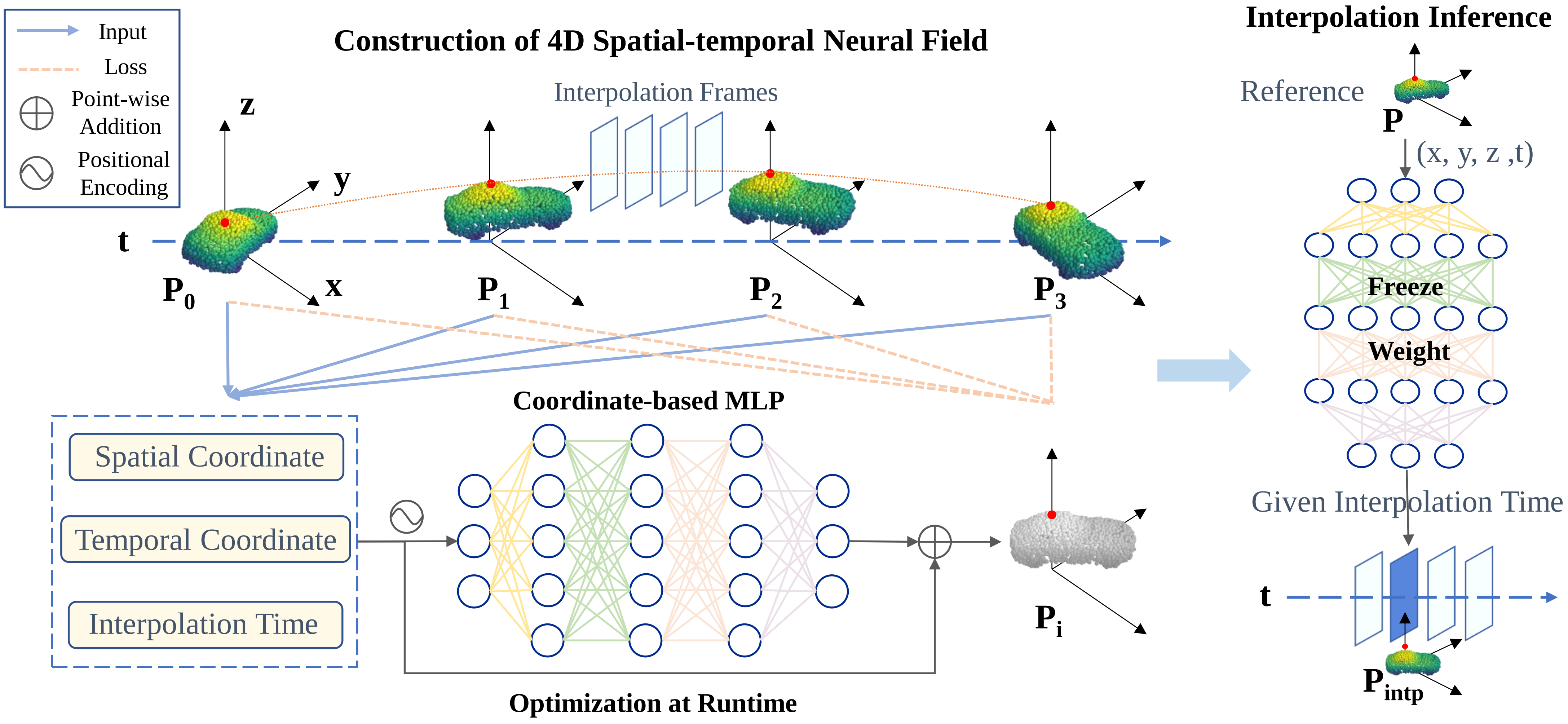



## Fast to converge



# Method

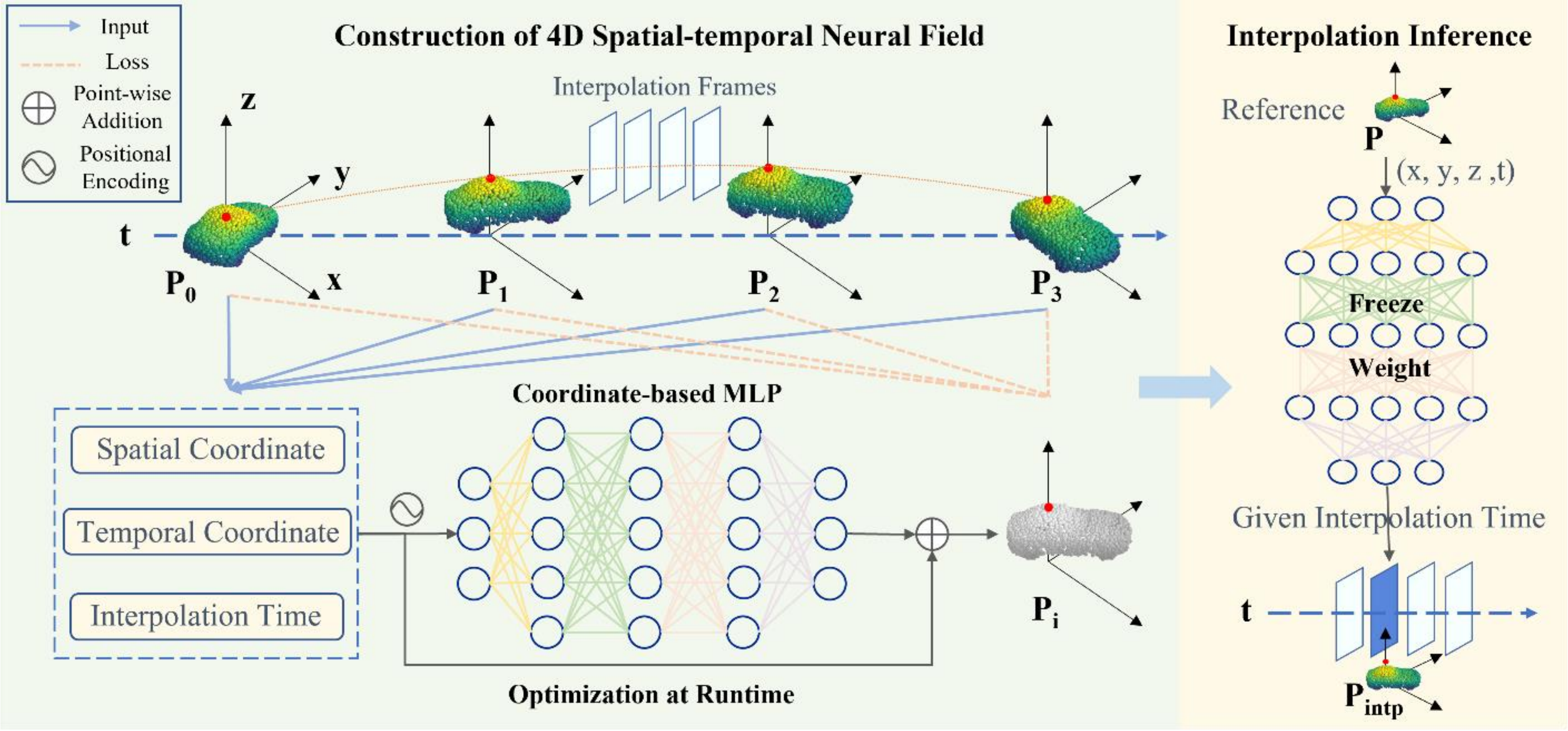
- Runtime optimization & **Inference**



# Method

**NeuralPCI**

- ✓ Multi-frame point cloud interpolation algorithm
- ✓ Deal with both the indoor and outdoor scenarios
- ✓ Integrate motion information implicitly over space and time
- ✓ Output point cloud frames at the arbitrary given time
- ✓ Flexible unified framework for interpolation and extrapolation

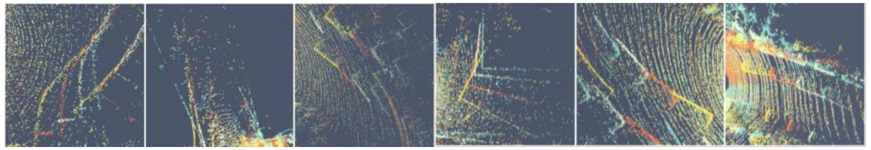


# Experiments

## Results on DHB Dataset

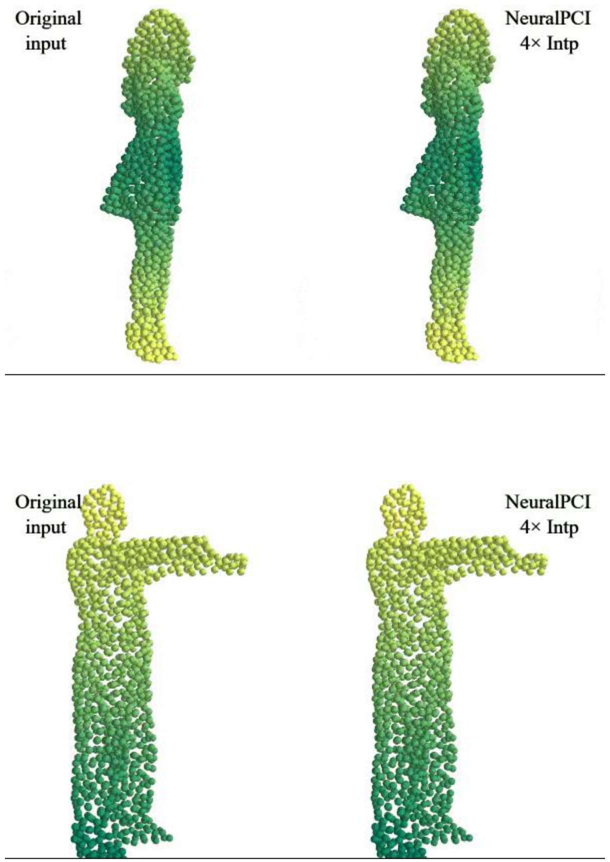
| Methods | Longdress | | Loot | | Red&Black | | Soldier | | Squat | | Swing | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CD | EMD | CD | EMD | CD | EMD | CD | EMD | CD | EMD | CD | EMD | CD ↓ | EMD ↓ |
| IDEA-Net | 0.89 | 6.01 | 0.86 | 8.62 | 0.94 | 10.34 | 1.63 | 30.07 | 0.62 | 6.68 | 1.24 | 6.93 | 1.02 | 12.03 |
| PointINet | 0.98 | 10.87 | 0.85 | 12.10 | 0.87 | 10.68 | 0.97 | 12.39 | 0.90 | 13.99 | 1.45 | 14.81 | 0.96 | 12.25 |
| NSFP | 1.04 | 7.45 | 0.81 | 7.13 | 0.97 | 8.14 | 0.68 | 5.25 | 1.14 | 7.97 | 3.09 | 11.39 | 1.22 | 7.81 |
| PV-RAFT | 1.03 | 6.88 | 0.82 | 5.99 | 0.94 | 7.03 | 0.91 | 5.31 | 0.57 | 2.81 | 1.42 | 10.54 | 0.92 | 6.14 |
| NeuralPCI | **0.70** | **4.36** | **0.61** | **4.76** | **0.67** | **4.79** | **0.59** | **4.63** | **0.03** | **0.02** | **0.53** | **2.22** | **0.54** | **3.68** |

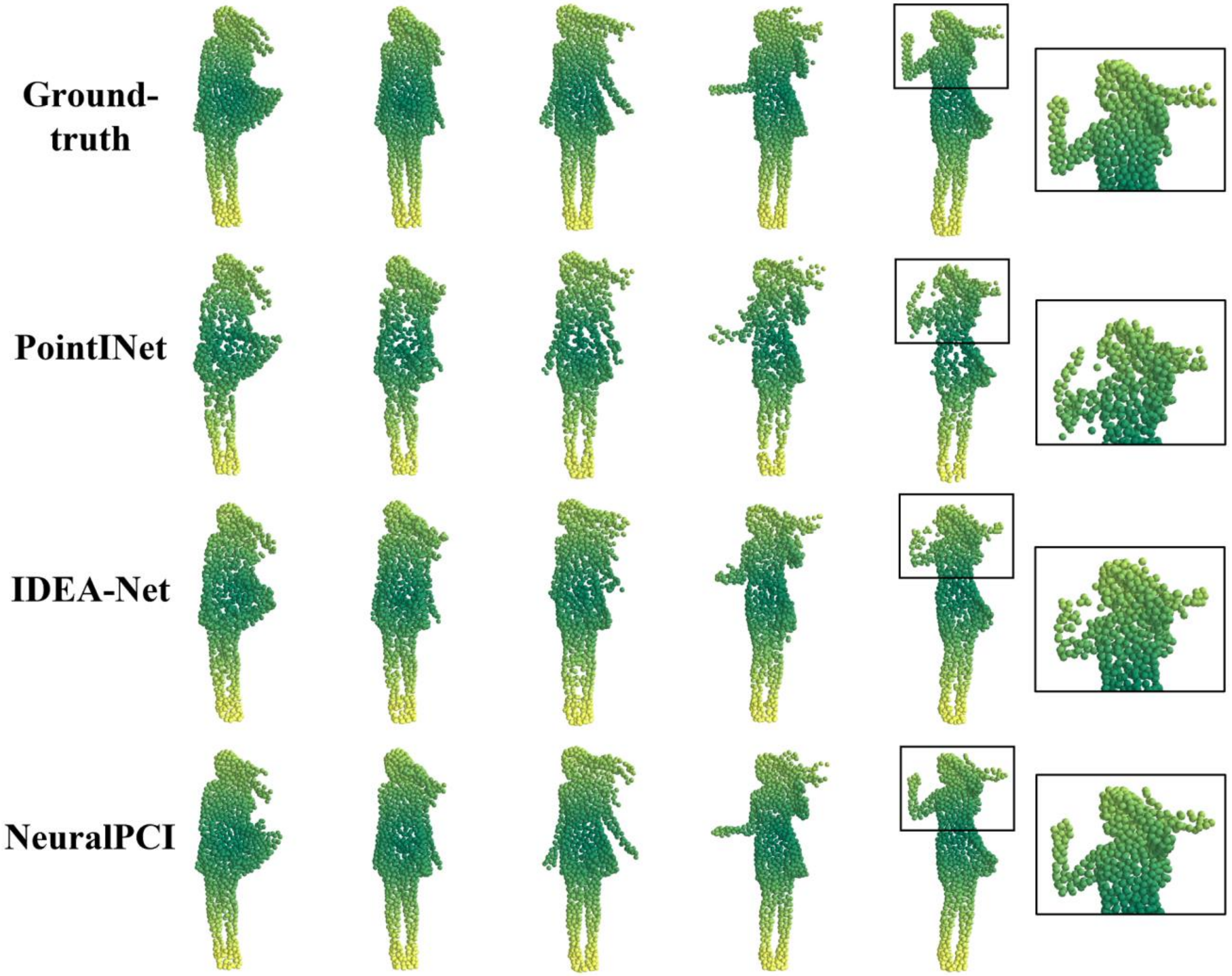
## Results on NL-Drive Dataset



| Methods | Type | Frame-1 | | Frame-2 | | Frame-3 | | Average | |
|---|---|---|---|---|---|---|---|---|---|
| | | CD | EMD | CD | EMD | CD | EMD | CD ↓ | EMD ↓ |
| NSFP | forward flow | 0.94 | 95.18 | 1.75 | 132.30 | 2.55 | 168.91 | 1.75 | 132.13 |
| | backward flow | 2.53 | 168.75 | 1.74 | 132.19 | 0.95 | 95.23 | 1.74 | 132.05 |
| PV-RAFT | forward flow | 1.36 | 104.57 | 1.92 | 146.87 | 1.63 | 169.82 | 1.64 | 140.42 |
| | backward flow | 1.58 | 173.18 | 1.85 | 145.48 | 1.30 | 102.71 | 1.58 | 140.46 |
| PointINet | bi-directional flow | 0.93 | 97.48 | 1.24 | **110.22** | 1.01 | 95.65 | 1.06 | 101.12 |
| NeuralPCI | neural field | **0.72** | **89.03** | **0.94** | 113.45 | **0.74** | **88.61** | **0.80** | **97.03** |

# Experiments



# NeuralPCI

*Neural field is awesome!*

- Convert **explicit** point clouds into **implicit** neural fields
- Reconstruct multi-frame point clouds with **unified** and **continuous** representations
- Optimize motion in a **self-supervised** manner
- Benefit from the **fitting ability** and **smoothness** of MLP

## Limitation & Assumption

- We assume the point clouds relatively complete
- We assume the point clouds are object-centric

*Flow Optimization*

↓

*Synthesize Point Clouds*

↑

*Large-scale Dynamic Scene Reconstruction*

## What if …

- Incomplete and partial point clouds
- Scene-centric point clouds (world coordinate system)

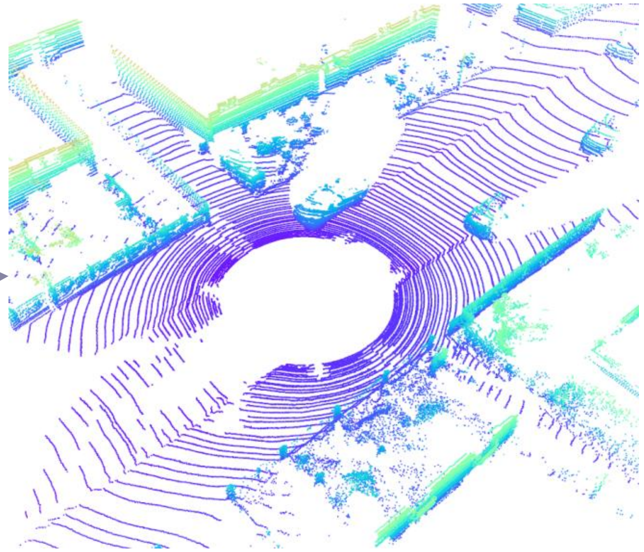
# LiDAR Point Clouds

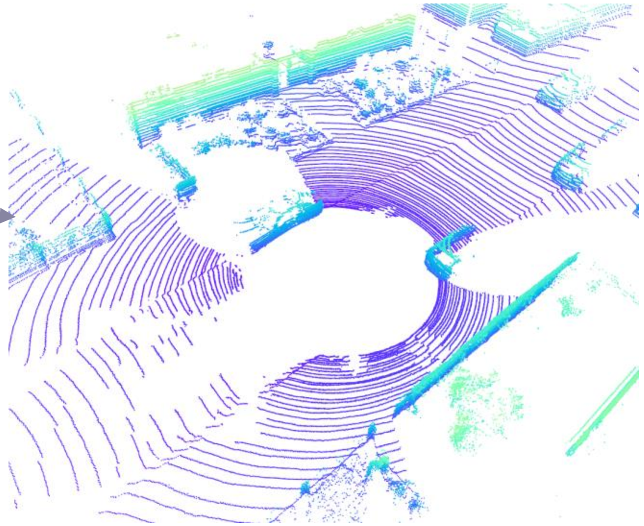
- LiDAR serves as the crucial sensor of autonomous driving for accurate 3D perception

- Sparsity and occlusion
- Varying at different locations and times

- Costly acquisition for a large-scale dataset
- Limited to specific sensor configuration and ego-vehicle trajectory

How can we generate/synthesize novel point clouds?

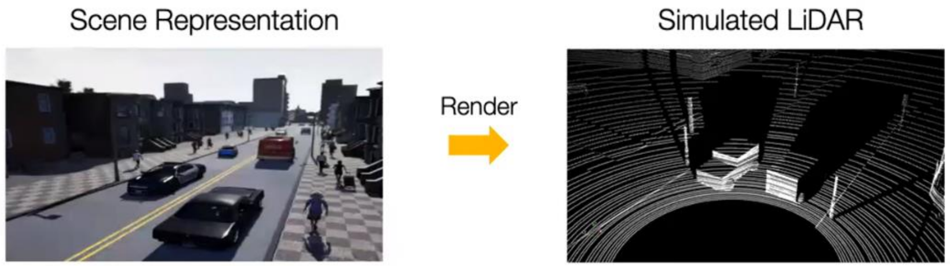




PRINCETON UNIVERSITY

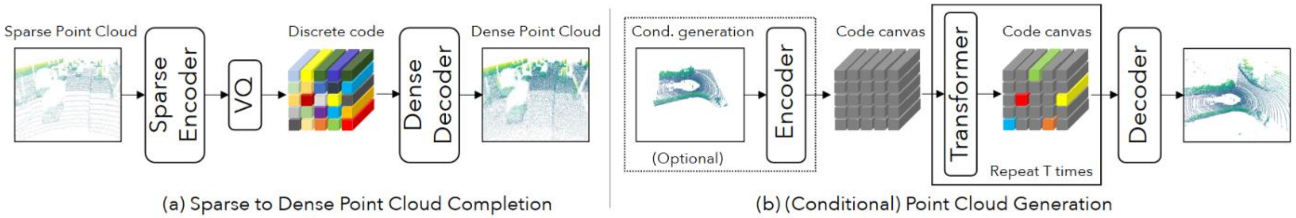
# Previous Methods

- Physical-based Simulation
  - ✕ Costly 3D assets
  - ✕ Domain gap



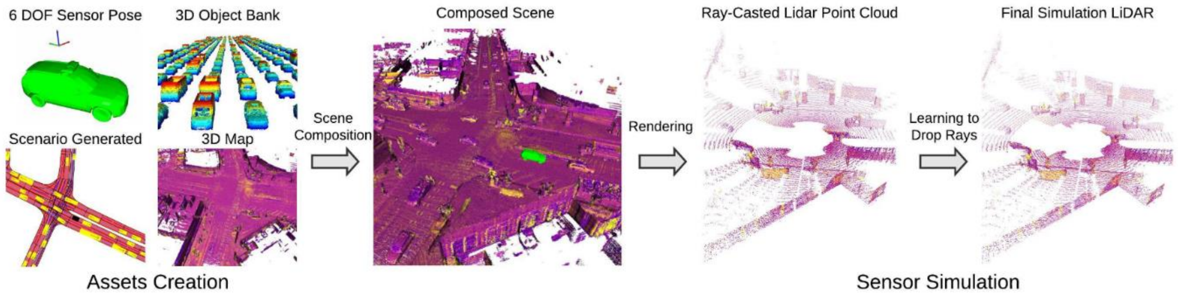

CARLA: An open urban driving simulator

- Generative Models
  - ✕ Hard to control/edit
  - ✕ Poor generalization




Learning Compact Representations for LiDAR Completion and Generation

- Scene Reconstruction
  - ✓ Realistic
  - ✓ Precise control



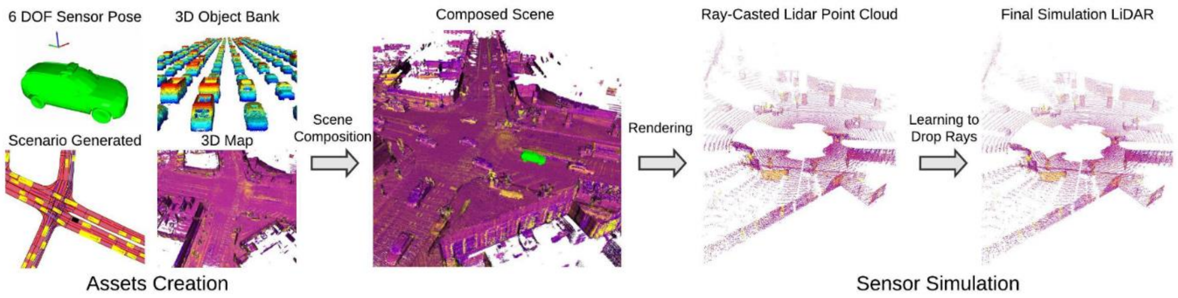

LiDARsim: Realistic LiDAR Simulation by Leveraging the Real World

# Previous Methods

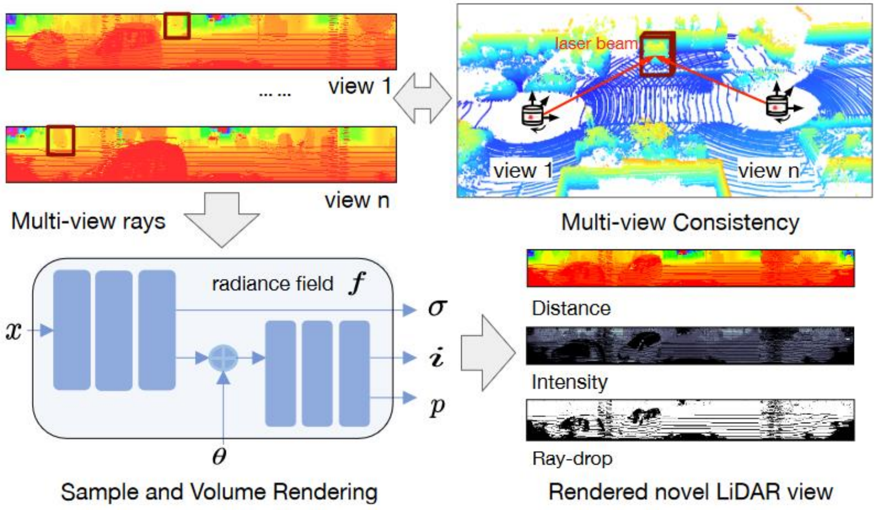
- Scene Reconstruction
  - ✓ Realistic
  - ✓ Precise control

Differentiable Rendering

**Novel View Synthesis**




LiDARsim: Realistic LiDAR Simulation by Leveraging the Real World




LiDAR-NeRF: Novel LiDAR View Synthesis via Neural Radiance Fields

✕ Complicated

✕ Limited to static scenes

How to deal with dynamic scenarios?

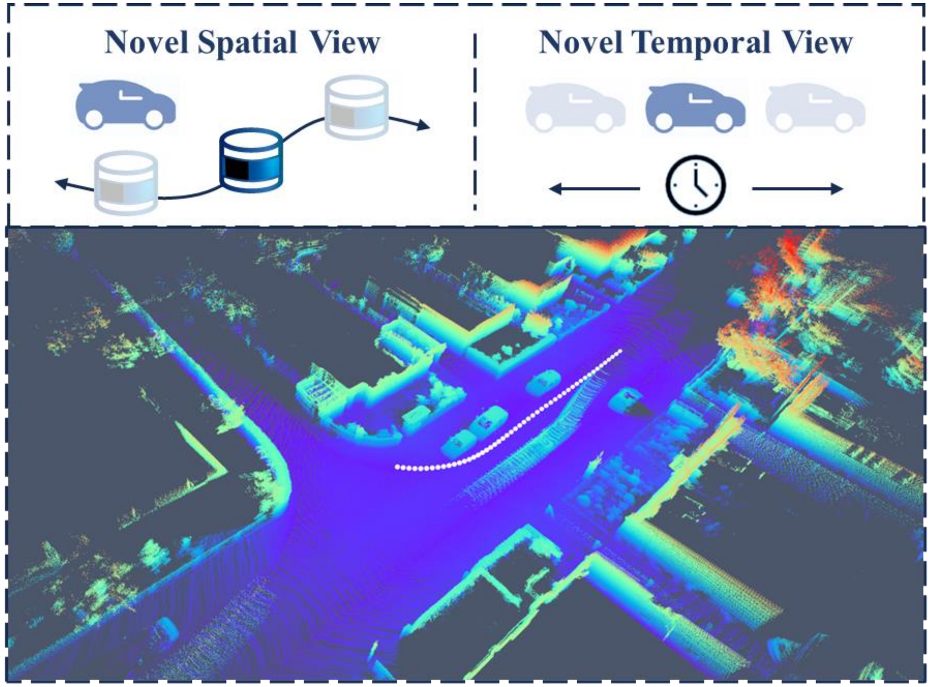
# Novel Space-time View LiDAR Synthesis

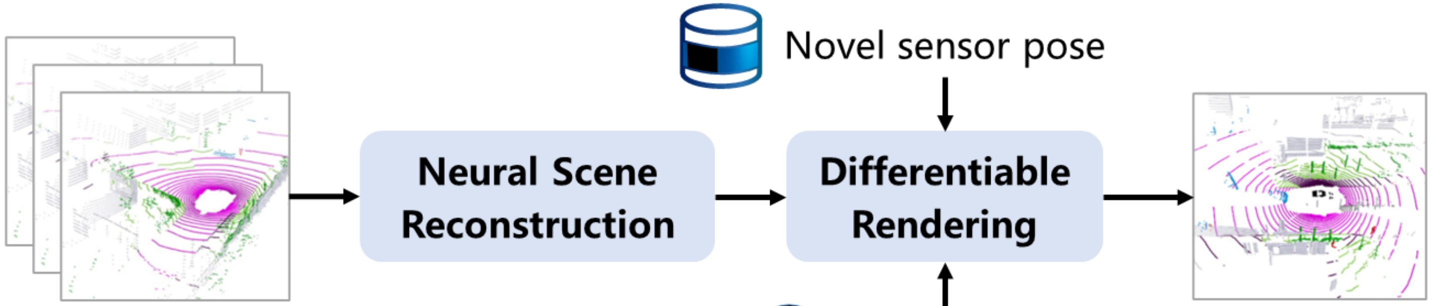
## Input

- LiDAR point cloud sequence $S = \{S_0, S_1, \ldots, S_{n-1}\}$ ( $S_i \in \mathbb{R}^{N\times 4}$ , *including intensity*)
- sensor poses $P = \{P_0, P_1, \ldots, P_M\}$ ( $P_i \in SE(3)$)
- timestamps $T = \{t_0, t_1, \ldots, t_{n-1}\}$ ($t_i \in \mathbb{R}$)

## Output

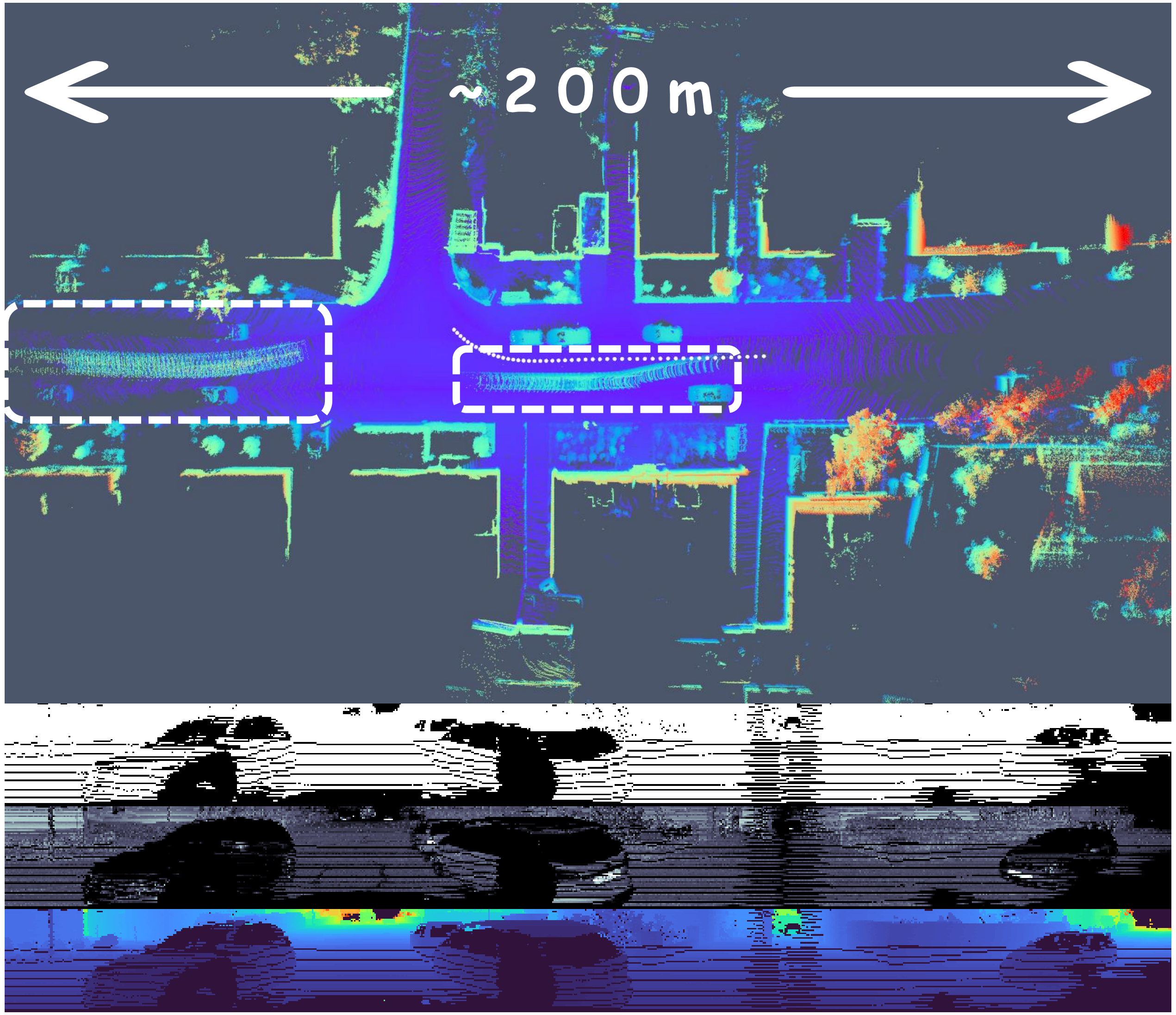
- LiDAR point cloud $S_{novel}$ given novel pose $P_{novel}$ and novel time $t_{novel}$



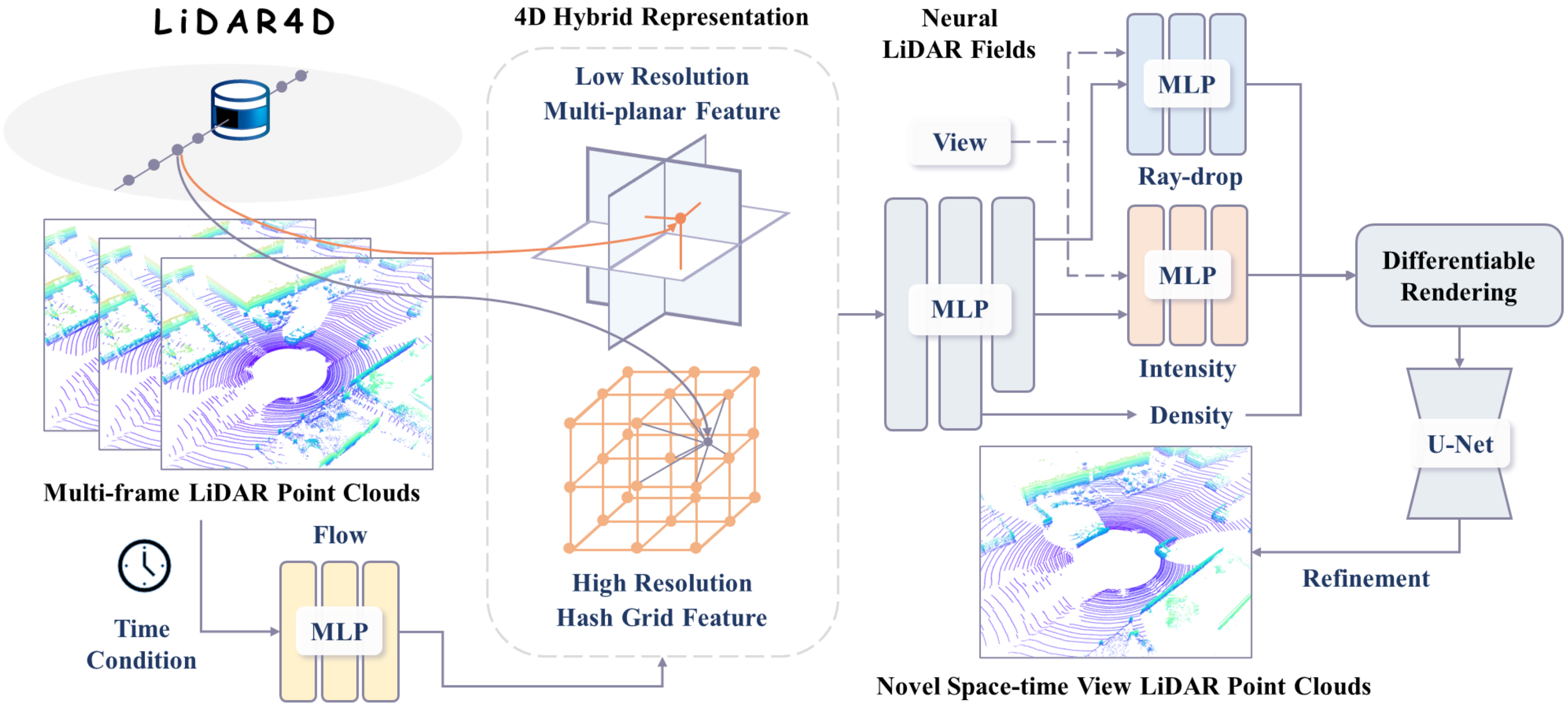
# However, challenges remain …

- **Large-scale reconstruction**
  - Scenes spanning hundreds of meters
  - Representation resolution
  - Sparsity of point clouds
- **Dynamic scenarios**
  - Long-distance vehicle motion
  - Temporal consistency
- **Generation realism**
  - Intensity reconstruction
  - Ray-drop characteristic



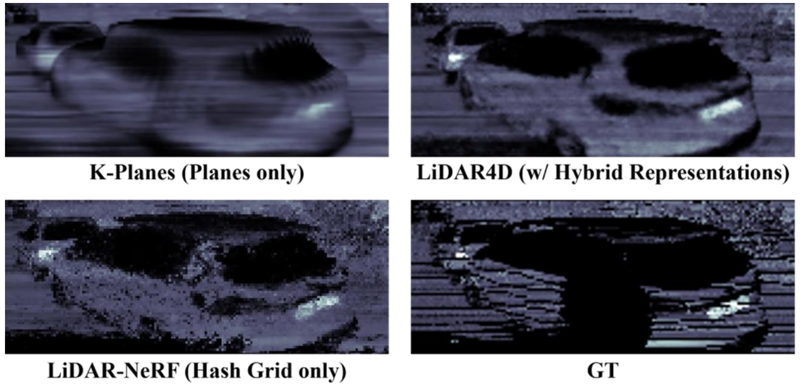
# Method — LiDAR4D

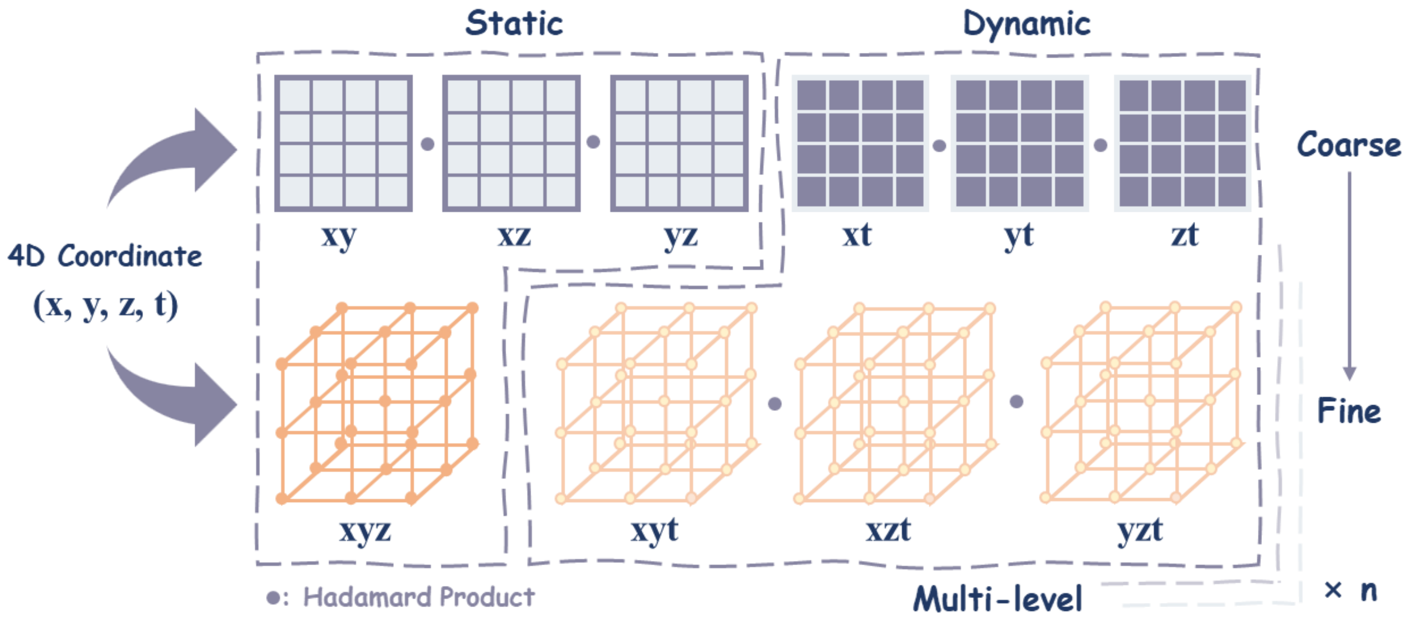
- ✓ Differentiable LiDAR-only framework for novel space-time LiDAR view synthesis
- ✓ Geometry-aware and time-consistent large-scale dynamic reconstruction
- ✓ Better generation realism with global refinement

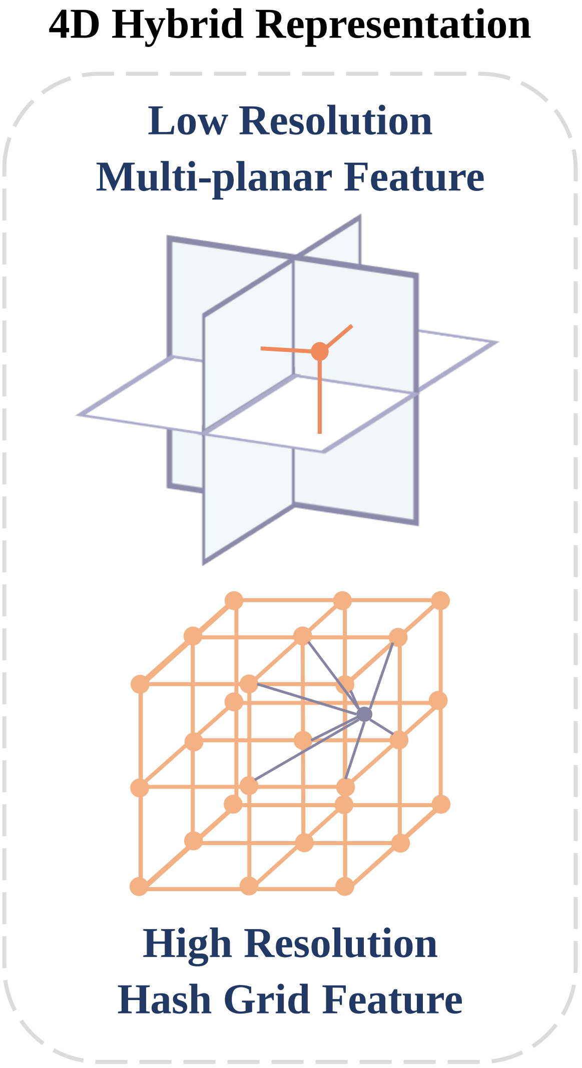


# Method

- **Hybrid Representation**
  - Planes & Hash Grids
  - Coarse-to-fine Resolution
  - 4D Decomposition

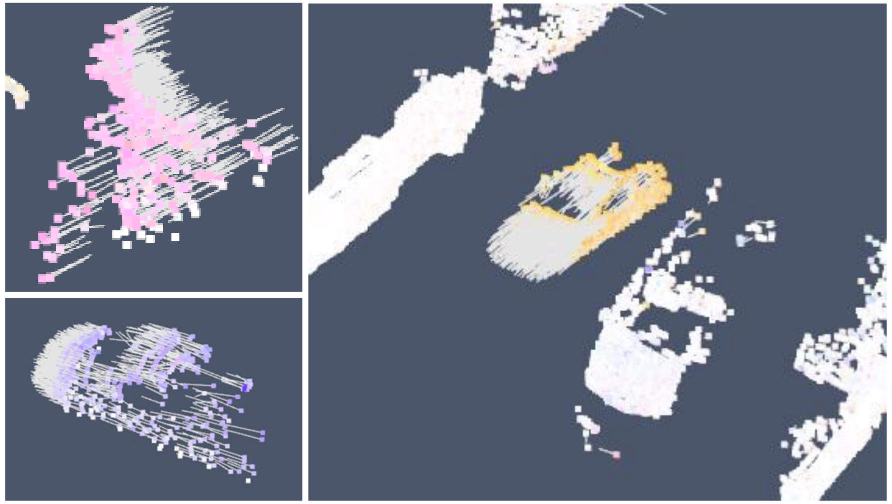


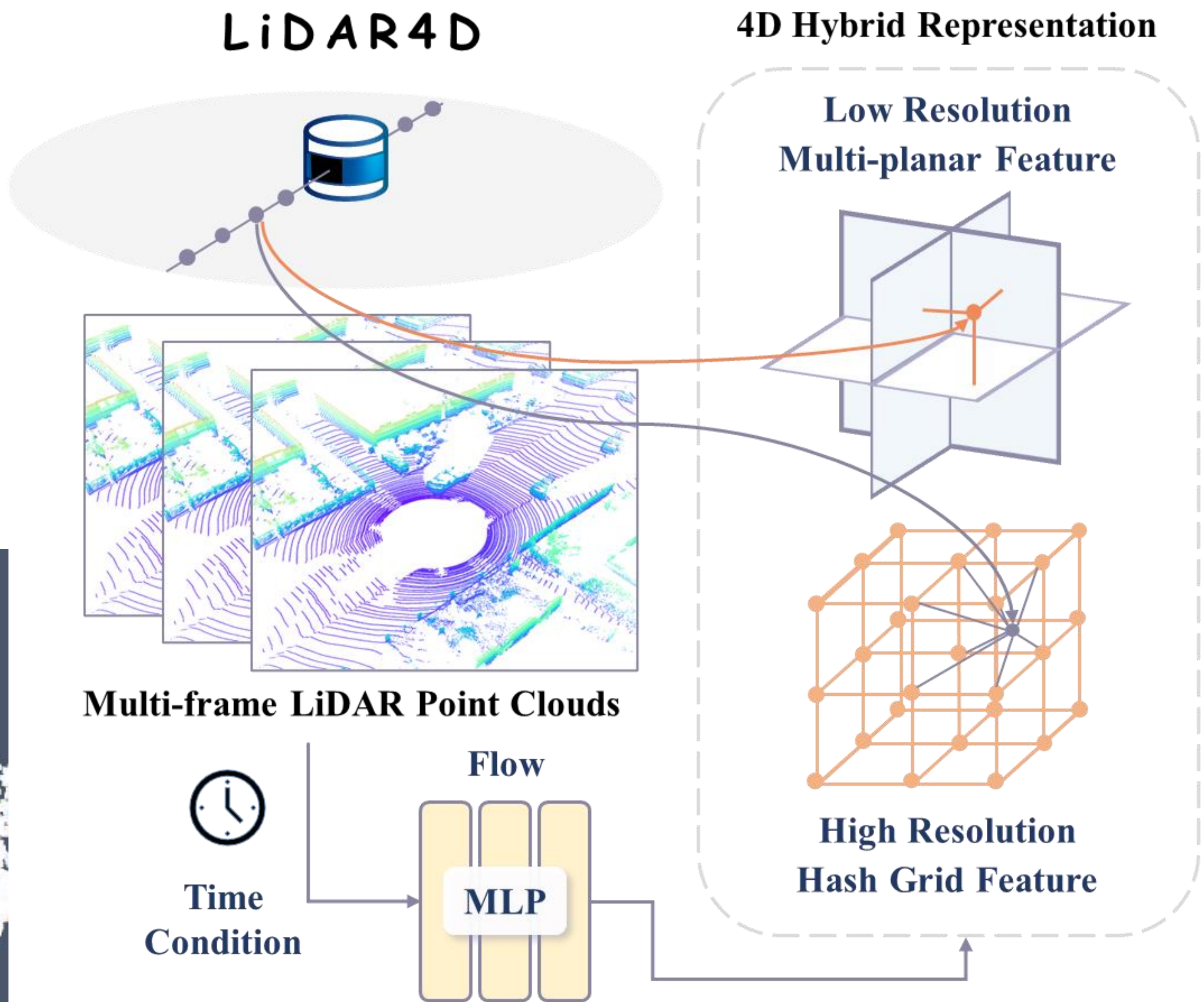



# Method

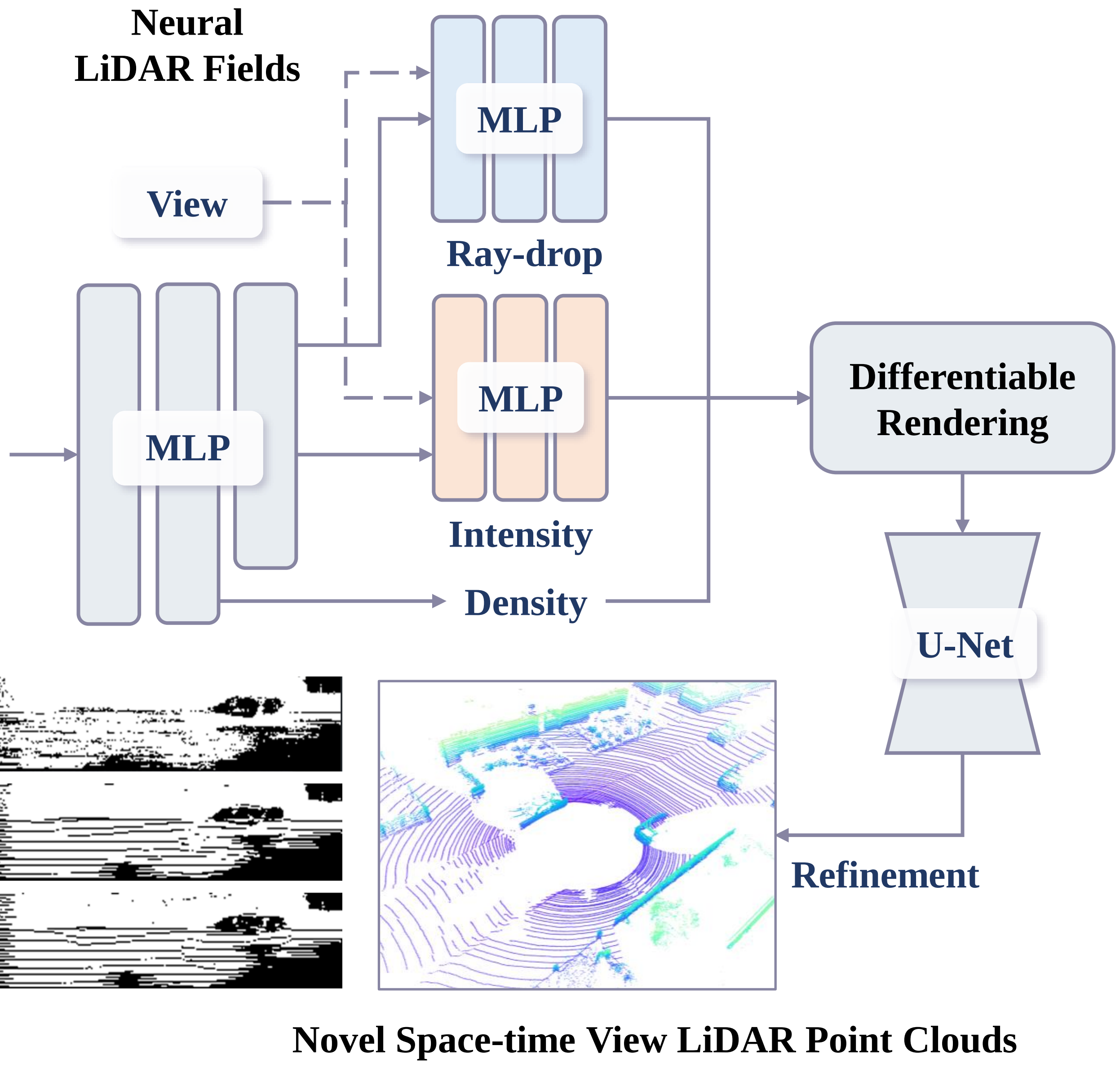
- Hybrid Representation
- **Scene Flow Prior**
  - Flow MLP
  - Geometry-aware Constraint (Chamfer Distance)
  - Temporal Feature Aggregation

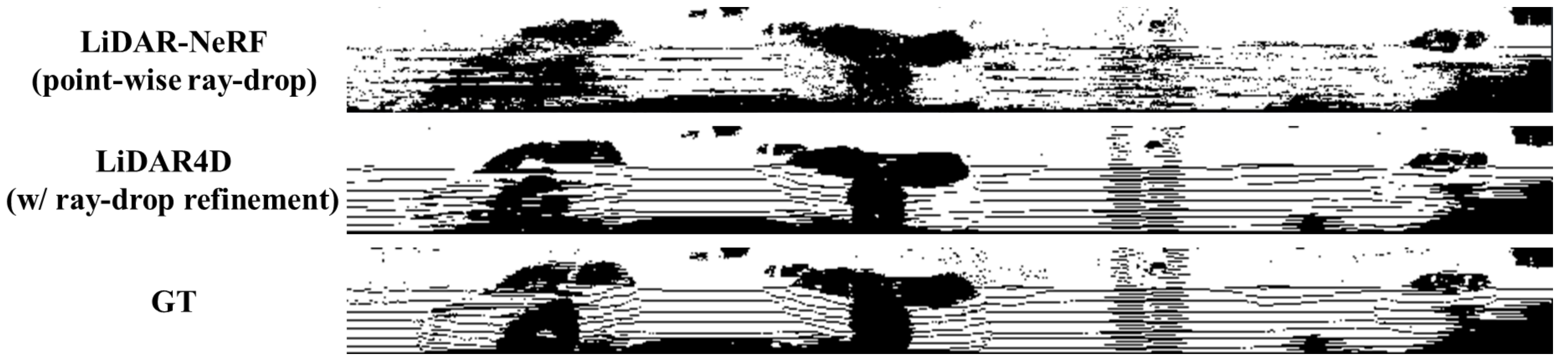




# Method

- Hybrid Representation
- Scene Flow Prior
- **Neural LiDAR Fields**
  - Separate MLPs for Depth/Intensity/Ray-drop
  - Global Optimization for Ray-drop Refinement via U-Net

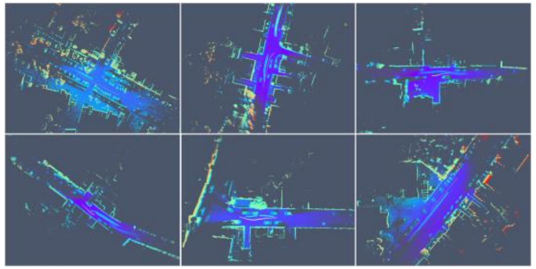


PRINCETON UNIVERSITY

# Experiments

- **SOTA Results**
  - KITTI-360



| Method | Type | Point Cloud | | Depth | | | | | Intensity | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CD↓ | F-score↑ | RMSE↓ | MedAE↓ | LPIPS↓ | SSIM↑ | PSNR↑ | RMSE↓ | MedAE↓ | LPIPS↓ | SSIM↑ | PSNR↑ |
| LiDARsim [25] | $\mathcal{E}$. / $\mathcal{S}$. / $\mathcal{M}$. | 3.2228 | 0.7157 | 6.9153 | 0.1279 | 0.2926 | 0.6342 | 21.4608 | 0.1666 | 0.0569 | 0.3276 | 0.3502 | 15.5853 |
| NKSR [15] | $\mathcal{E}$. / $\mathcal{S}$. / $\mathcal{M}$. | 1.8982 | 0.6855 | 5.8403 | 0.0996 | 0.2752 | 0.6409 | 23.0368 | 0.1742 | 0.0590 | 0.3337 | 0.3517 | 15.2081 |
| PCGen [19] | $\mathcal{E}$. / $\mathcal{S}$. | 0.4636 | 0.8023 | 5.6583 | 0.2040 | 0.5391 | 0.4903 | 23.1675 | 0.1970 | 0.0763 | 0.5926 | 0.1351 | 14.1181 |
| LiDAR-NeRF [39] | $\mathcal{I}$. / $\mathcal{S}$. | 0.1438 | 0.9091 | 4.1753 | 0.0566 | 0.2797 | 0.6568 | 25.9878 | 0.1404 | 0.0443 | 0.3135 | 0.3831 | 17.1549 |
| D-NeRF [32] | $\mathcal{I}$. / $\mathcal{D}$. | 0.1442 | 0.9128 | 4.0194 | 0.0508 | 0.3061 | 0.6634 | 26.2344 | 0.1369 | 0.0440 | 0.3409 | 0.3748 | 17.3554 |
| TiNeuVox-B [9] | $\mathcal{I}$. / $\mathcal{D}$. | 0.1748 | 0.9059 | 4.1284 | 0.0502 | 0.3427 | 0.6514 | 26.0267 | 0.1363 | 0.0453 | 0.4365 | 0.3457 | 17.3535 |
| K-Planes [12] | $\mathcal{I}$. / $\mathcal{D}$. | 0.1302 | 0.9123 | 4.1322 | 0.0539 | 0.3457 | 0.6385 | 26.0236 | 0.1415 | 0.0498 | 0.4081 | 0.3008 | 17.0167 |
| **LiDAR4D (Ours)** | $\mathcal{I}$. / $\mathcal{D}$. | **0.1089** | **0.9272** | **3.5256** | **0.0404** | **0.1051** | **0.7647** | **27.4767** | **0.1195** | **0.0327** | **0.1845** | **0.5304** | **18.5561** |

Table 1. **Quantitative comparison on KITTI-360 dataset**. We compare our method to different types of previous approaches and color the top results as best and second best. $\mathcal{E}$: Explicit, $\mathcal{I}$ : Implicit, $\mathcal{S}$: Static, $\mathcal{D}$: Dynamic, $\mathcal{M}$: Mesh.

  - NuScenes

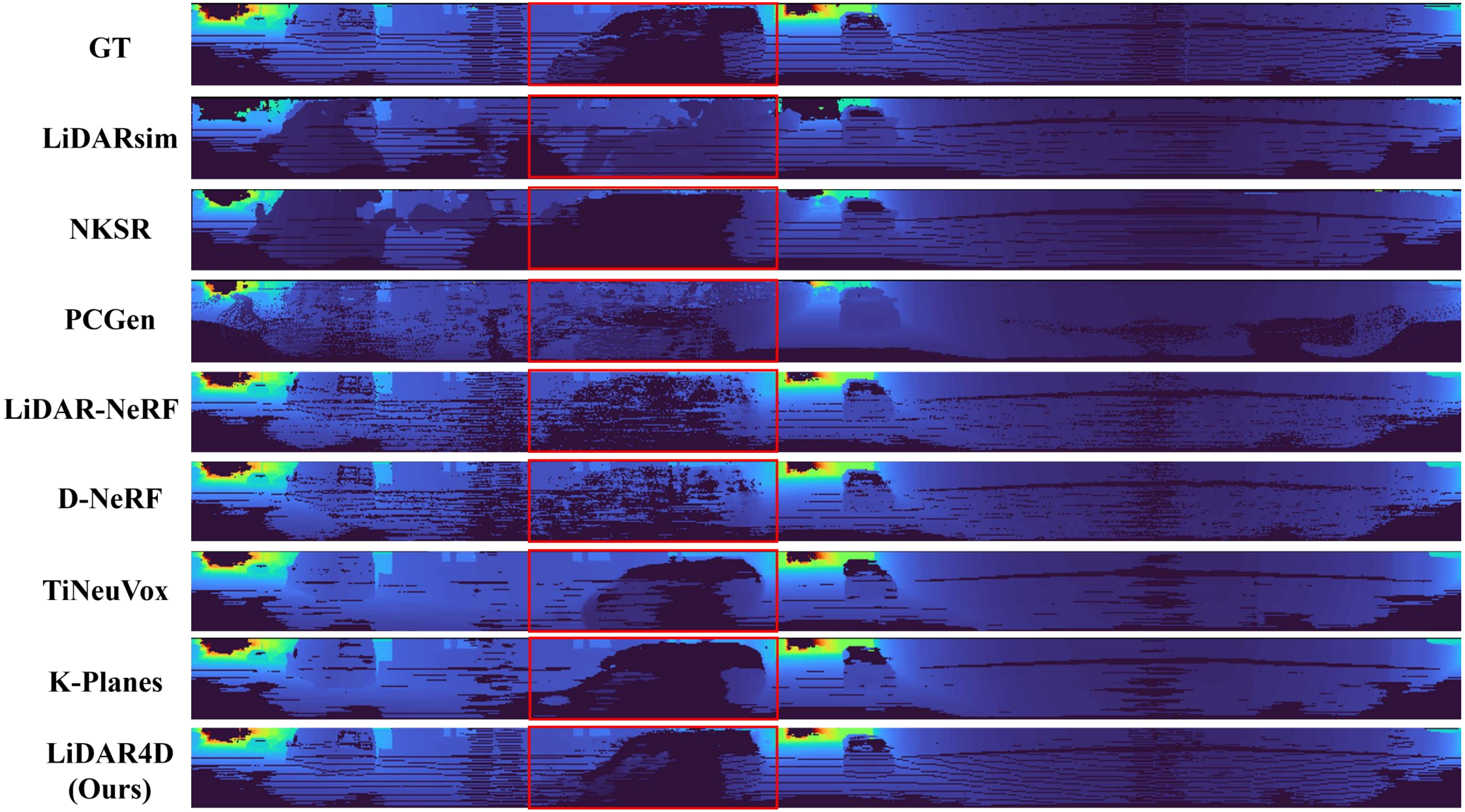
| Method | Type | Point Cloud | | Depth | | | | | Intensity | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CD↓ | F-score↑ | RMSE↓ | MedAE↓ | LPIPS↓ | SSIM↑ | PSNR↑ | RMSE↓ | MedAE↓ | LPIPS↓ | SSIM↑ | PSNR↑ |
| LiDARsim [25] | $\mathcal{E}$. / $\mathcal{S}$. / $\mathcal{M}$. | 12.1383 | 0.6512 | 10.5539 | 0.3572 | 0.1871 | 0.5653 | 17.7841 | 0.0659 | 0.0115 | 0.1160 | 0.5170 | 23.7791 |
| NKSR [15] | $\mathcal{E}$. / $\mathcal{S}$. / $\mathcal{M}$. | 11.4910 | 0.6178 | 9.3731 | 0.5763 | 0.2111 | 0.5637 | 18.7774 | 0.0680 | 0.0119 | 0.1290 | 0.5031 | 23.4905 |
| PCGen [19] | $\mathcal{E}$. / $\mathcal{S}$. | 2.1998 | 0.6341 | 8.8364 | 0.4011 | 0.1792 | 0.5440 | 19.2799 | 0.0768 | 0.0147 | 0.1308 | 0.4410 | 22.4428 |
| LiDAR-NeRF [39] | $\mathcal{I}$. / $\mathcal{S}$. | 0.3225 | 0.8576 | 7.1566 | 0.0338 | 0.0702 | 0.7188 | 21.2129 | 0.0467 | 0.0076 | 0.0483 | 0.7264 | 26.9927 |
| D-NeRF [32] | $\mathcal{I}$. / $\mathcal{D}$. | 0.3296 | 0.8513 | 7.1089 | 0.0368 | 0.0789 | 0.7130 | 21.2594 | 0.0467 | 0.0080 | 0.0492 | 0.7180 | 26.9951 |
| TiNeuVox-B [9] | $\mathcal{I}$. / $\mathcal{D}$. | 0.3920 | 0.8627 | 7.2093 | 0.0290 | 0.1549 | 0.6873 | 21.0932 | 0.0462 | 0.0080 | 0.1294 | 0.7107 | 26.8620 |
| K-Planes [12] | $\mathcal{I}$. / $\mathcal{D}$. | 0.2982 | 0.8887 | 6.7960 | **0.0209** | 0.1218 | 0.7258 | 21.6203 | 0.0438 | 0.0076 | 0.1127 | 0.7364 | 27.4227 |
| **LiDAR4D (Ours)** | $\mathcal{I}$. / $\mathcal{D}$. | **0.2443** | **0.8915** | **6.7831** | 0.0258 | **0.0569** | **0.7396** | **21.7189** | **0.0426** | **0.0071** | **0.0459** | **0.7498** | **27.7977** |

Table 2. **Quantitative comparison on NuScenes dataset**. The notations are consistent with the KITTI-360 Table 1 above.

# Experiments

- **More Comparisons**

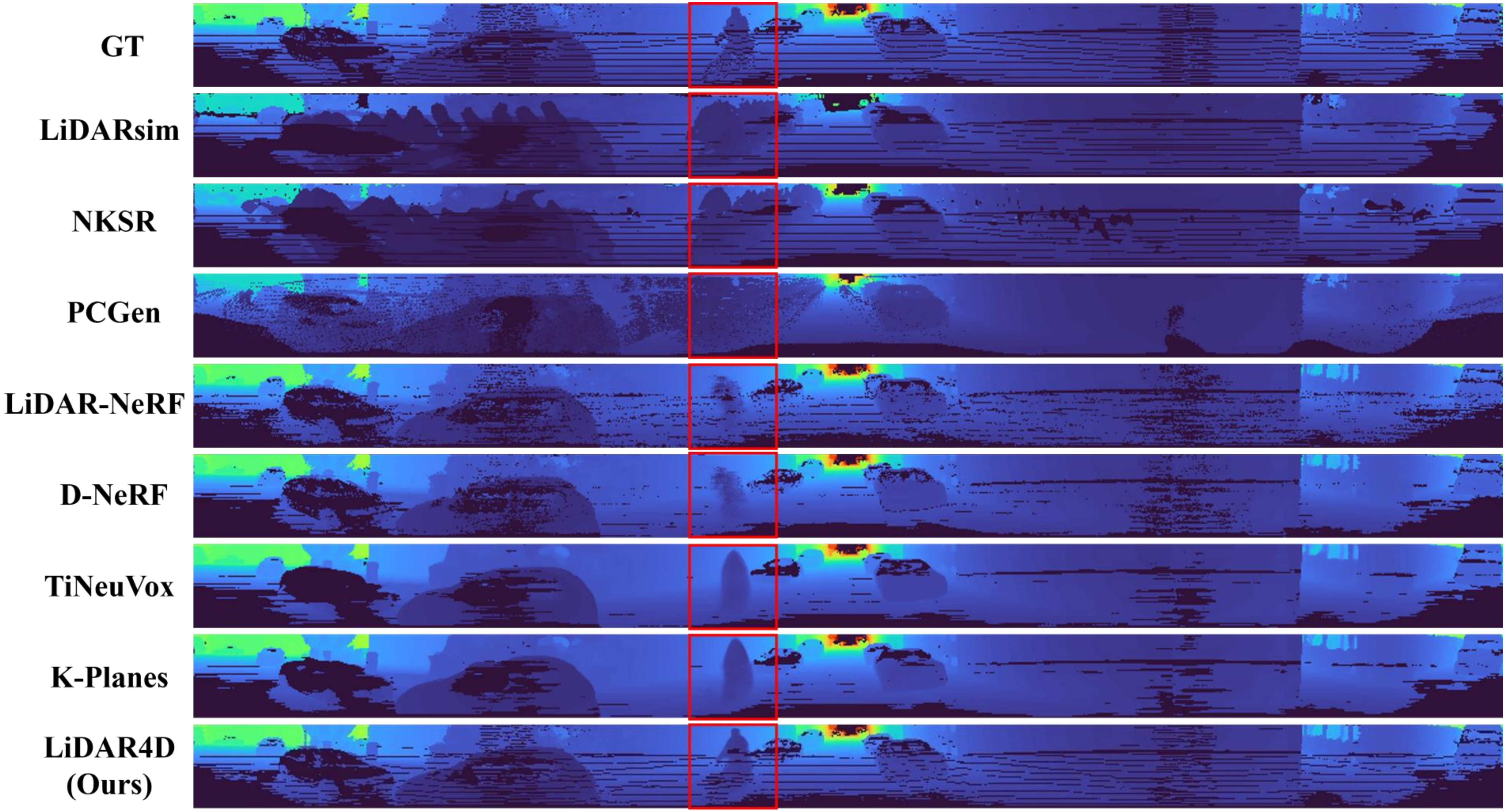
Depth reconstruction on dynamic vehicles



# Experiments

- **More Comparisons**

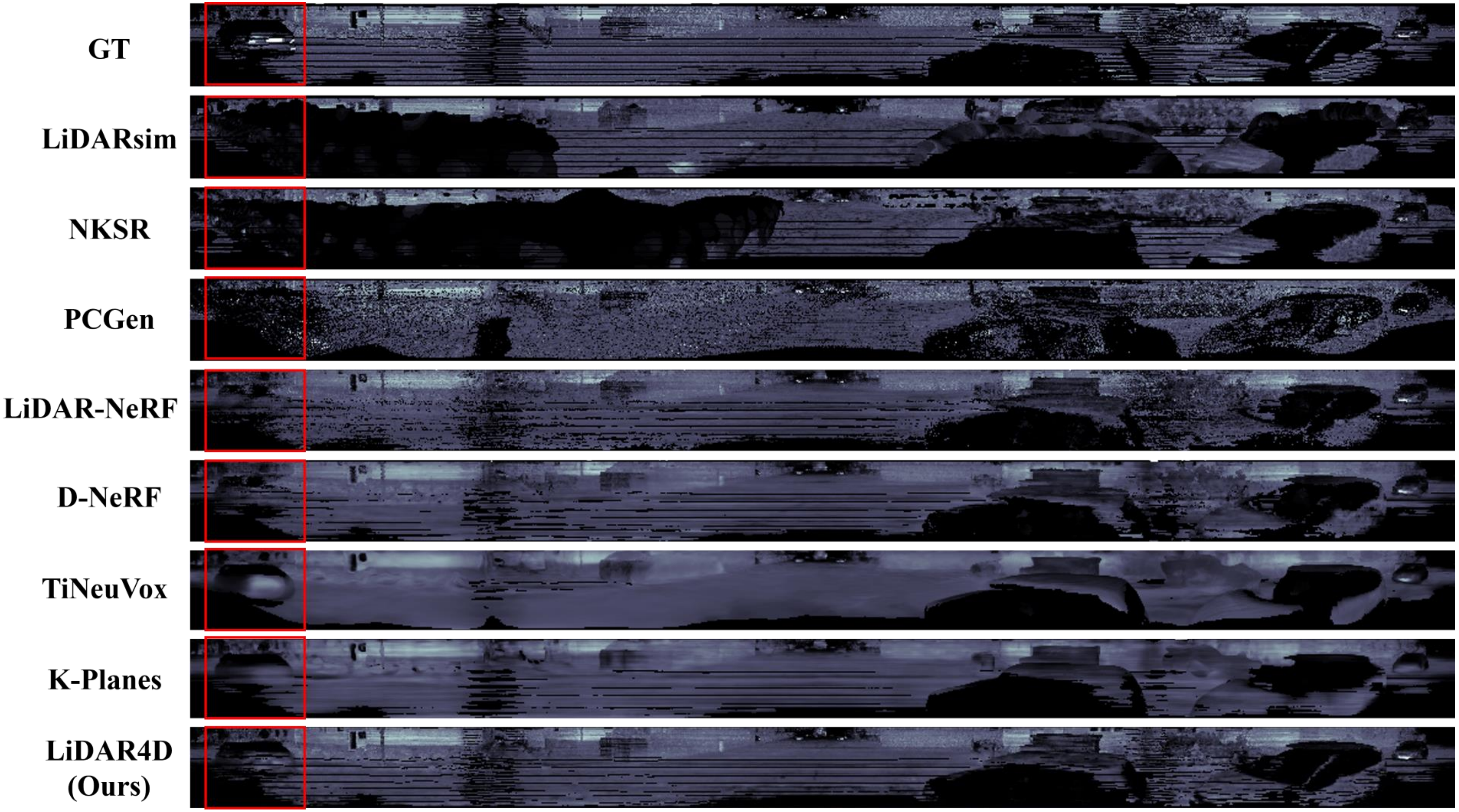
Even on small objects



# Experiments

- **More Comparisons**

Also the intensity reconstruction

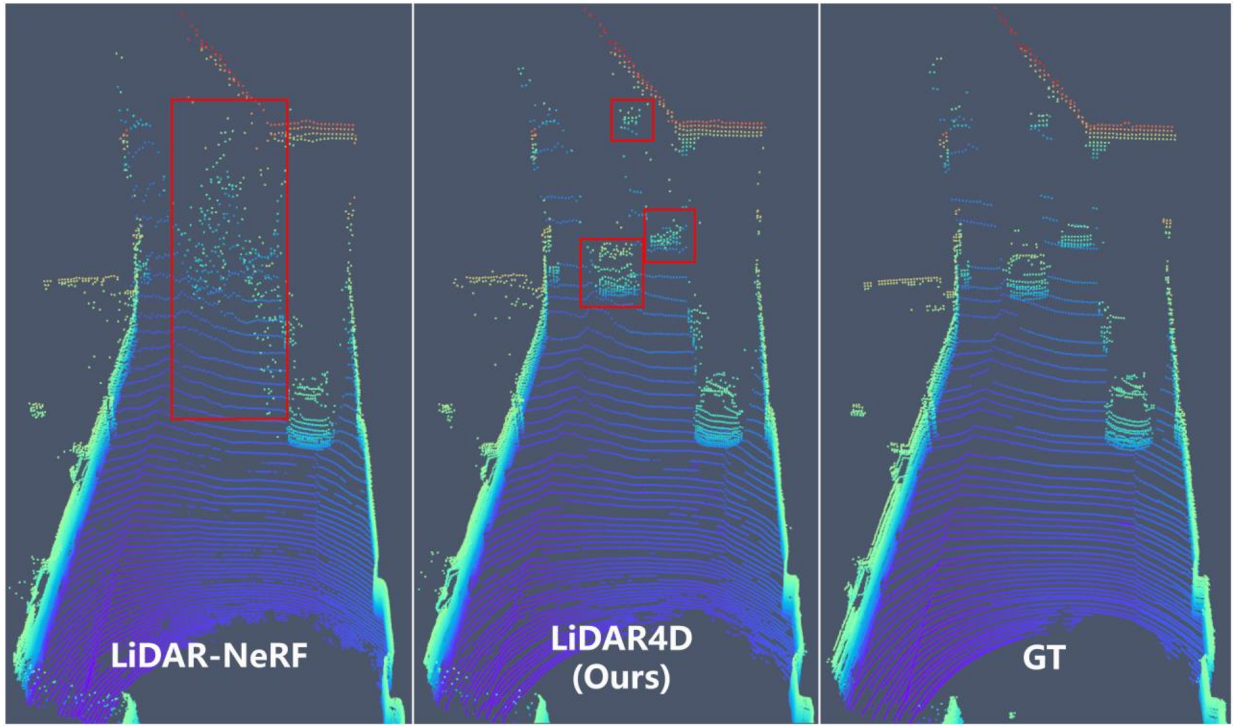


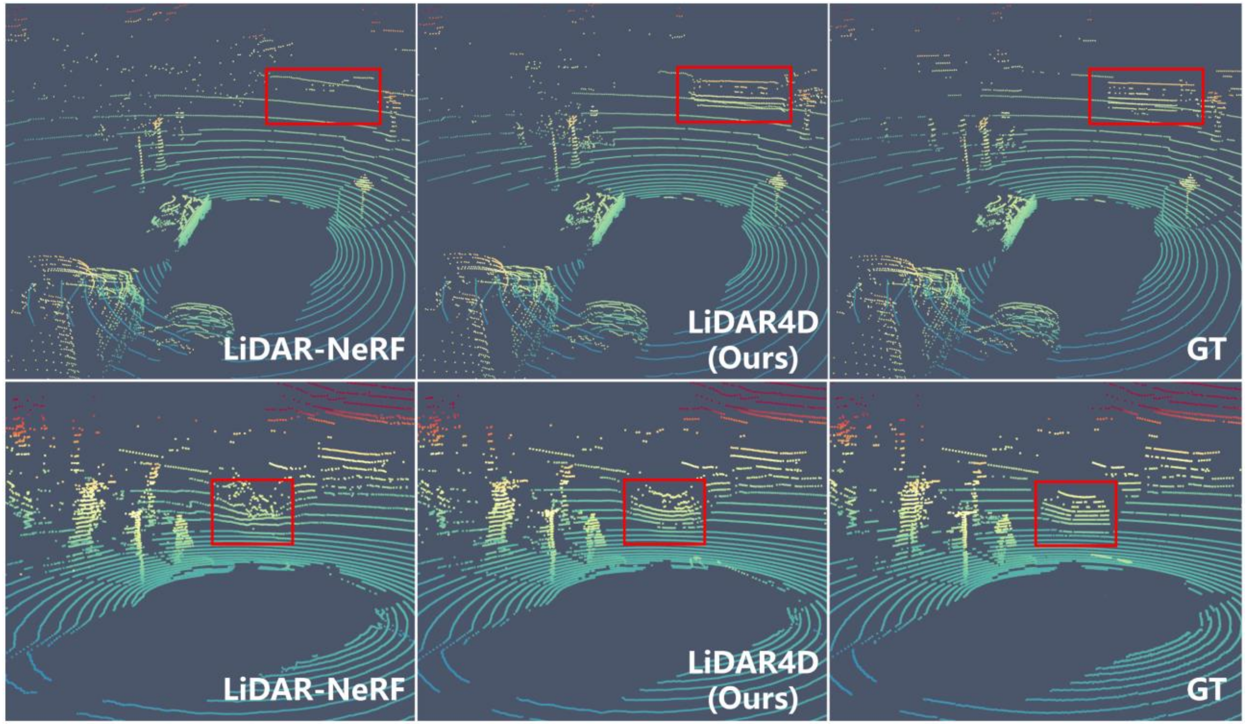
PRINCETON UNIVERSITY

# Experiments

- **More Comparisons**

✓ LiDAR4D achieves much better ***dynamic*** reconstruction results



# LiDAR4D

- Take advantages of **explicit** and **implicit** representations (hybrid one)
- Differentiable rendering for end-to-end optimization
- Geometry-aware and time-consistent reconstruction
- Without bounding box labeling of dynamic objects

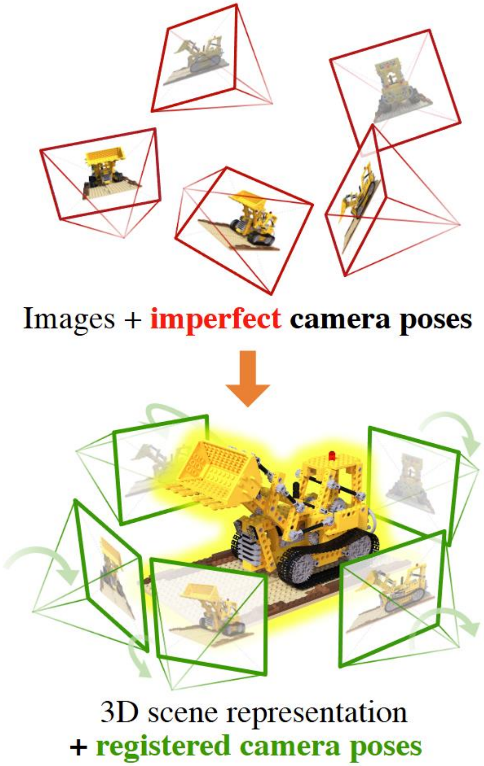
Minimal Human Supervision

*Can we further reduce the need for ground-truth, e.g., the sensor poses?*

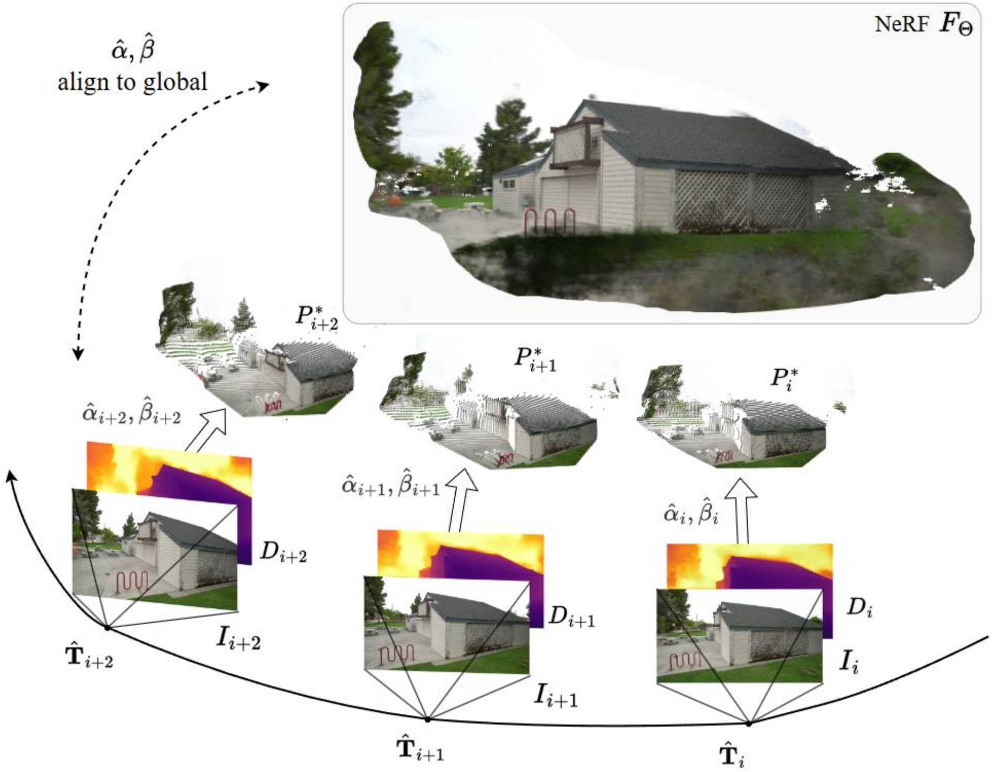
## Pose-free Reconstruction

- Related works in image reconstruction
- Domain gap between images and point clouds

**Geometry Guidance**




BARF: Bundle-Adjusting Neural Radiance Fields



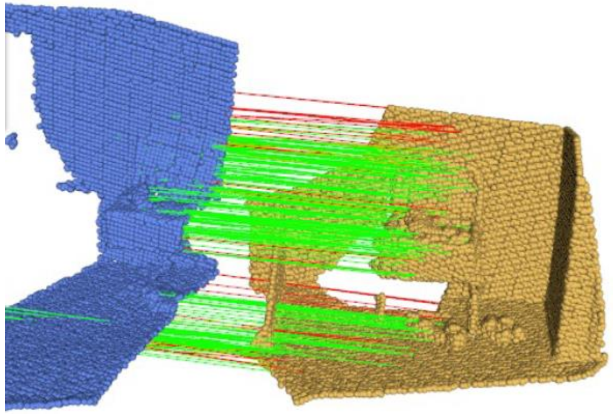

NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior

PRINCETON UNIVERSITY

## ■ Point Cloud Registration

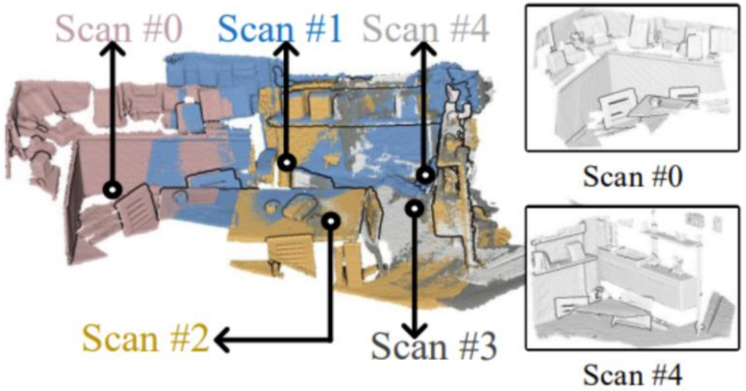
- Poor Generalization
- Trapped in Local Optima
- Error Accumulation

**Global Optimization**



*Pair-wise*

Geometric Transformer for Fast and Robust Point Cloud Registration

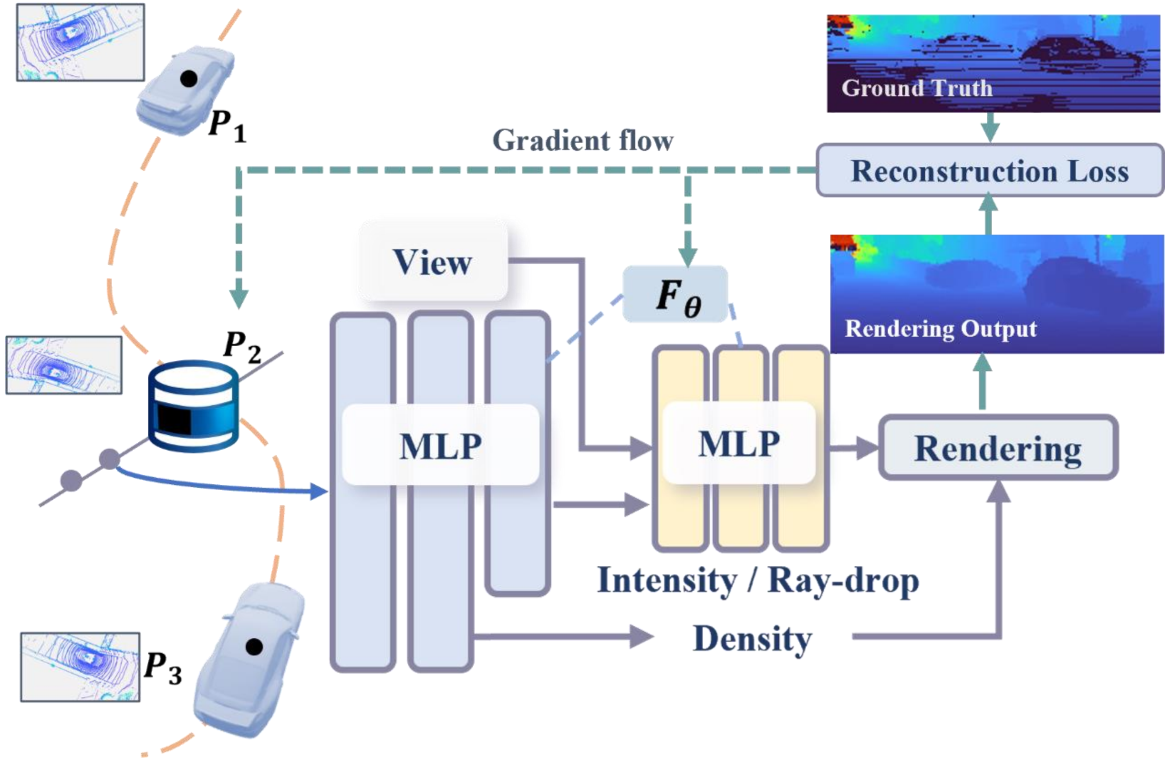



*Multi-view*

Robust Multiview Point Cloud Registration with Reliable Pose Graph Initialization and History Reweighting

# Method - GeoNLF

- Gradient flow to Poses
- Bundle Adjustment
- Global Optimization

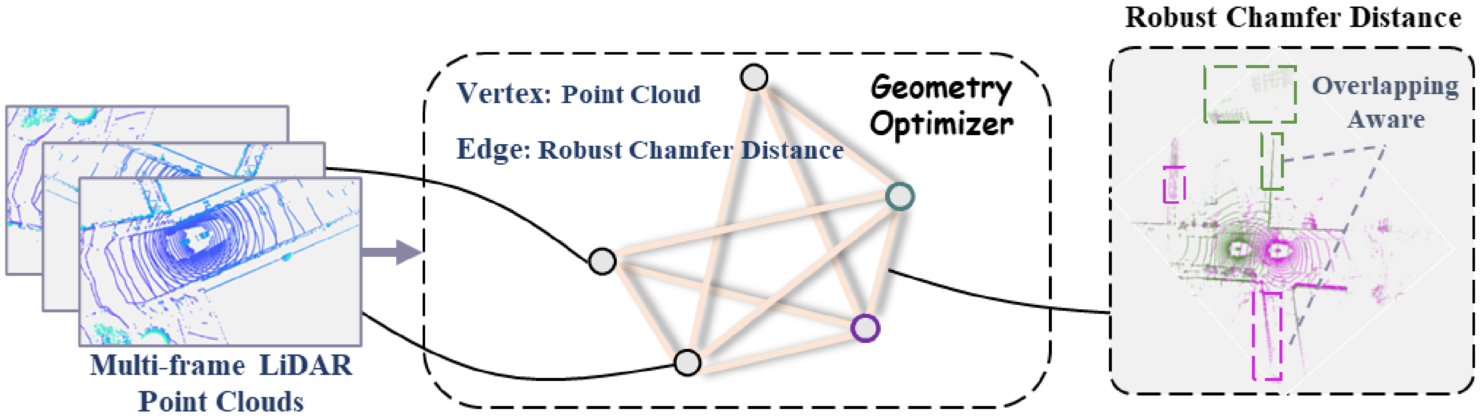



*However, optimizing the geometry and poses simultaneously is very tricky*

# Method - GeoNLF

## Introduce Geometry Guidance

- Graph-based Optimization




- Selective-Reweighting Strategy

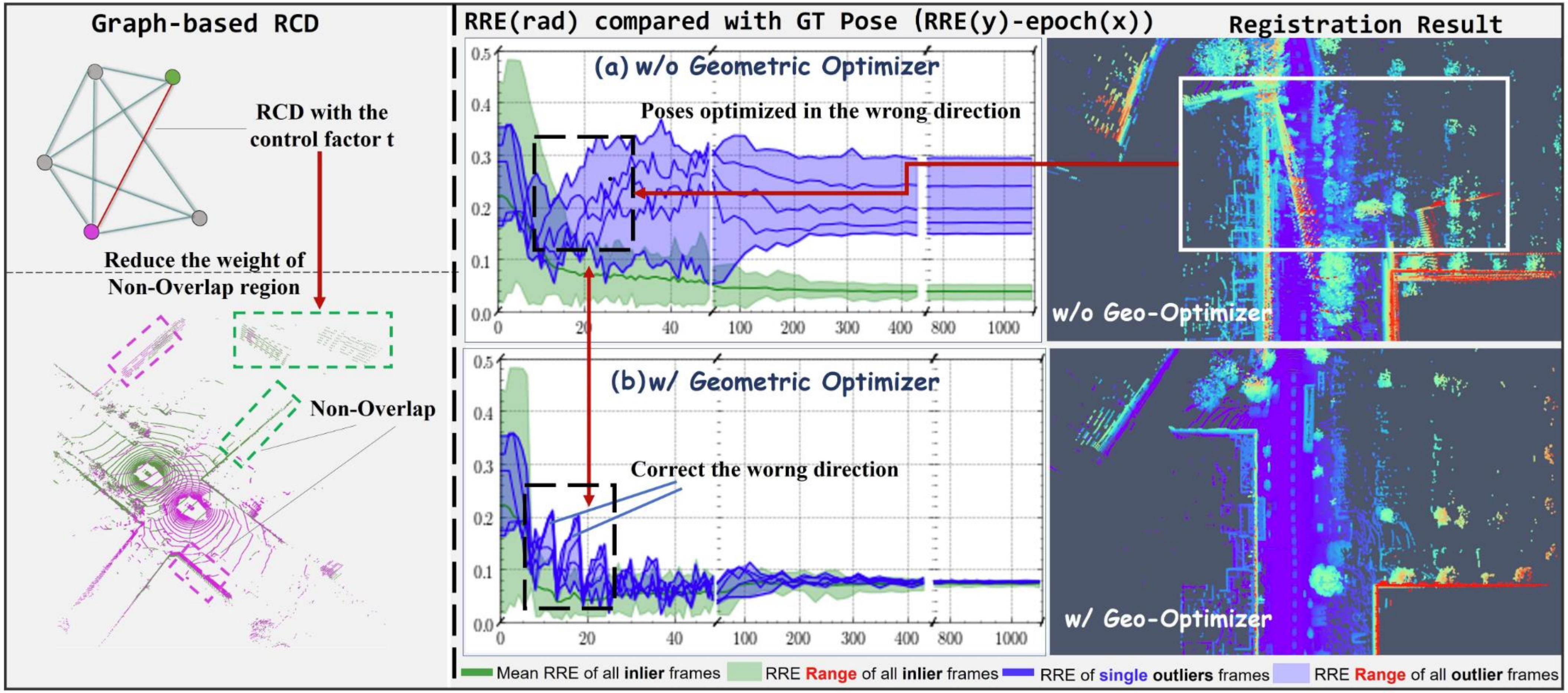
decrease the learning rate of top-k frames with highest rendering losses

- Geometry Constraints

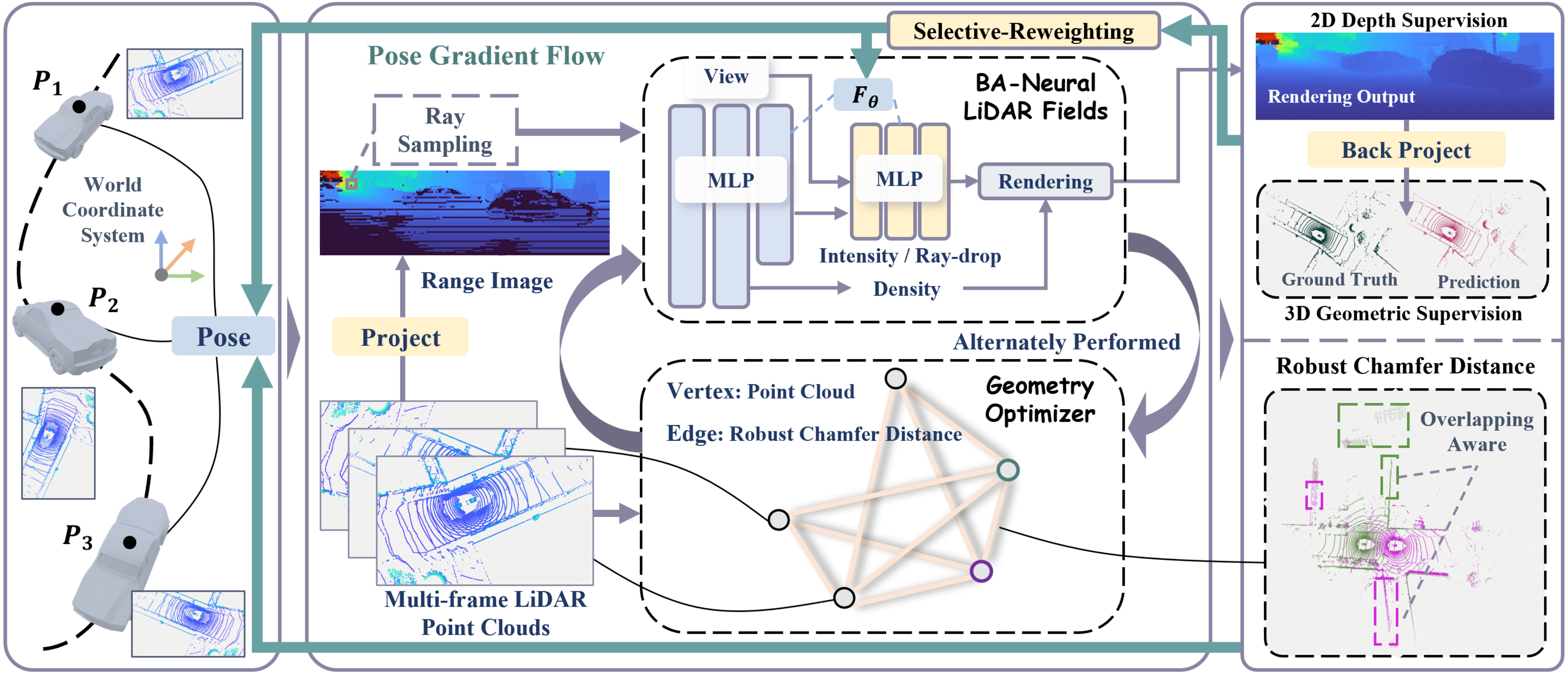
Chamfer Distance and Normal loss of rendered point clouds

# Method - GeoNLF

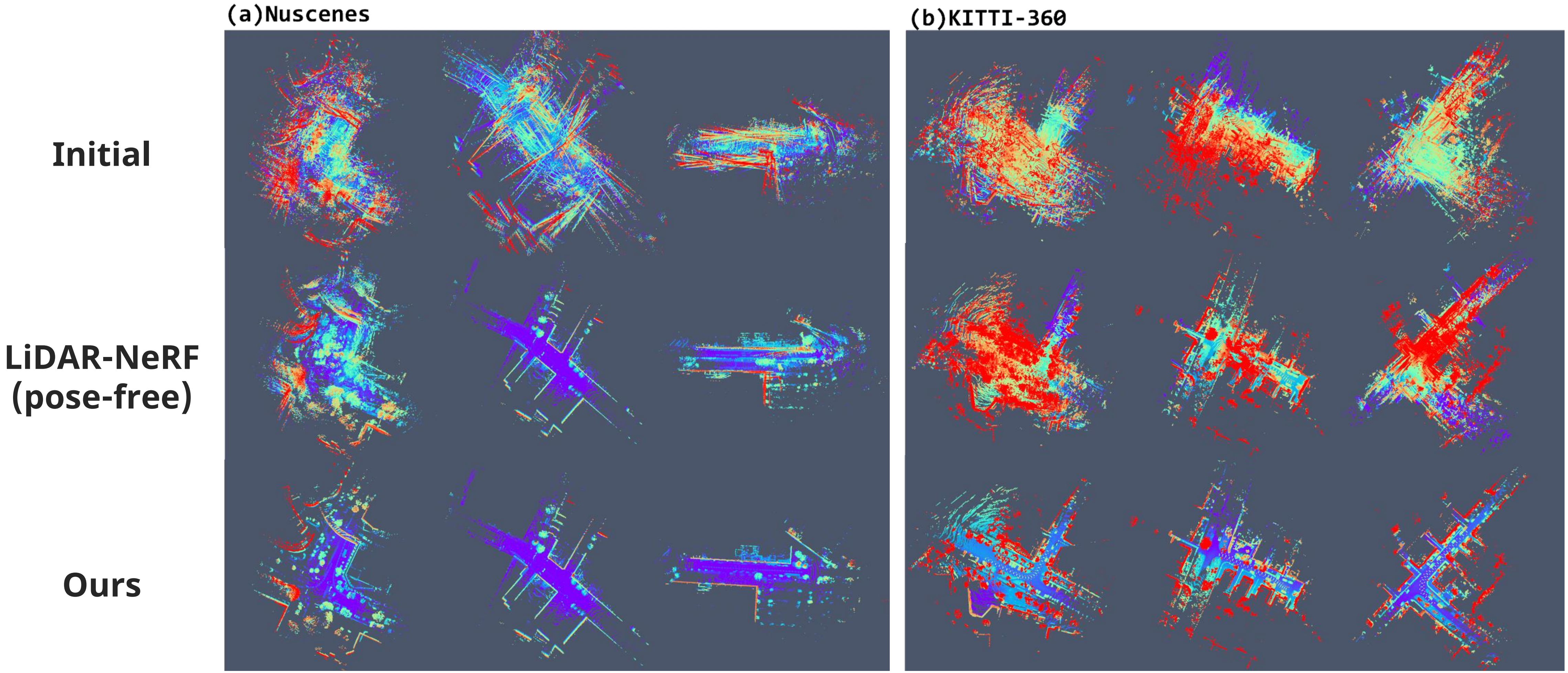
**Introduce Geometry Guidance**



# Method - GeoNLF



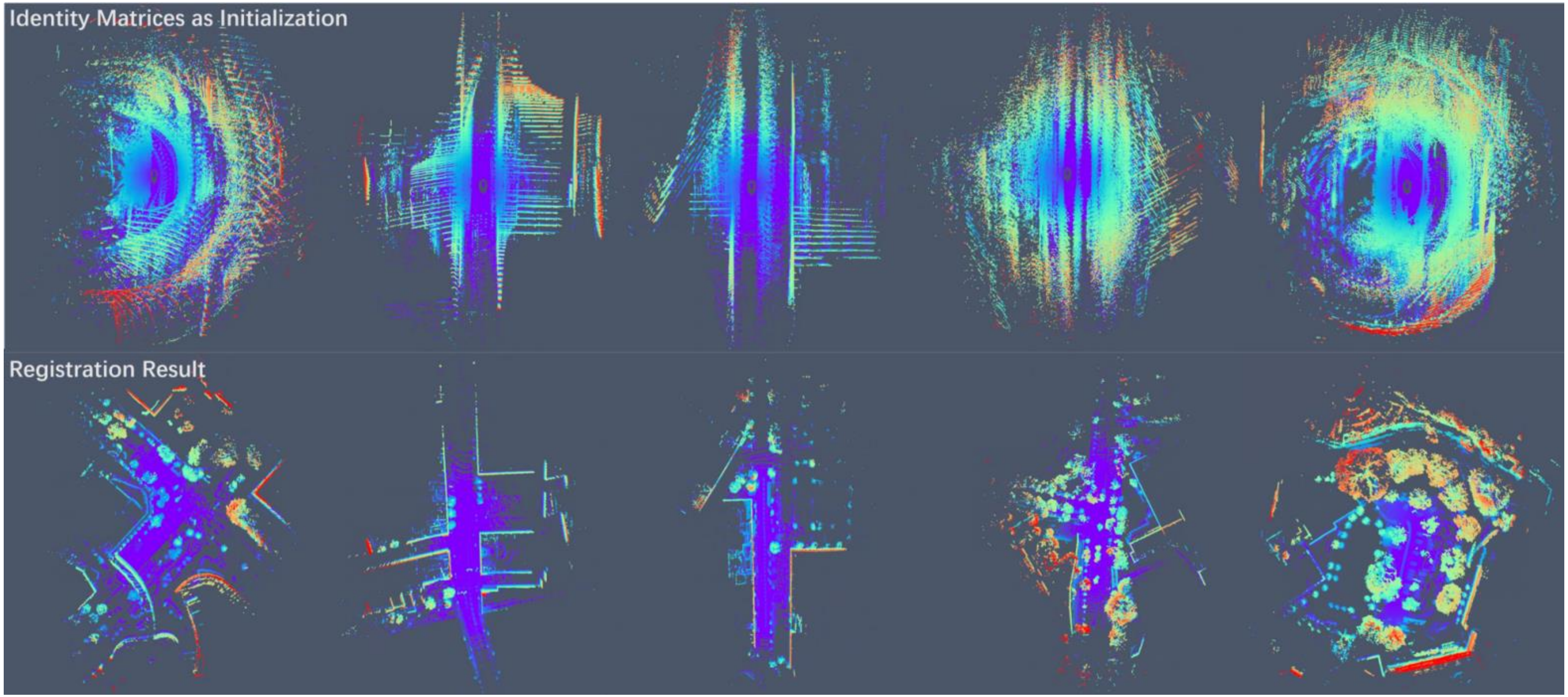
# Pose Initialization with Random Perturbation



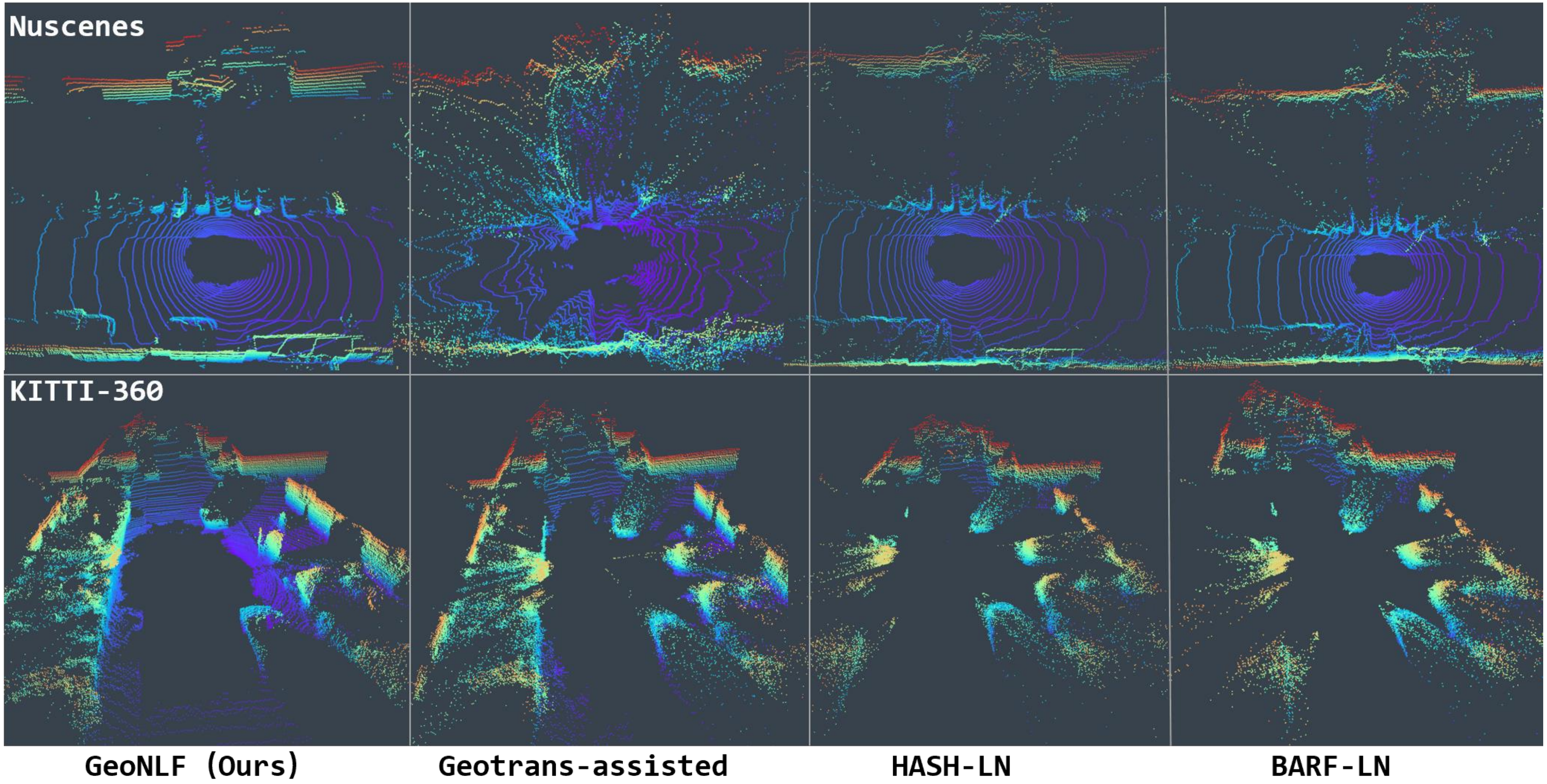
# Pose Initialization with Identity Matrix

Initial

Result (Ours)



# Pose-free Reconstruction (NVS)



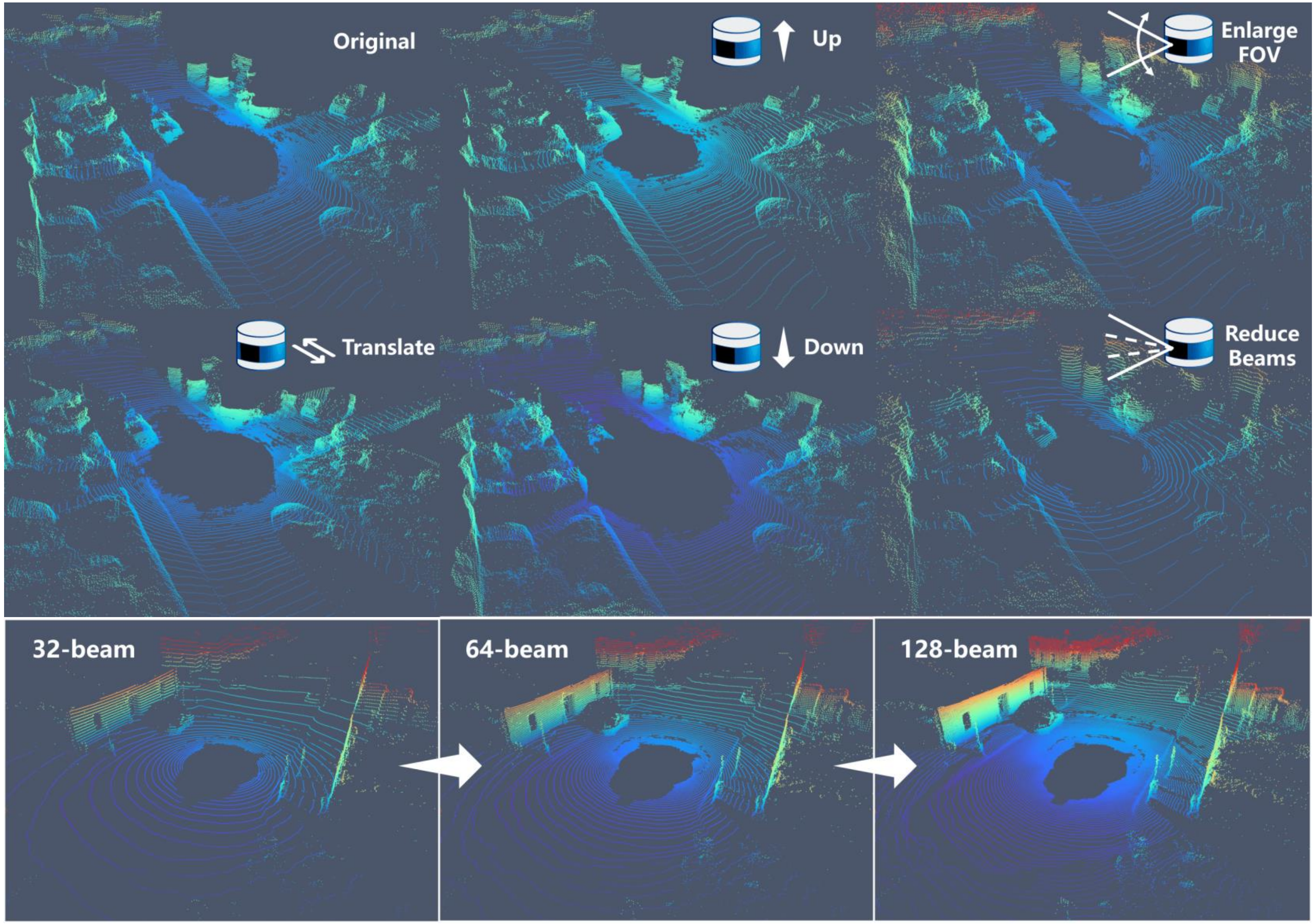
# GeoNLF

- Reduce the dependence of accurate **poses** for reconstruction
- **Global** robust optimization with **geometry guidance**
- Simultaneous point cloud **registration** and **reconstruction**

*What can we do after the reconstruction?*

# Simulation

- **Shift poses**
  - Sensor Height
  - Translation / Rotation
- **Configuration**
  - Field of View
  - Angular resolution
  - LiDAR beams
- **Dynamics**
  - Scene Re-play
  - Novel Trajectory



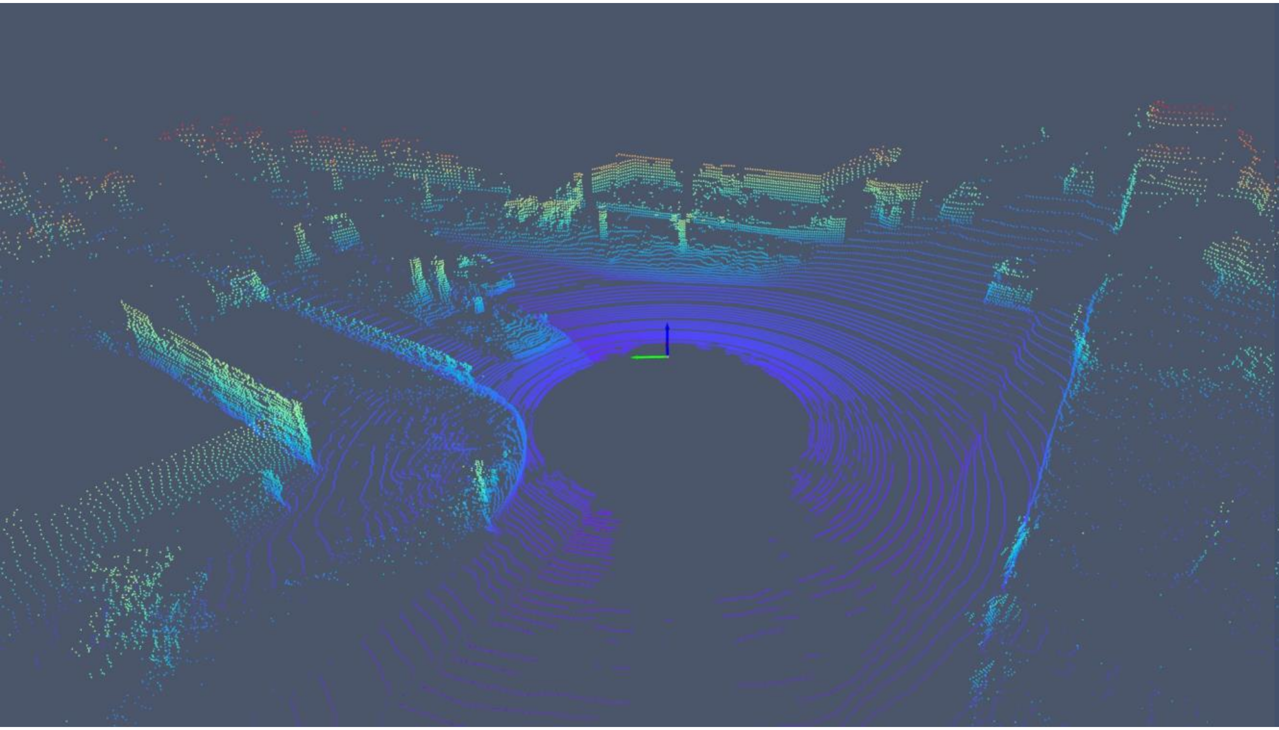
# Simulation

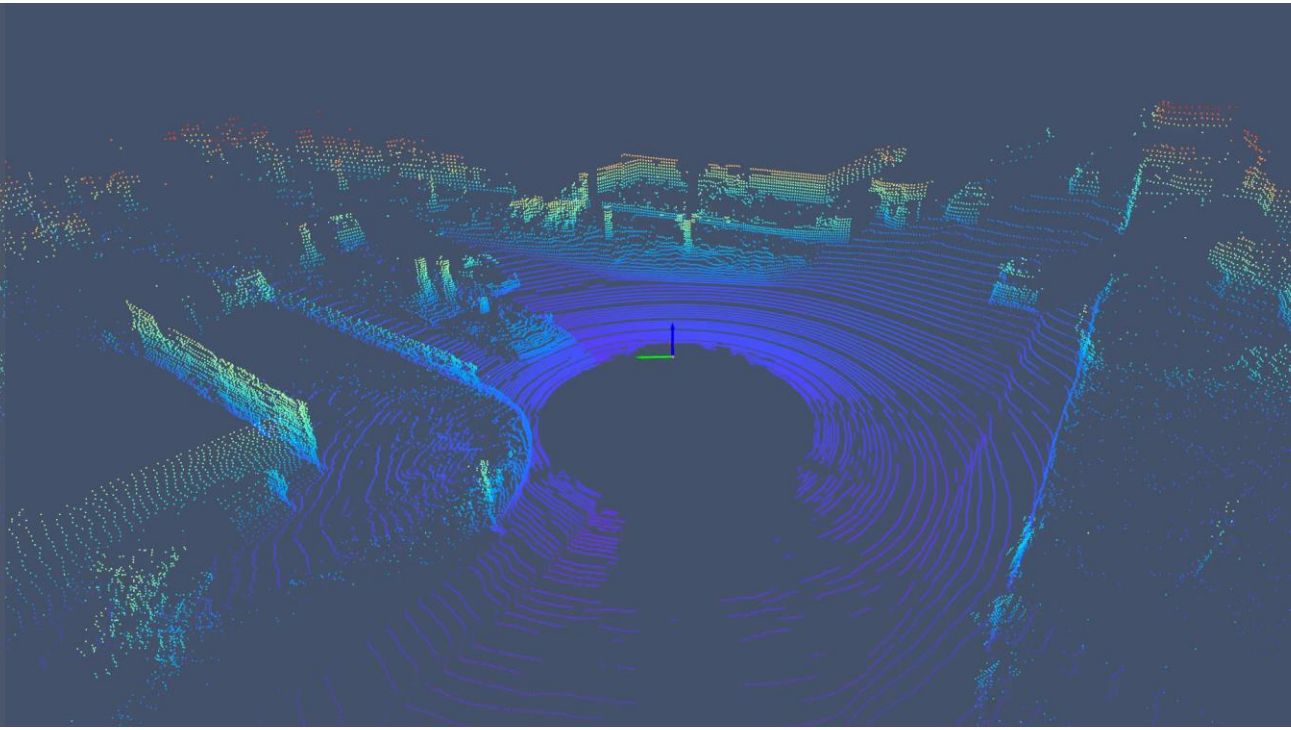
Original Placement

simulate

Horizontal / Vertical

Displacement



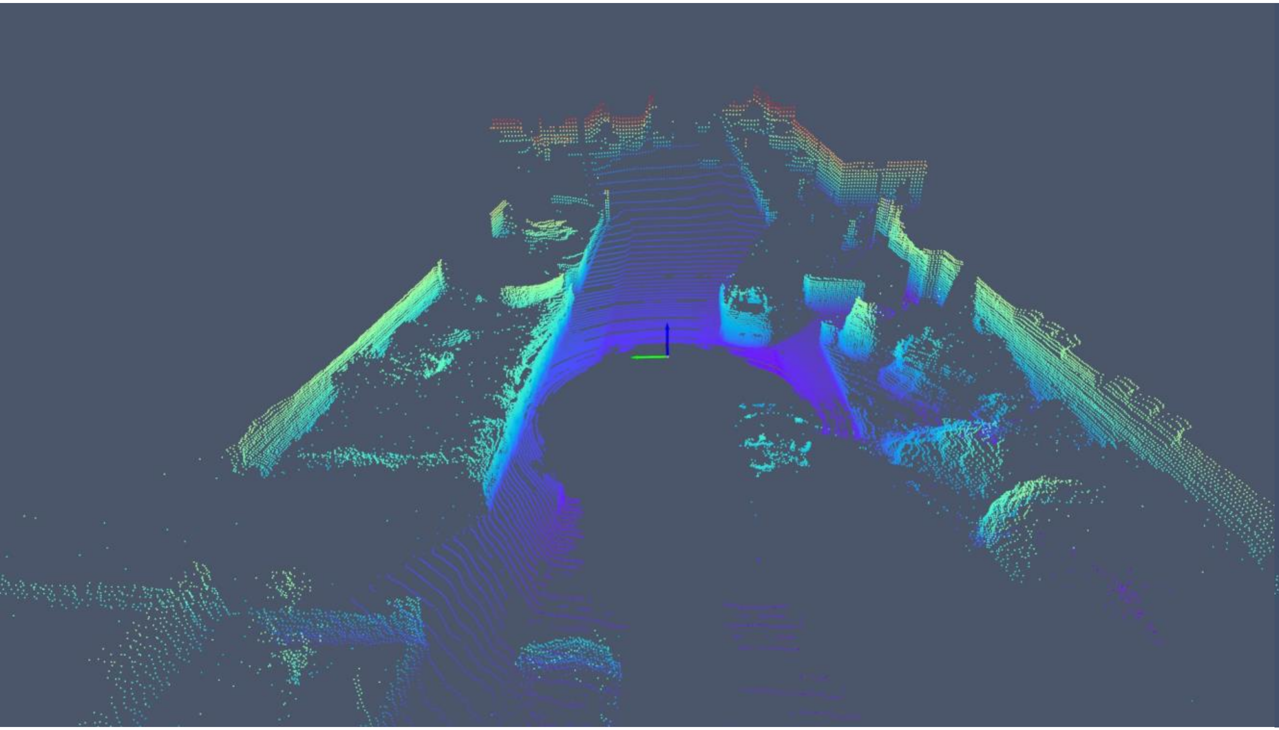


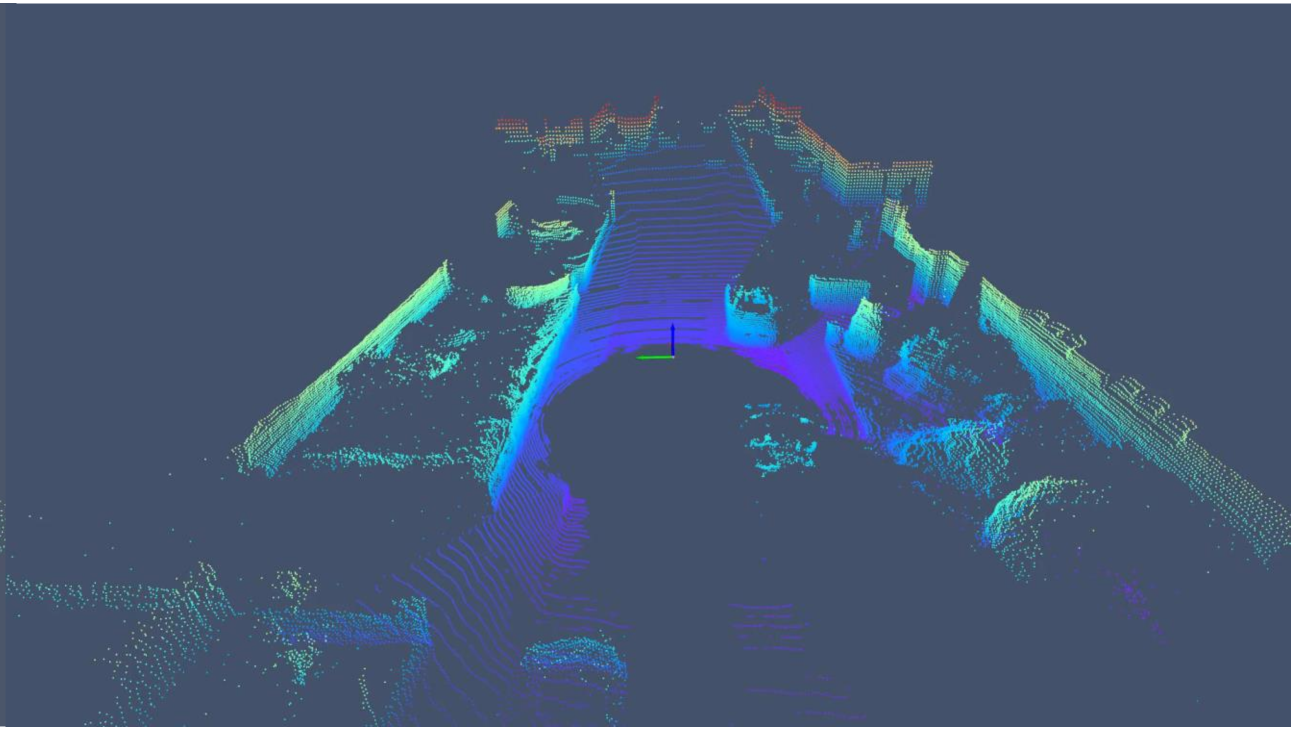
# Simulation

Original Field of View

simulate

Enlarge / Reduce Field of View



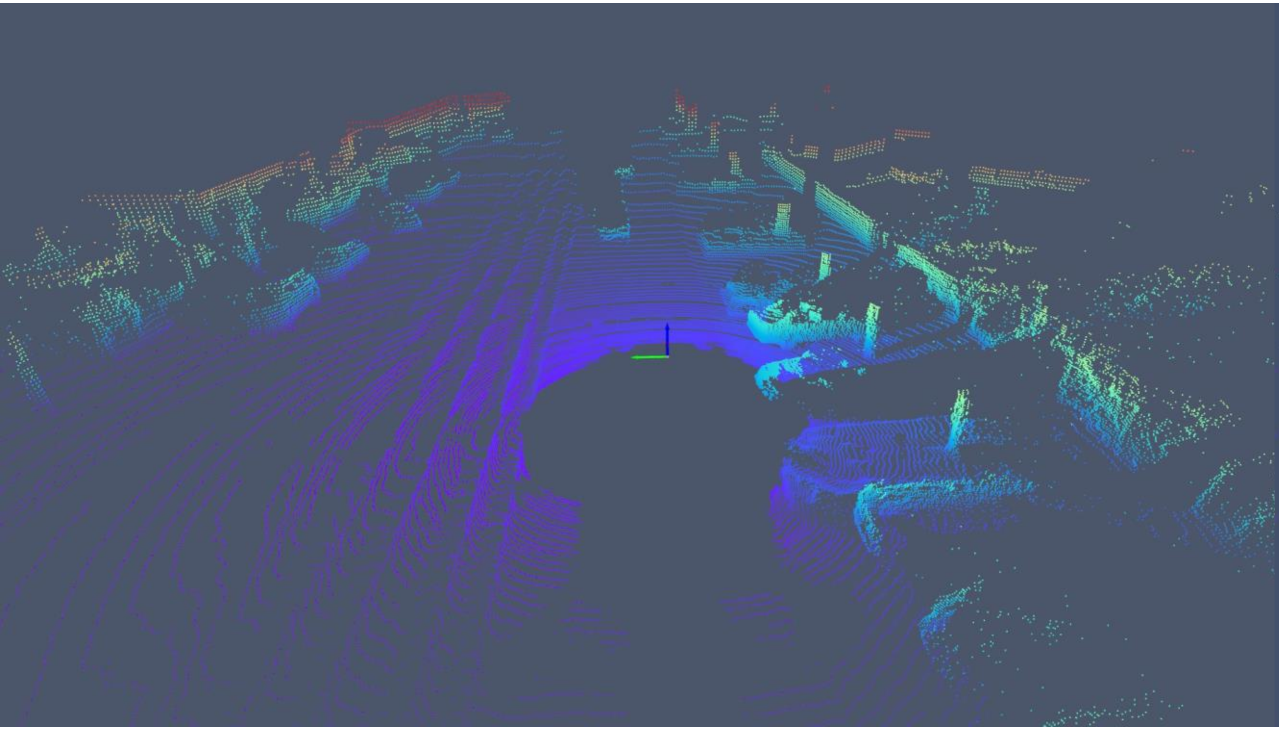


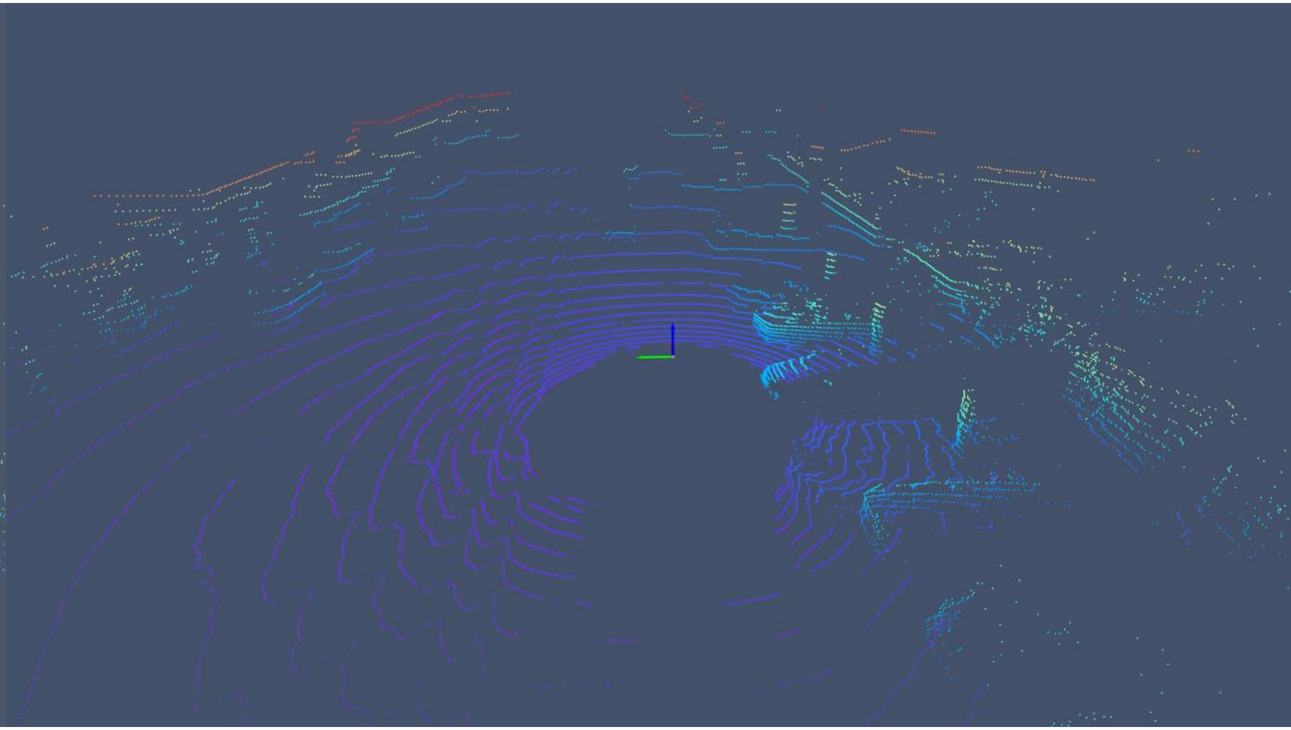
# Simulation

Original Beams

simulate

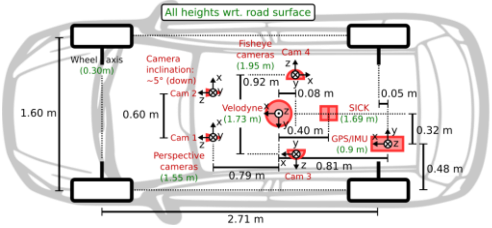
Increase / Decrease Beams





# Simulation

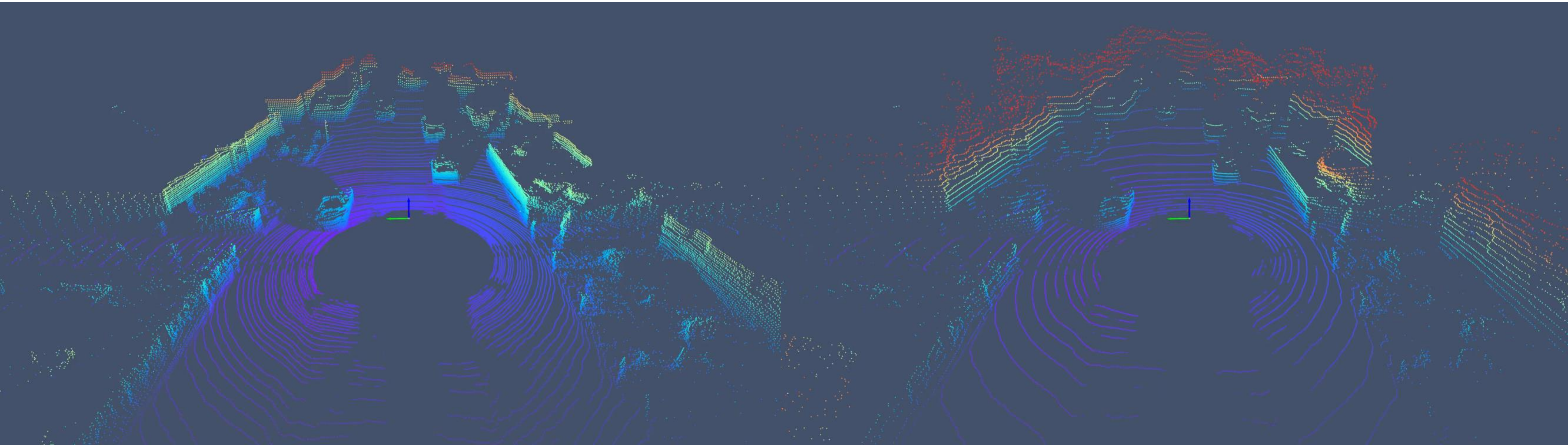
## KITTI-360 LiDAR Configuration

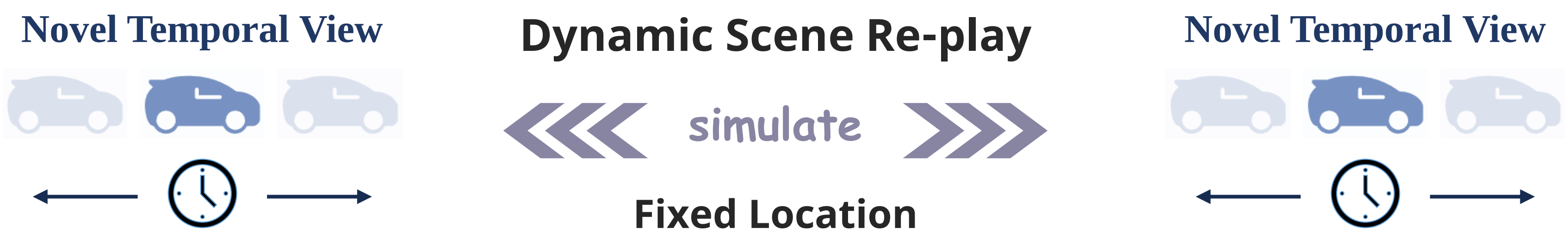

simulate

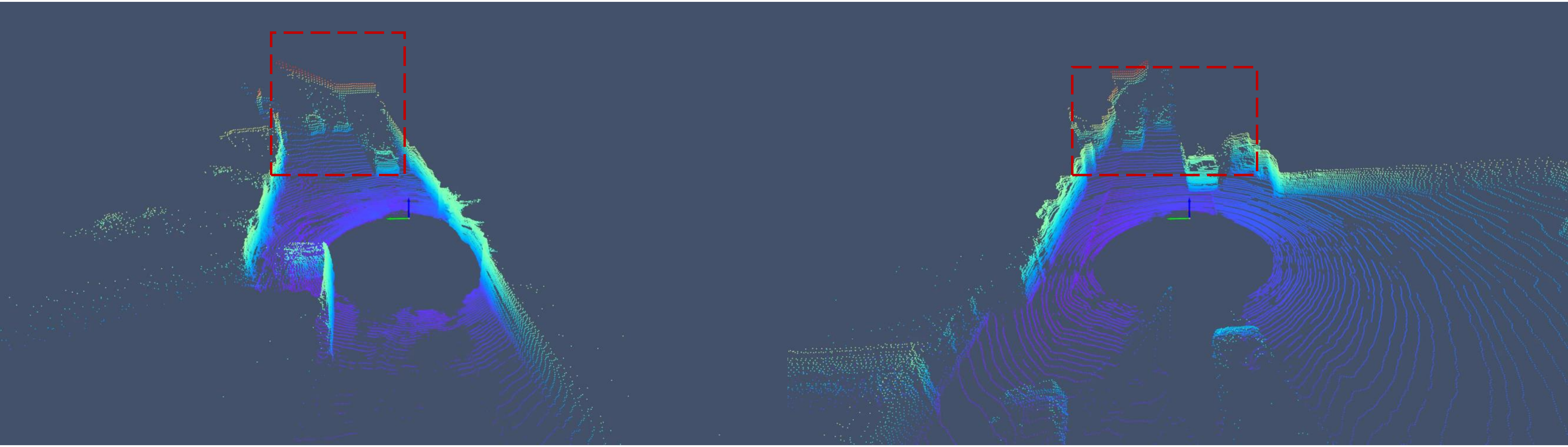
## NuScenes LiDAR Configuration


FOV,
Height,
Beams,
Range...



# Simulation


Novel Temporal View
Dynamic Scene Re-play
simulate
Fixed Location
Novel Temporal View

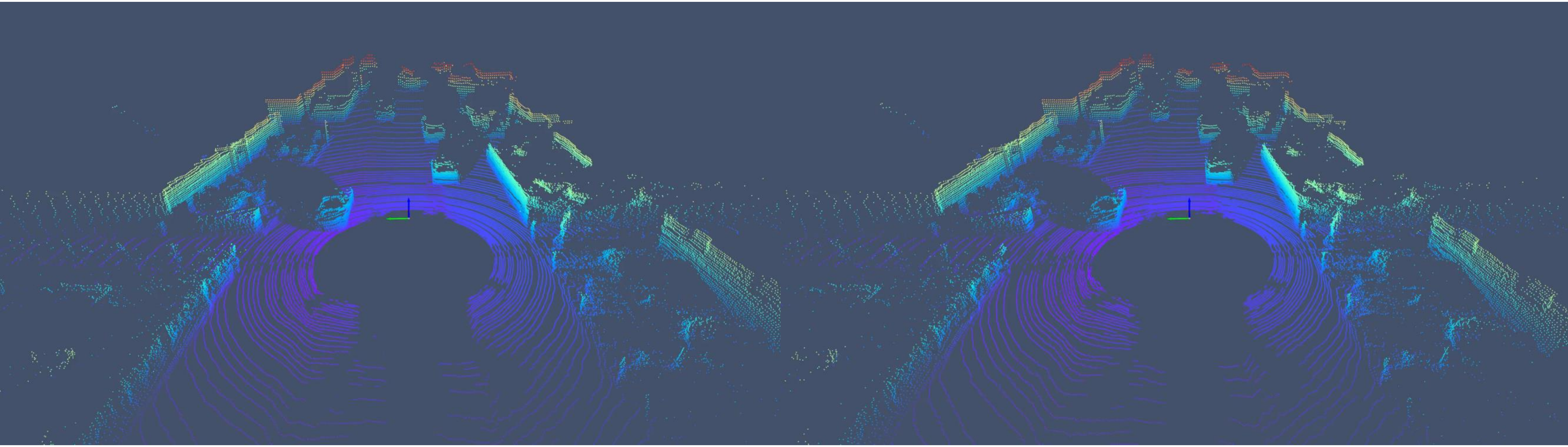


# Simulation

Original Trajectory

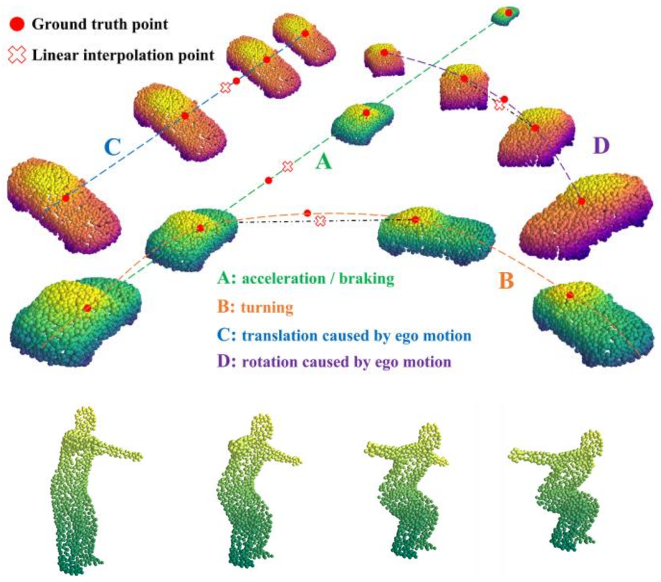
simulate

Novel Trajectory



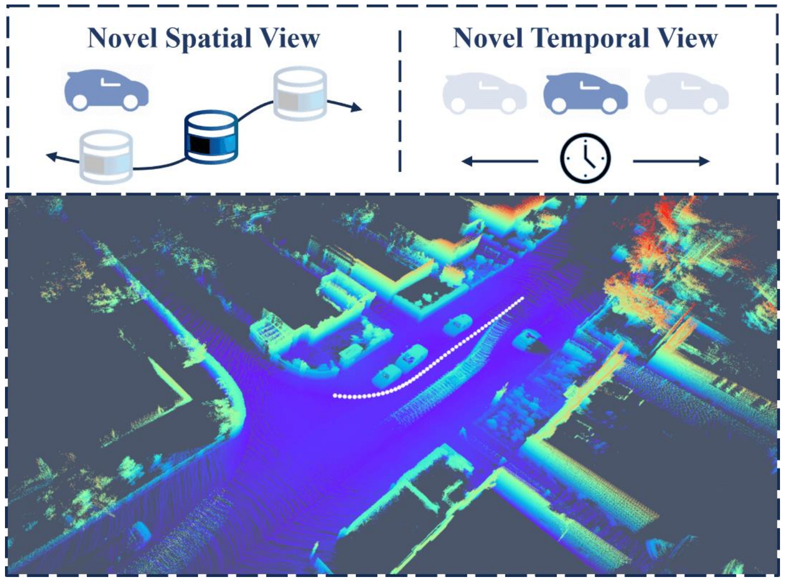
# *Dynamic Reconstruction*

## *Flow Optimization*



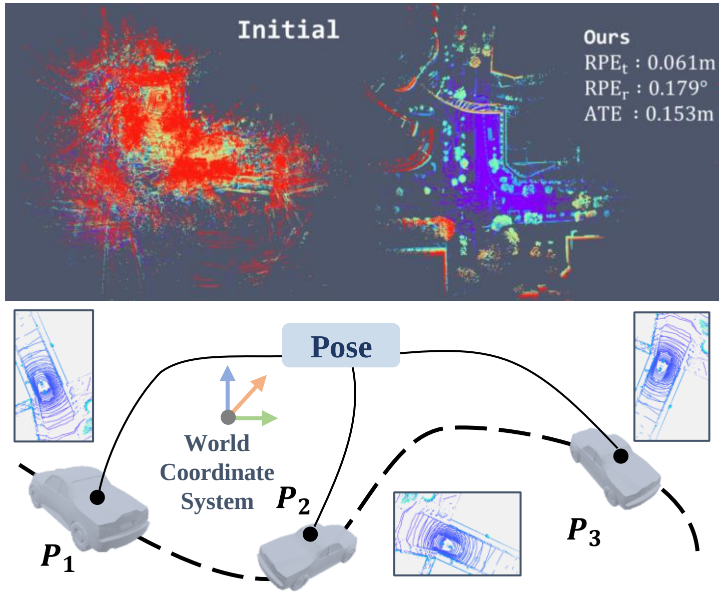

**NeuralPCI**
**[CVPR'23]**

## *Novel View Synthesis & Simulation*




**LiDAR4D**
**[CVPR'24]**

## *Pose Optimization*




**GeoNLF**
**[NeurIPS'24]**

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

SEATTLE, WA CVPR JUNE 17-21, 2024

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS 2024

# Summary

- Representation matters
- Minimal human supervision
- Combine optimization, reconstruction and simulation
- Generative priors in the future

# Thank you for listening

# Q&A