NeuralPCI: Spatio-temporal **Neural** Field for 3D **P**oint **C**loud Multi-frame Non-linear **I**nterpolation

Zehan Zheng, Danni Wu, Ruisi Lu, Fan Lu, Guang Chen, Changjun Jiang

ISPC Lab, Tongji University

TUE-AM-086

Project Page: https://dyfcalid.github.io/NeuralPCI

Point Cloud Interpolation

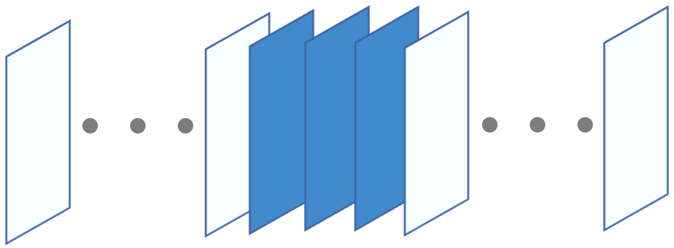
*Let's take a glance at **NeuralPCI** first!*

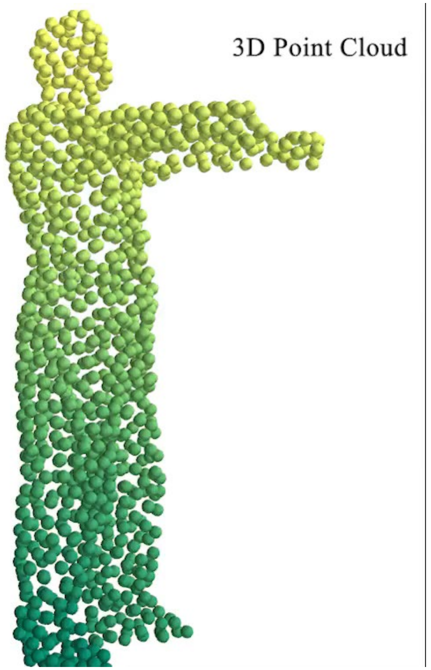
Given the point cloud sequence $\{P_0, P_1, \ldots, P_M\}$, $P_i \epsilon \mathbb{R}^{N\times 3}$

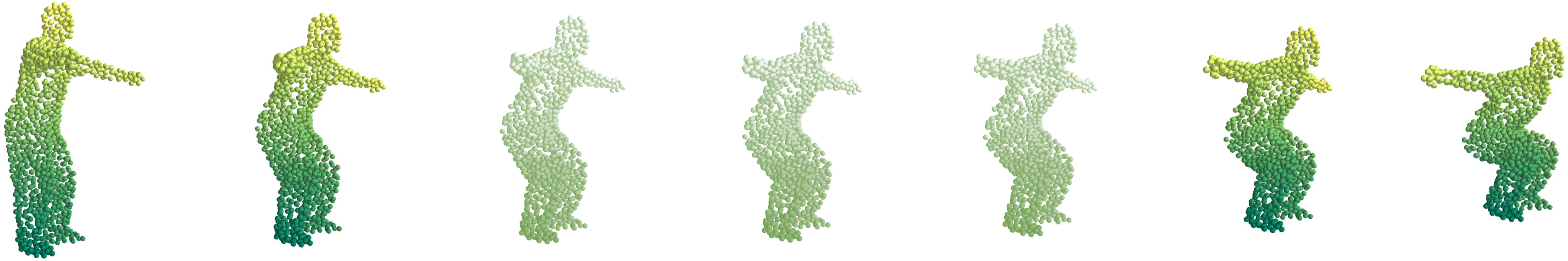
Interpolate $\boldsymbol{k}$ frames

Low Frame Rate → **High** Frame Rate

between every two frames







Background & Challenge

- **Hardware Limitation**

  The frame rate of the 3D Scanner / LiDAR is limited (i.e. 10-20Hz)

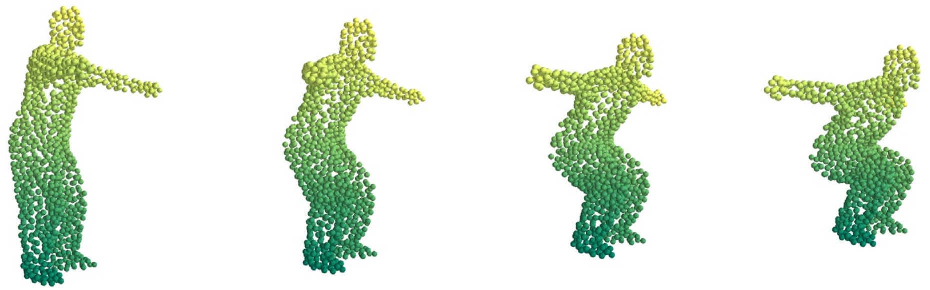
- **Point Cloud Structure**

  Irregular, unordered, and hard to find correspondences between frames

- **Nonlinear Motion**

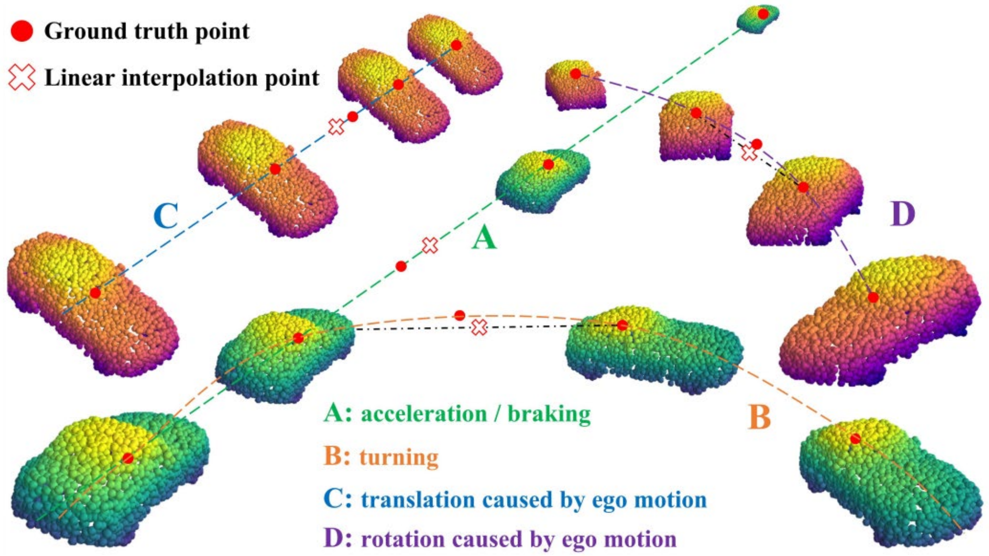
  Large amount of nonlinear complex motion exists in the real-world scenarios (i.e. dynamic human / vehicle motions)

Nonlinear Interpolation is the key



Interpolation is **needed**

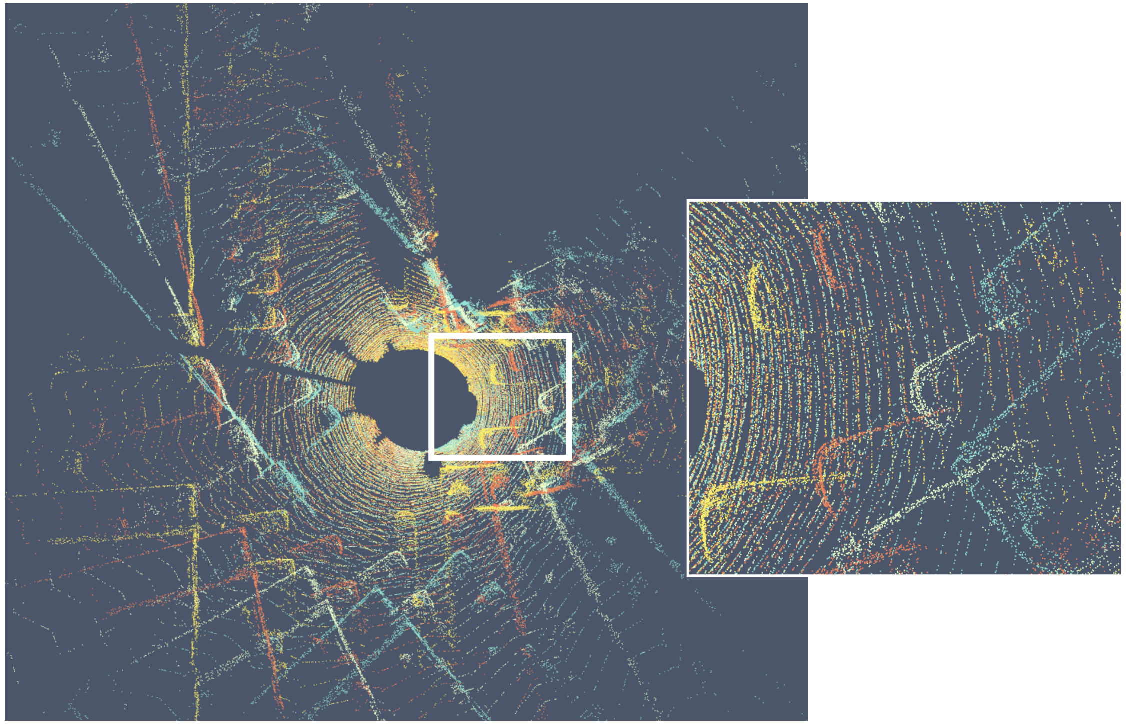
Can't **directly** interpolate



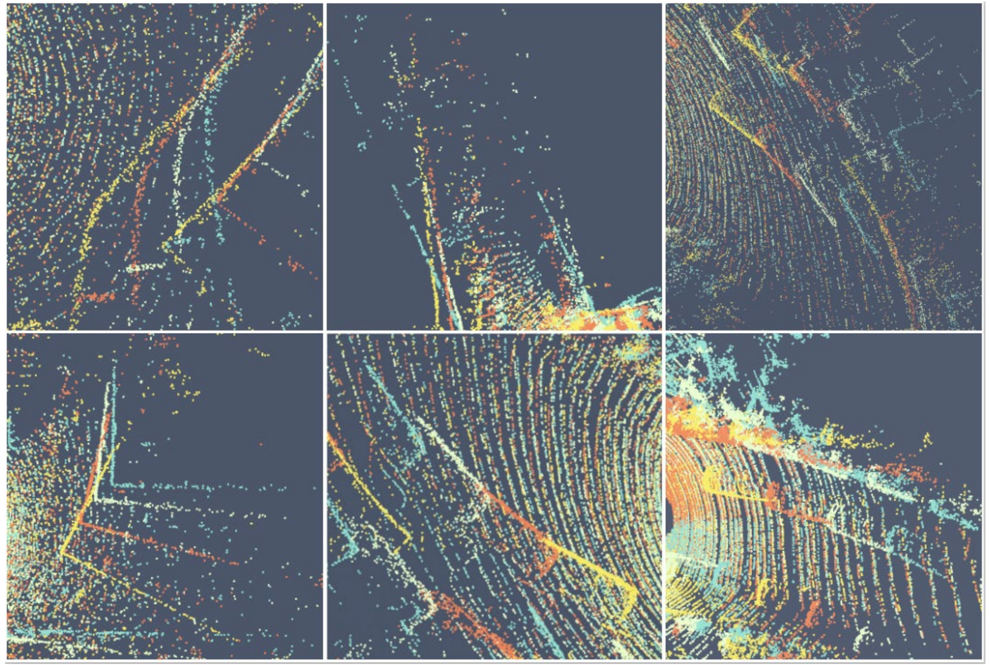
Background & Challenge

- **Nonlinear Motion**

More examples ...



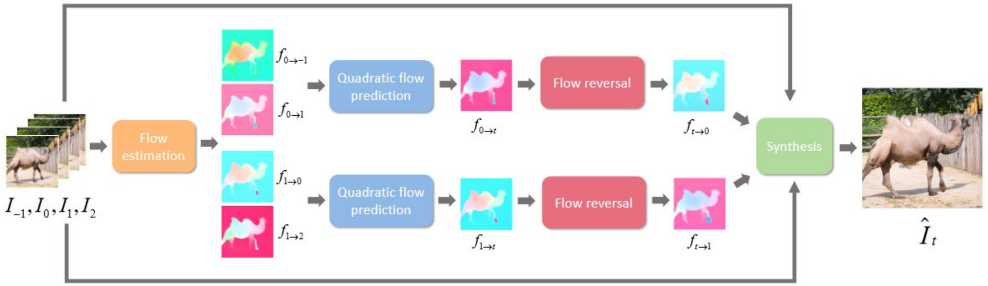
We construct the NL-Drive Dataset



Related Work & Motivation

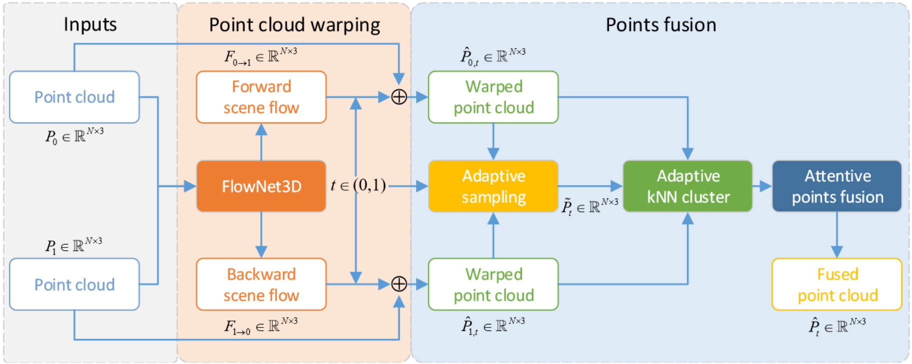
- **Video Frame Interpolation**

  Super SloMo / QVI / RIFE ⟶ optical flow based



- **Point Cloud Interpolation**

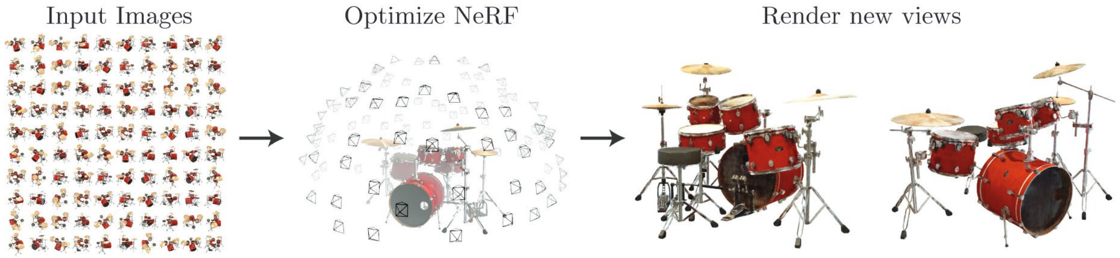
  PointINet / IDEA-Net ⟶ scene flow based



- **Neural Field** (*Implicit parameterization*)

  NeRF ⟶ take images of different views as input and render new images of unknown views

  We can view NeRF as an interpolation

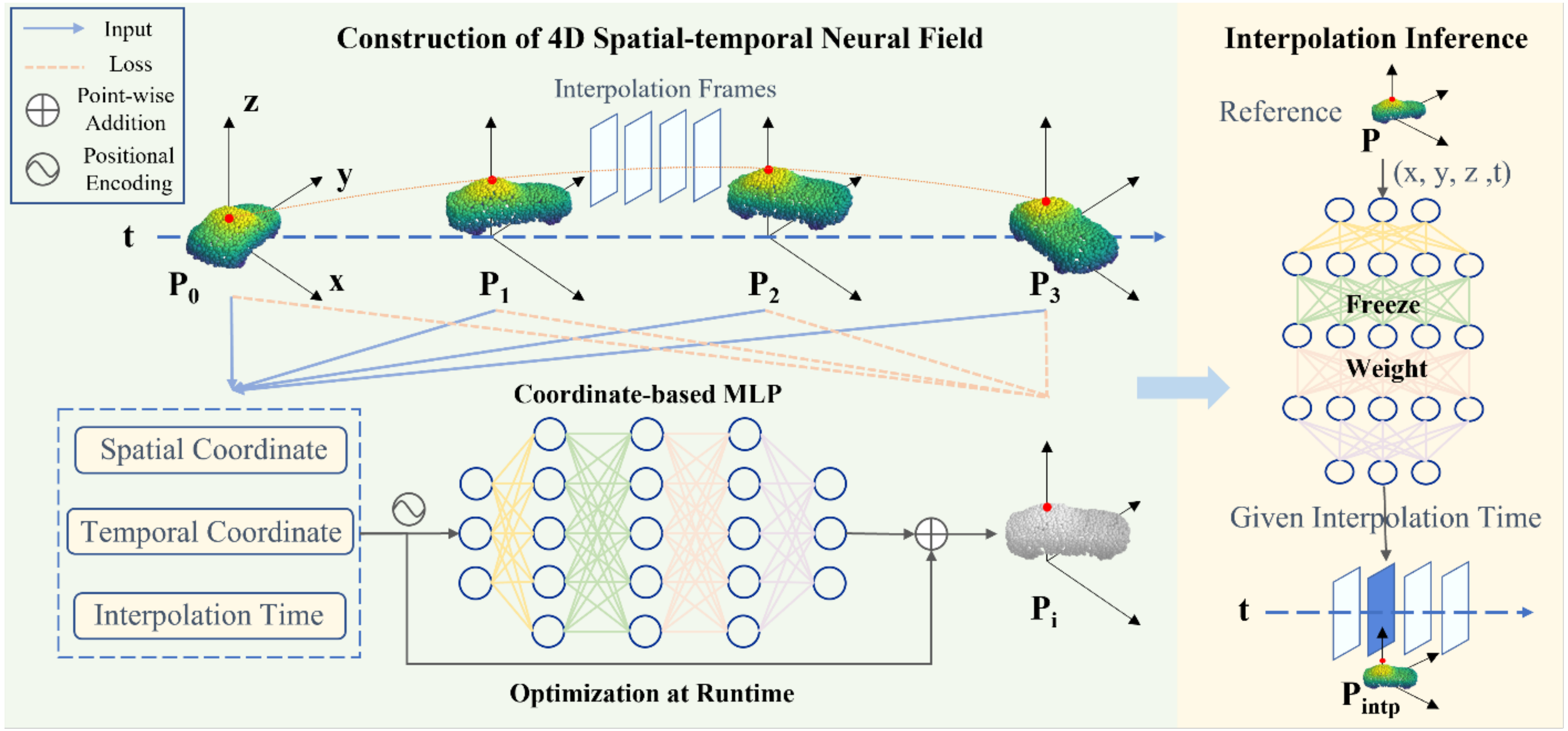


Multiple point cloud frames as input; Construct 4D Spatio-temporal neural field

Method

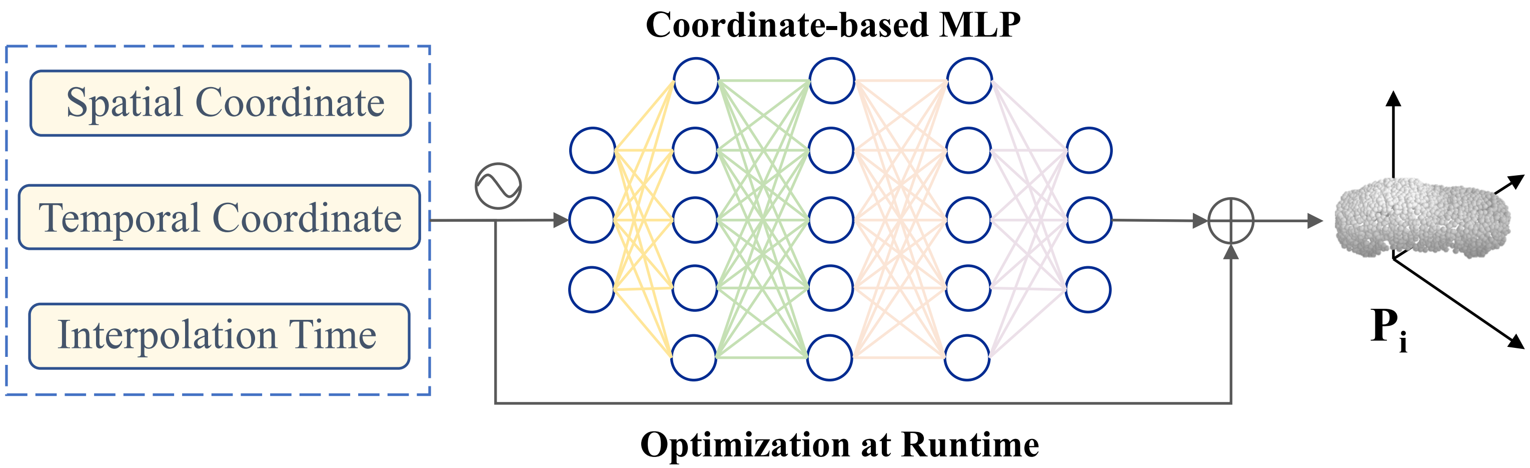
NeuralPCI

- ✓ 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



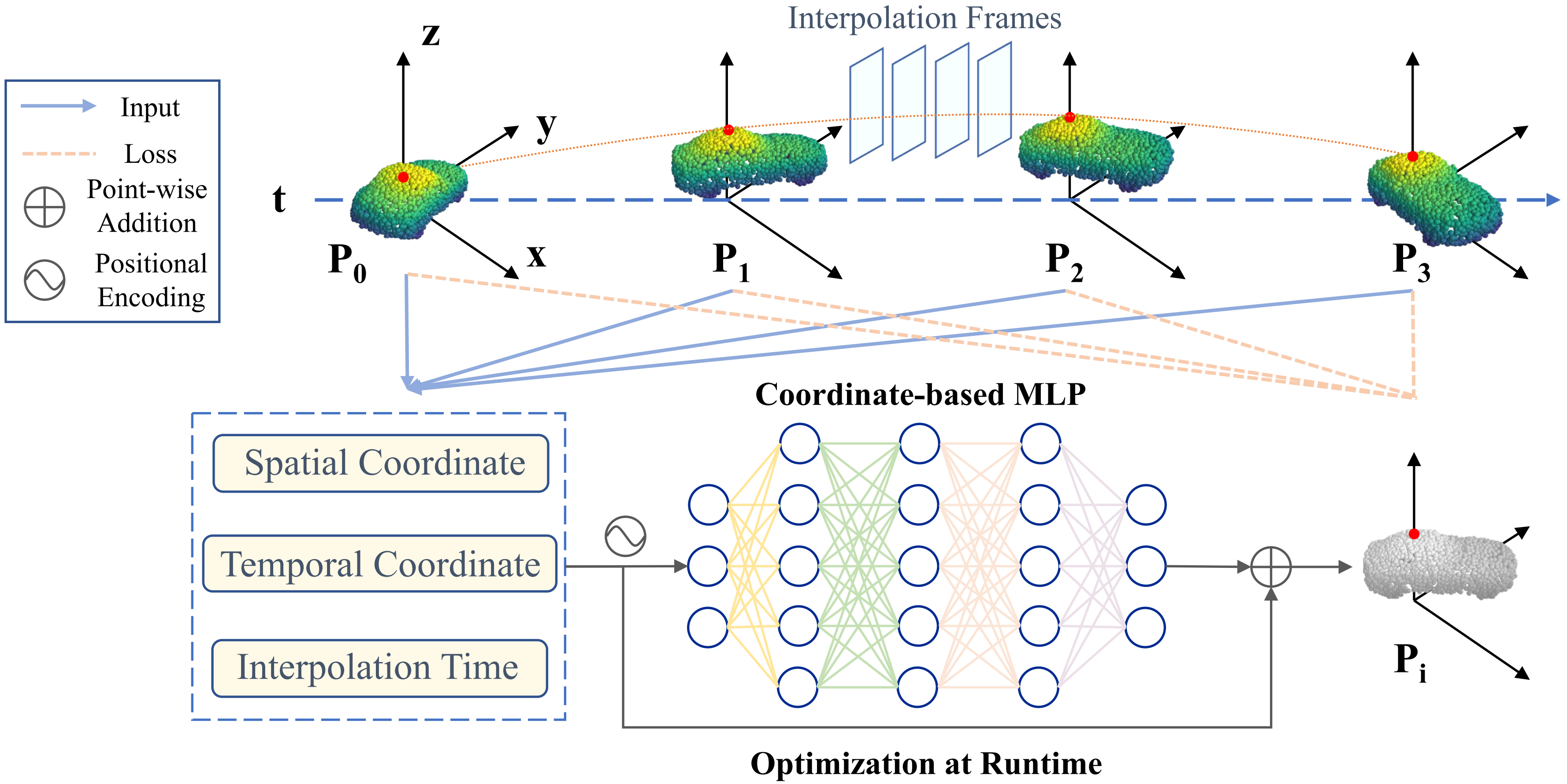
Method

- **4D** (x, y, z, t) **Spatio-temporal neural field**
  - Establish mapping: *Coordinate Field* ⟶ *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 $$\mathcal{L}_S=\sum_{p_i\in P}\frac{1}{|N(p_i)|}\sum_{p_j\in N(p_i)}\|\Delta x(p_j)-\Delta x(p_i)\|_2^2$$

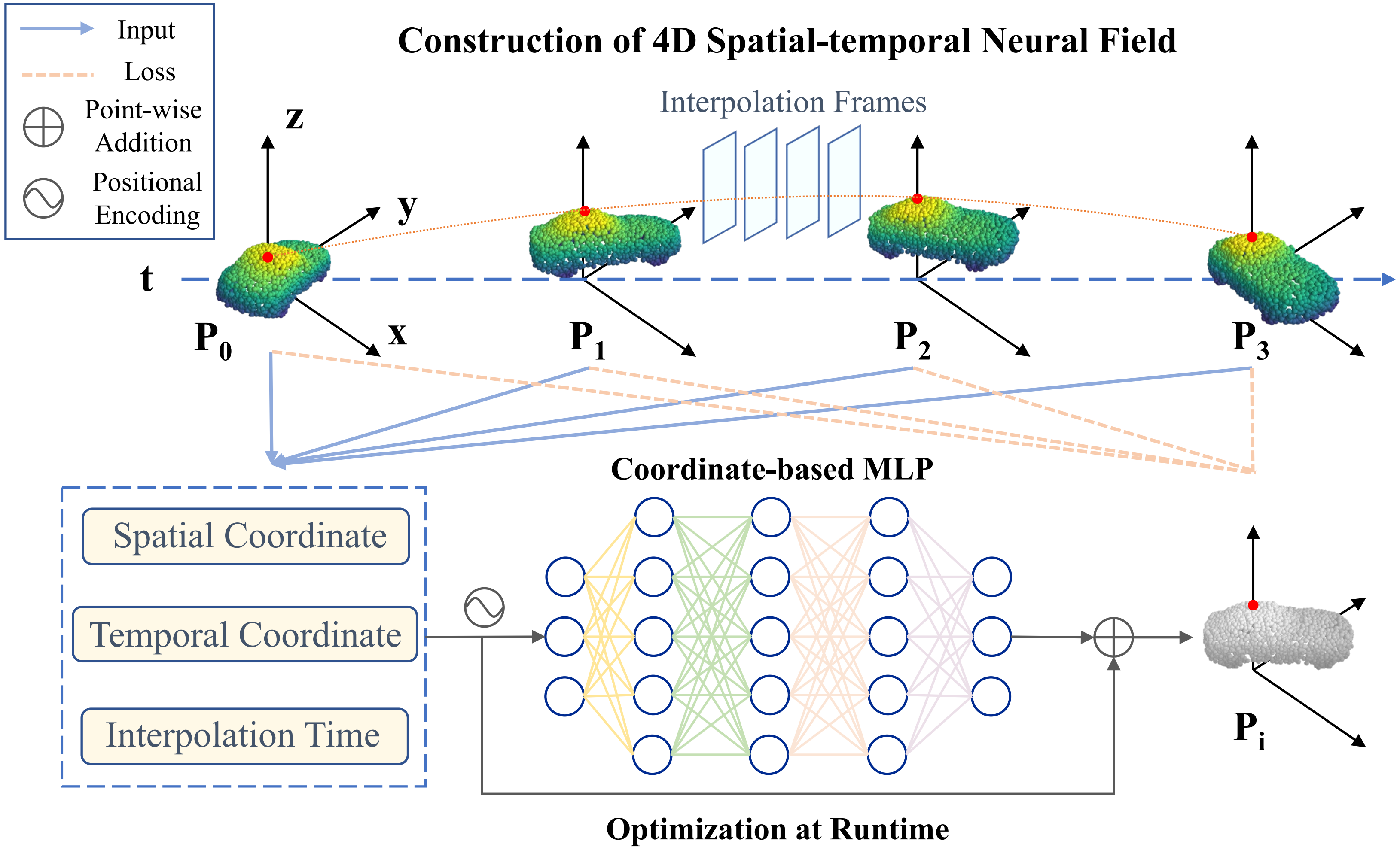
Overall Loss

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

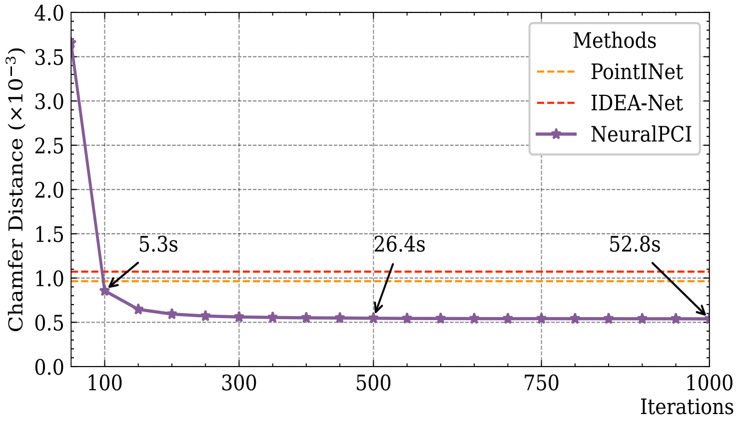
$$\mathcal{L}=\sum_{P_i\in S}\sum_{t_j\in T}\Psi\left(P_{t_j},\hat{P}_i^{t_j}\right)$$

Method

- **Runtime optimization**

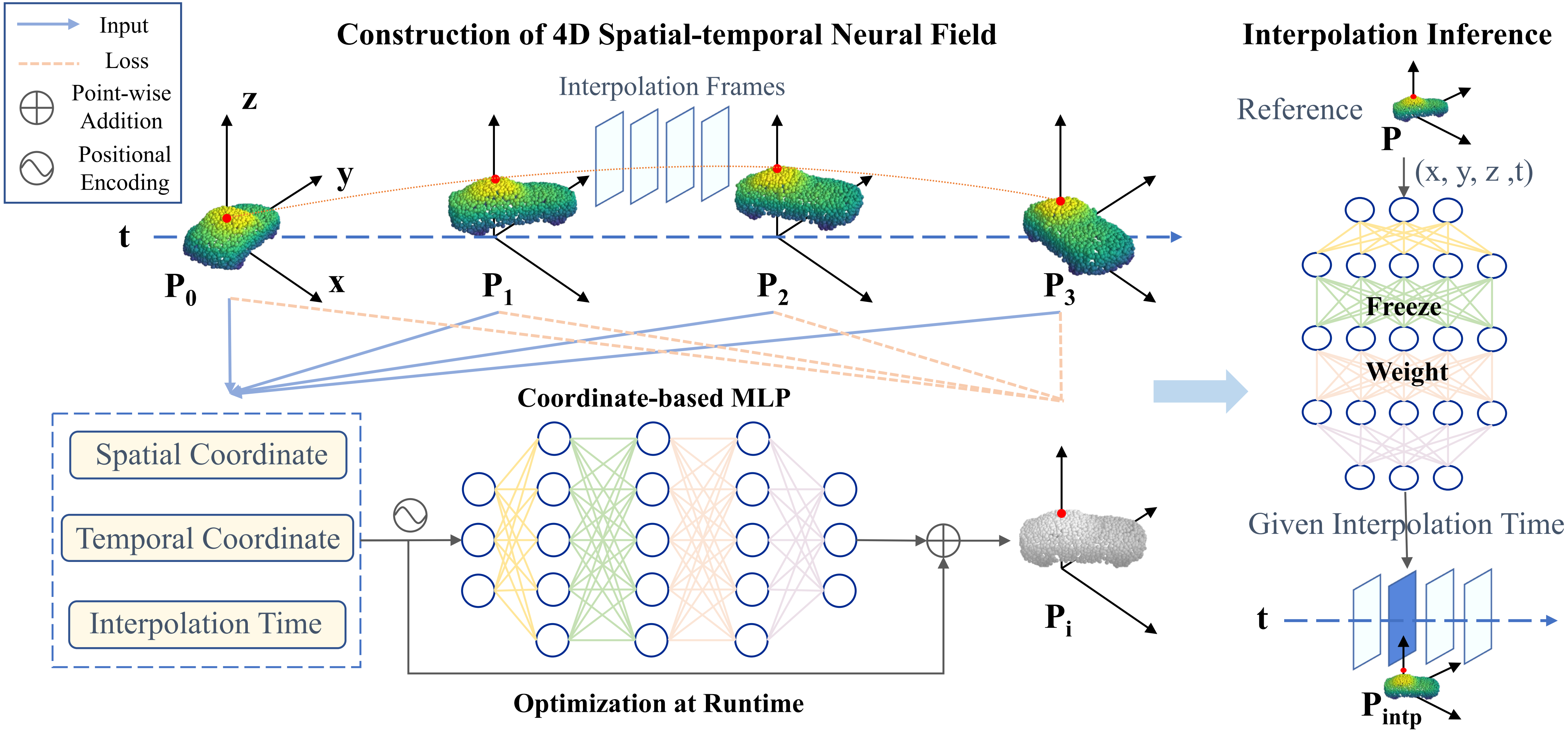


Fast to converge



Method

- Runtime optimization & **Inference**

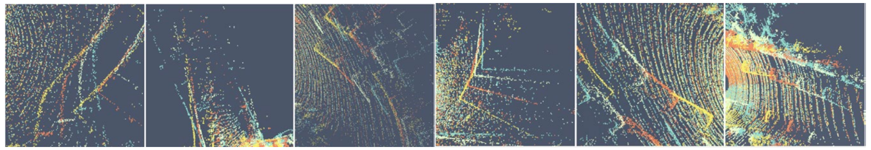


Experiments

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**

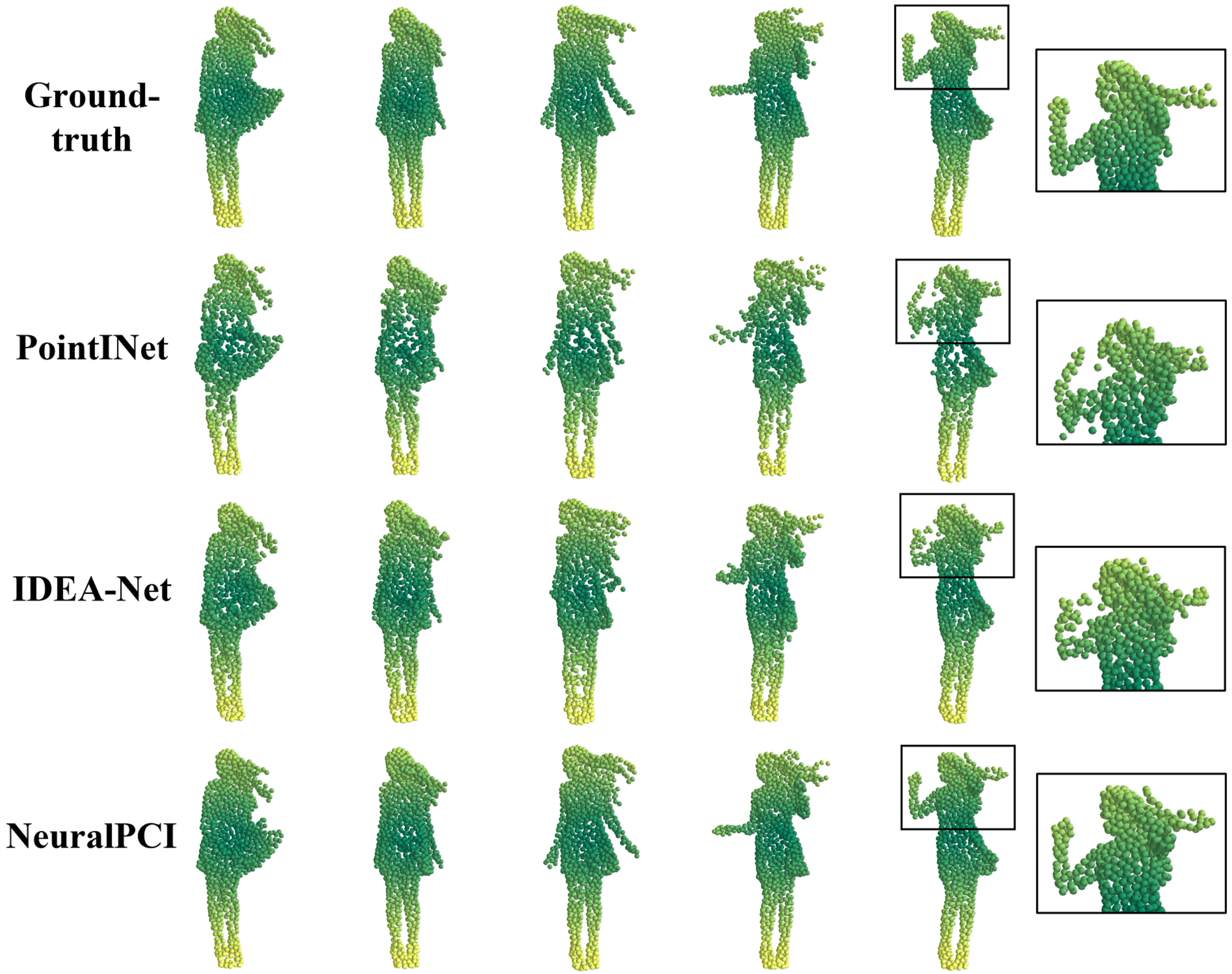
Results on NL-Drive Dataset



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**

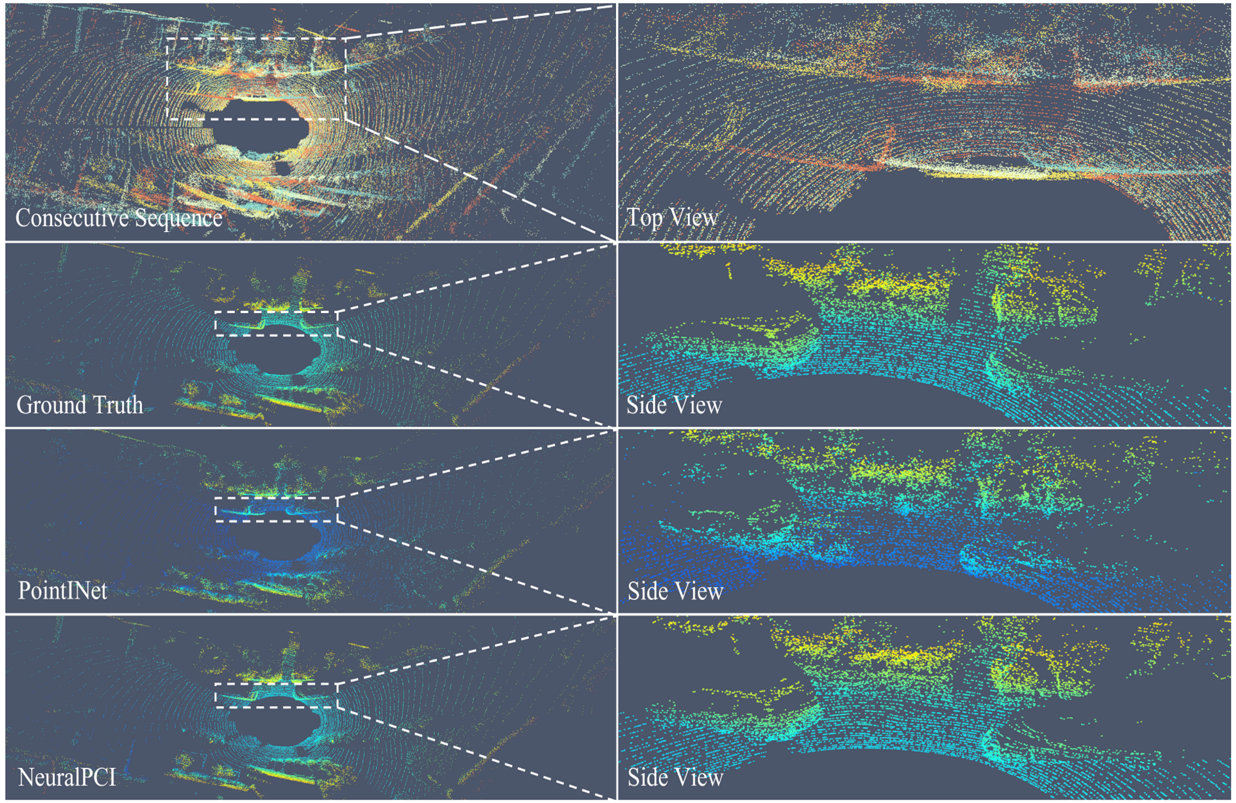
Results

Indoor Scenes



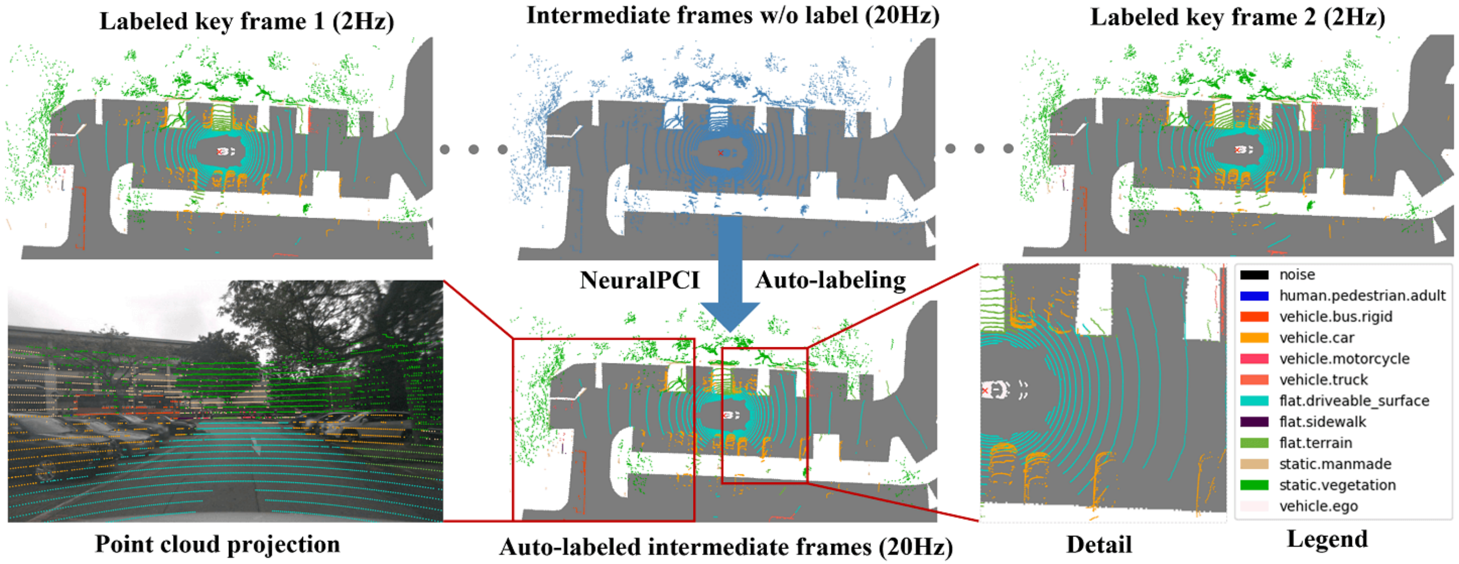
Results

Outdoor Scenes



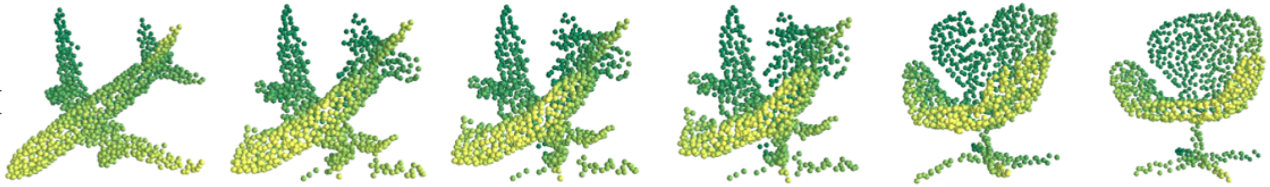
Other applications

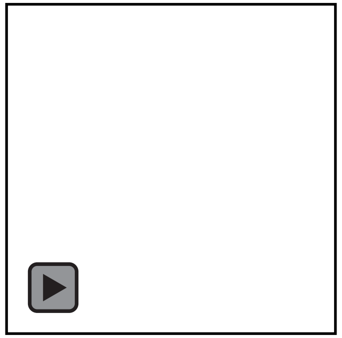
Auto-labeling



Point Cloud Morphing

Neural PCI





Conclusion & Takeaways

Neural field is awesome!

- Extending to multi-frame increases the receptive field of the time domain
- Benefiting from the fitting ability and smoothness of MLP
- Optimizing interpolation in a self-supervised manner
- Bridging different tasks via neural fields

Limitation

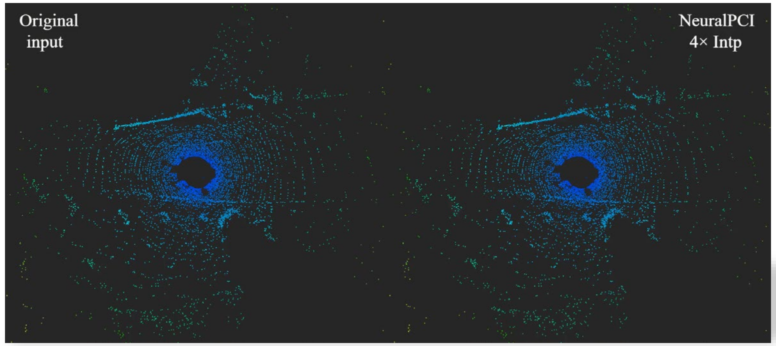
- Runtime optimization (per-scene fitting) limits the real-time application
- There is still improvement space on large scenes and extrapolation

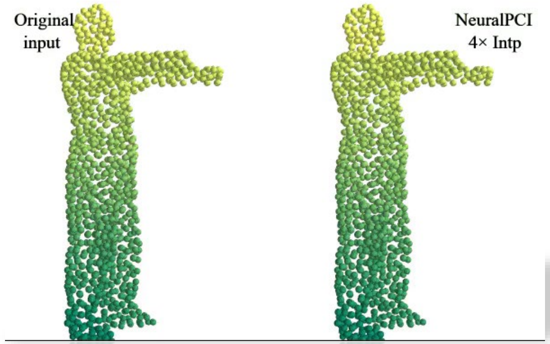
同濟大學 TONGJI UNIVERSITY

同濟大學 智能汽车研究所 TONGJI UNIVERSITY Institute of Intelligent Vehicles

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Thank you for listening





Project Page: https://dyfcalid.github.io/NeuralPCI