Physics-Informed Machine Learning for Real-Time Sensing Systems

Xiyuan Zhang, Xiaohan Fu, Diyan Teng, Chengyu Dong, Keerthivasan Vijayakumar, Jiayun Zhang, Ranak Roy Chowdhury, Junsheng Han, Dezhi Hong, Rashmi Kulkarni, Jingbo Shang, Rajesh Gupta

University of California, San Diego
Qualcomm

Outline

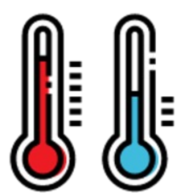
- Motivation
- Method
- Experiments
- Conclusion

Outline

- Motivation
- Method
- Experiments
- Conclusion

Motivation

- Sensors measuring real-life physical processes play a critical role in acquiring data for downstream pattern recognition or decision making
- Challenge: noisy measurements degrade data quality
    - Environmental interference
    - Electrical fluctuation
    - Imprecision of the sensors



Motivation

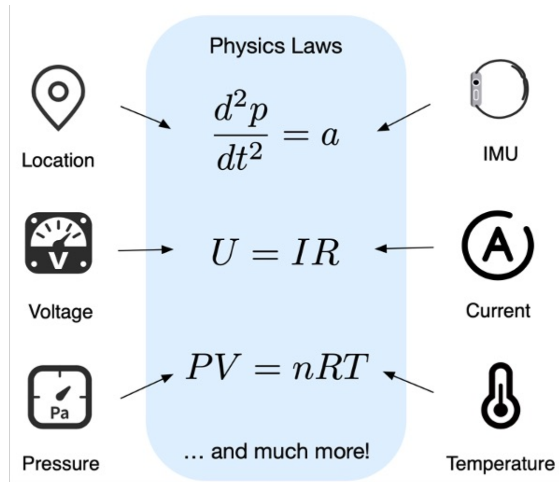
Limitation

- Traditional filtering methods
  - rely on prior knowledge of signal characteristics
- Supervised machine learning methods
  - assume availability of ground truth clean data
- Unsupervised machine learning methods
  - make simplified assumptions on noise/signal distributions

Motivation

Key Observation

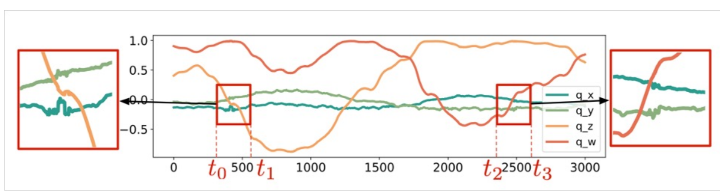
- Different variables of sensor data often have correlations that can be characterized by physics equations
    - Motion law: location, acceleration
    - Ohm's law: voltage, current
    - Ideal gas law: pressure, temperature
    - And more!



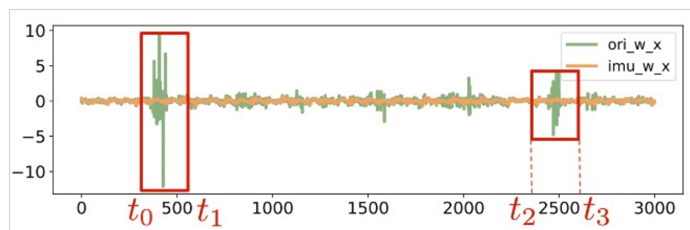
Motivation

Key Observation

- Key observation: Such physics-based constraints among measured channels can be used to make the learning process of denoising easier
- Example: Regions with higher noise level in the orientation data map to regions with large misalignment of physics equations



Noisy orientation



Physics Equation Misalignment

Motivation

Key Contributions

- The first **P**hysics-**I**nformed **L**earning for den**O**ising **T**echnology (**PILOT**) that supports practical sensing applications
  - Inertial navigation
  - CO_2 monitoring
  - HVAC control system
- State-of-the-art denoising performance with real-world deployment

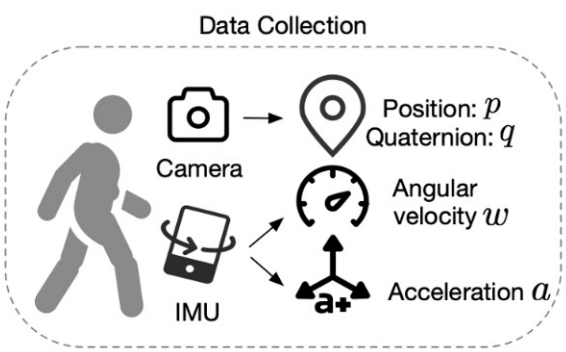
Outline

- Motivation
- Method
- Experiments
- Conclusion

Method

Example - Inertial Navigation

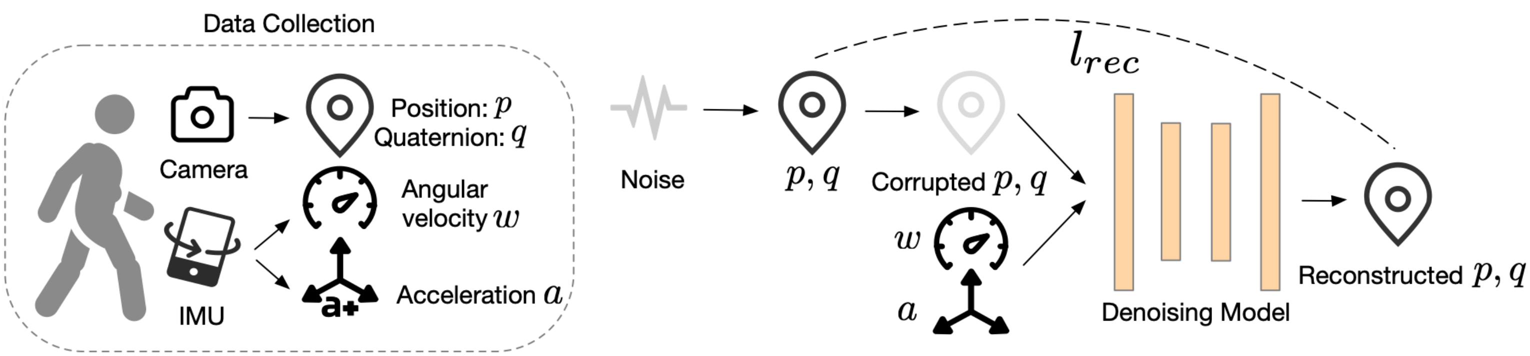
- Data collection
  - Camera: position/orientation
  - IMU: angular velocity/acceleration



Method

Example - Inertial Navigation

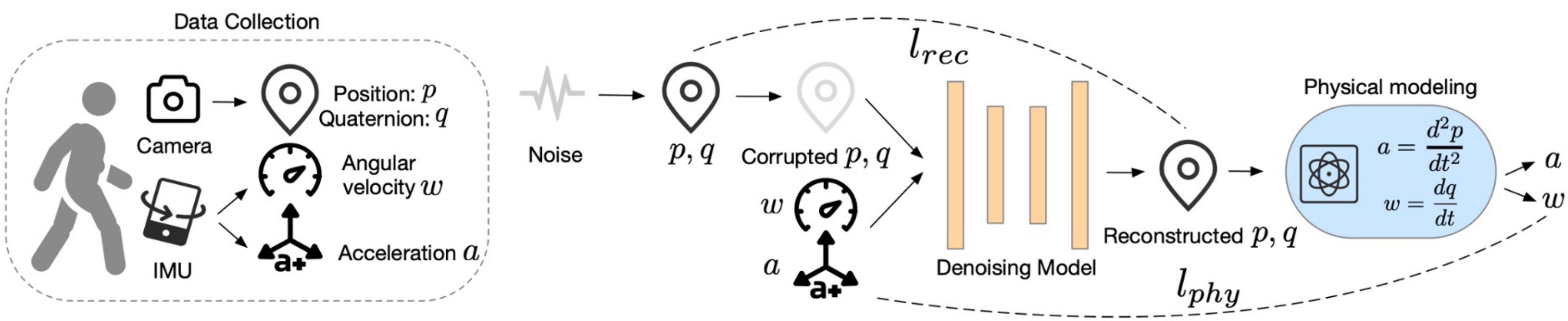
- Manually add noise to position and orientation
- Conditional denoising autoencoder (4 convolutional layers)
- Reconstruction loss $L_{rec} = ||(\text{Reconstructed } p, q) - (p, q)||^2$



Method

Example - Inertial Navigation

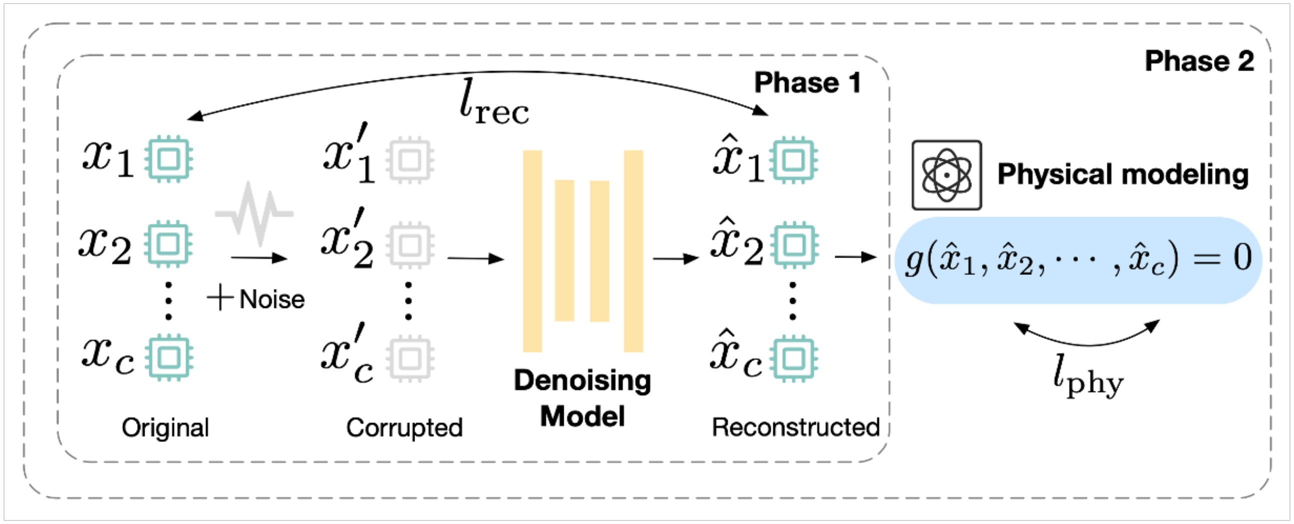
- Physics modeling: compute 1st and 2nd order derivatives of reconstructed orientation and positions
- Physics loss $L_{phy} = ||(\text{Derived } w, a) - (\text{IMU } w, a)||^2$



Method

General Framework

- Phase 1: reconstruction loss as warm-up phase
- Phase 2: reconstruction loss + physics loss



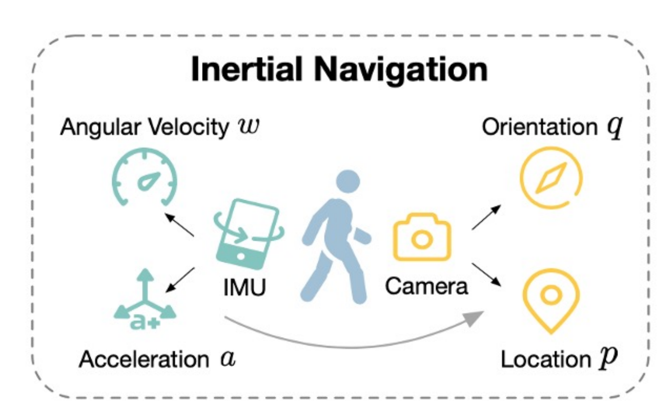
Method

More Applications

- Inertial navigation

$$g_1(a, p, q) = a - R_q^T\left(\frac{d^2 p}{dt^2} - g_0\right)$$

$$g_2(w, q) = \frac{dq}{dt} - \frac{1}{2} q \otimes w$$



Method

More Applications

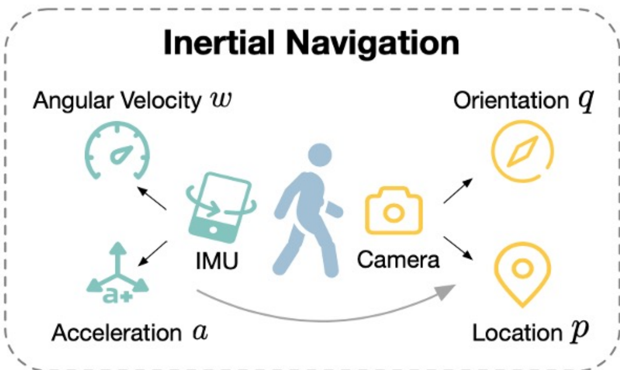
- Inertial navigation

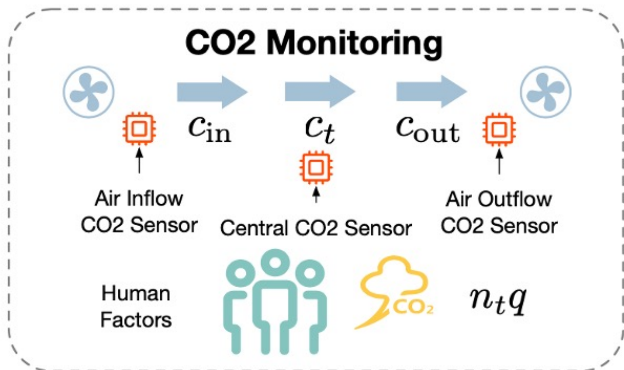
$$g_1(a,p,q) = a - R_q^T\left(\frac{d^2p}{dt^2} - g_0\right)$$

$$g_2(w,q) = \frac{dq}{dt} - \frac{1}{2}q \otimes w$$

CO_2 monitoring

$$g(c_t, c_{\text{in}}^t, c_{\text{out}}^t) = c_tV - (c_0V + \sum_t (c_{\text{in}}^t v\Delta t)$$
$$+ \sum_t (n_t q \Delta t) - \sum_t (c_{\text{out}}^t v \Delta t))$$





Method

More Applications

- Inertial navigation

$$g_1(a,p,q) = a - R_q^T\left(\frac{d^2p}{dt^2} - g_0\right)$$

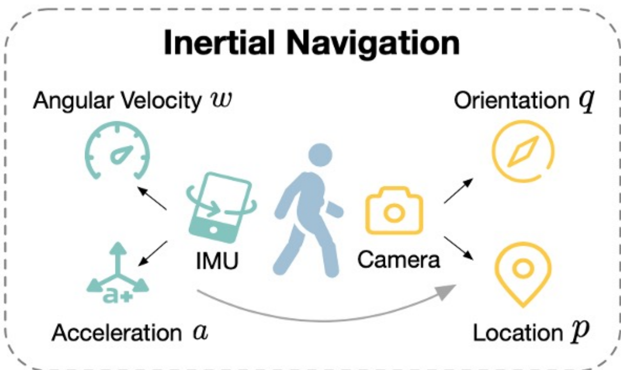
$$g_2(w,q) = \frac{dq}{dt} - \frac{1}{2}q \otimes w$$

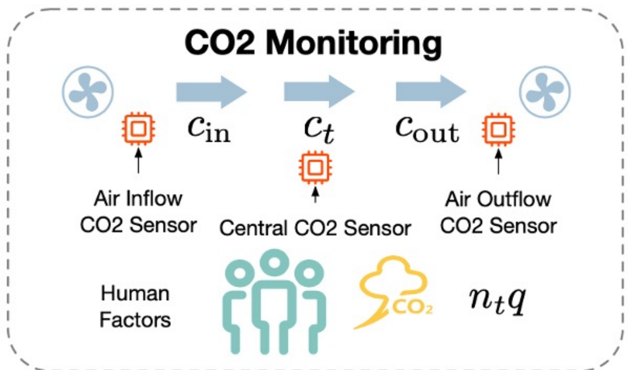
CO_2 monitoring

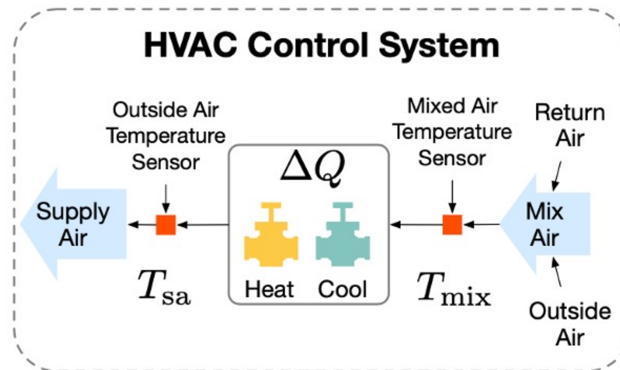
$$g(c_t, c_{\text{in}}^t, c_{\text{out}}^t) = c_tV - (c_0V + \sum_t (c_{\text{in}}^t v\Delta t) + \sum_t (n_t q \Delta t) - \sum_t (c_{\text{out}}^t v \Delta t))$$

HVAC control

$$g(\Delta Q, m, c, T_{\text{sa}}, T_{\text{mix}}) = \Delta Q - mc(T_{\text{sa}} - T_{\text{mix}})$$







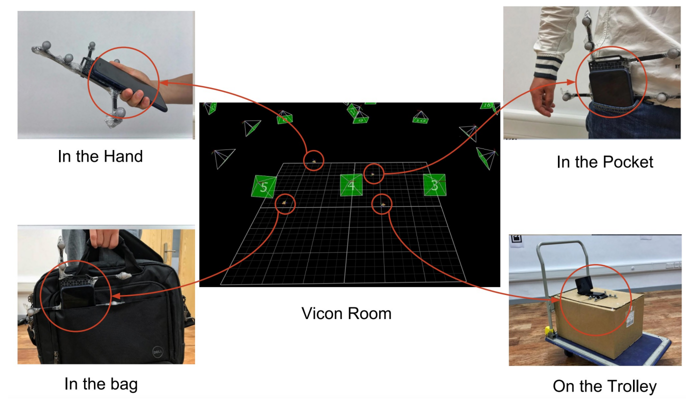
Outline

- Motivation
- Method
- Experiments
- Conclusion

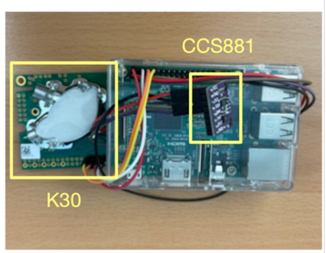
Experiments

Datasets

- Inertial navigation: public OxIOD dataset
- CO_2 monitoring: deployment in lab
- HVAC control: deployment on campus



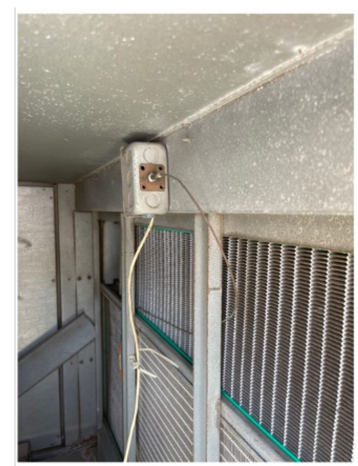
IMU + Vicon



CO_2 Desk

CO_2 Vent

HVAC ΔQ



HVAC ΔT

Experiments

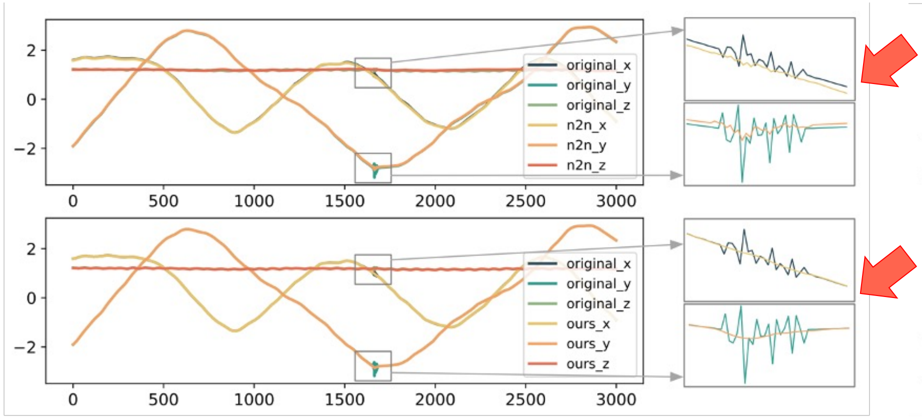
Metrics

- Three perspectives to evaluate the performance
  - Reconstruction performance
    - High quality data collected from other sources as approximate ground truth clean data
    - Visualizations
  - Physics alignment: how well the denoised data align with physics
  - Performance of downstream task based on the denoised data

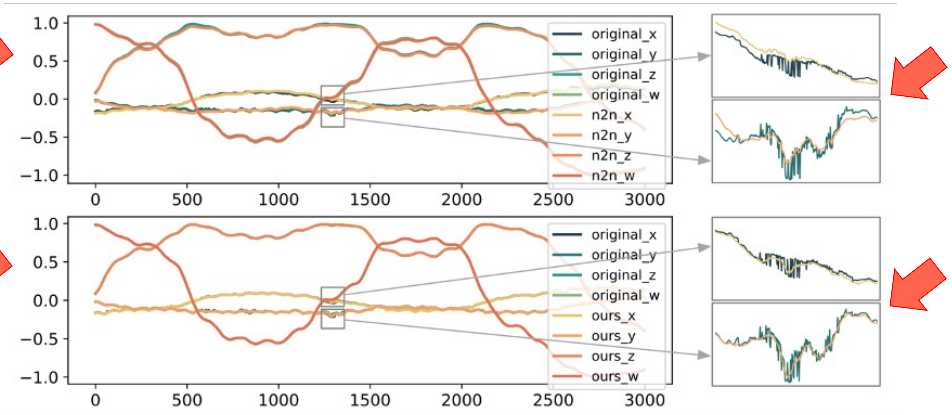
Experiments

Inertial Navigation

- Reconstruction performance
  - Visualizations (top: best baseline, bottom: PILOT)



Location



Orientation

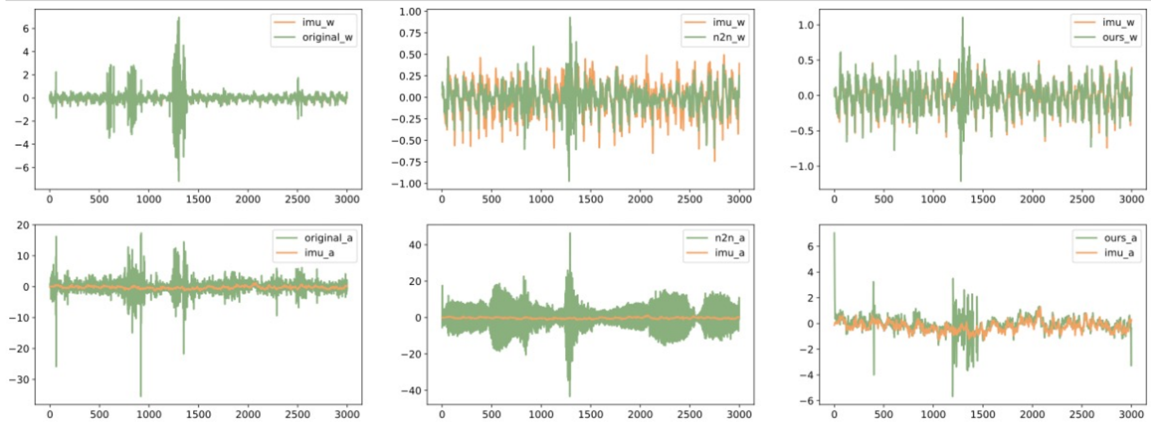
Experiments

Inertial Navigation

- Physics alignment (left: original noisy, middle: best baseline, right: PILOT)

Model	Acceleration (m/s^2)		Angular Velocity (rad/s)	
Metrics	MSE	MAE	MSE	MAE
Original	762.6	3.7862	2.6219	0.2376
Gaussian	363.2	3.2295	1.6277	0.2161
DWT	854.9	5.4534	2.6034	0.2701
DnCNN	312.5	8.3830	0.3470	0.1896
TSTNN	3272.0	30.513	0.4184	0.4836
DIP	2153.6	33.938	0.3788	0.4013
N2N	118.7	4.5749	0.3565	0.1756
PILOT	**1.8695**	**0.6372**	**0.0380**	**0.0690**

Angular Velocity from IMU vs 1st-order Derivative from Orientation



Acceleration from IMU vs 2nd-order Derivative from Location

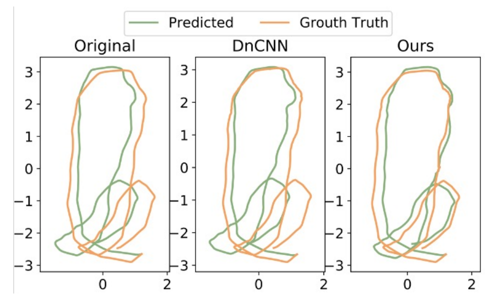
Experiments

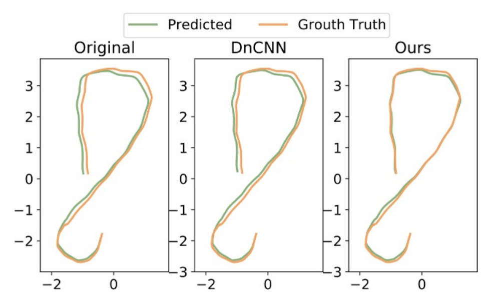
Inertial Navigation

- Downstream performance

Model	IONet [5]						RoNIN [19]					
Metrics	vx (m/s)	vy (m/s)	vz (m/s)	mean v (m/s)	ATE (m)	RTE (m)	vx (m/s)	vy (m/s)	vz (m/s)	mean v (m/s)	ATE (m)	RTE (m)
Original	0.0207	0.0642	0.0093	0.0314	0.3076	0.8194	0.0180	0.0621	0.0090	0.0297	0.2472	0.6337
Gaussian	0.0249	0.0496	0.0145	0.0297	0.6111	1.8727	0.0242	0.0498	0.0147	0.0296	0.5988	1.8427
DWT	0.0266	0.0732	0.0094	0.0364	0.3142	0.8079	0.0243	0.0714	0.0091	0.0349	0.2665	0.7023
DnCNN	0.0200	0.0235	0.0144	0.0193	0.3001	0.7891	0.0177	0.0213	0.0139	0.0176	0.2476	0.6598
TSTNN	0.2857	0.3250	0.0935	0.2348	0.6496	1.6575	0.2865	0.3253	0.0938	0.2352	0.6256	1.5794
DIP	0.1971	0.2650	0.0105	0.1576	0.5759	1.5108	0.1926	0.2570	0.0101	0.1533	0.3989	1.0358
N2N	0.0246	0.0144	0.0183	0.0191	0.3151	0.8317	0.0224	0.0122	0.0182	0.0176	0.2605	0.6956
PILOT	**0.0102**	**0.0095**	**0.0031**	**0.0076**	**0.2998**	**0.7875**	**0.0081**	**0.0078**	**0.0017**	**0.0059**	**0.2413**	**0.6309**

Example Trajectory 1





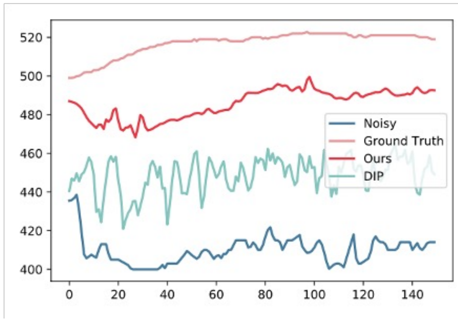
Example Trajectory 2

Experiments

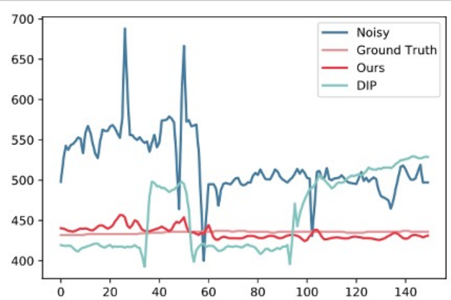
CO_2 Monitoring

- Reconstruction and physics alignment

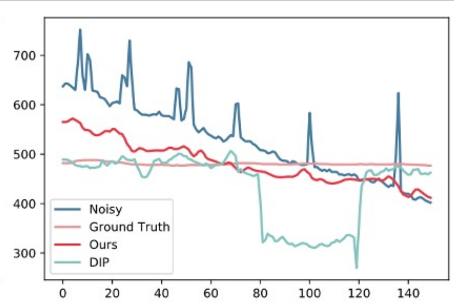
Model	Recons (1×10^6 ppm)		Physics (1×10^6 ppm)	
Metrics	MSE	MAE	MSE	MAE
Original	1.5654	0.0020	0.1082	0.1908
Gaussian	0.6265	0.0016	0.0278	0.1019
DWT	1.5076	0.0020	0.1045	0.1846
DnCNN	1.5381	0.0020	0.1064	0.1897
TSTNN	0.0956	0.0007	0.0027	0.0498
DIP	0.0841	0.0006	0.0018	0.0308
N2N	0.4396	0.0018	0.0085	0.0789
PILOT	**0.0371**	**0.0004**	**0.0012**	**0.0200**



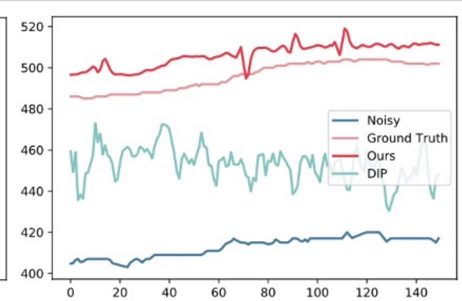
Example 1



Example 2



Example 3



Example 4

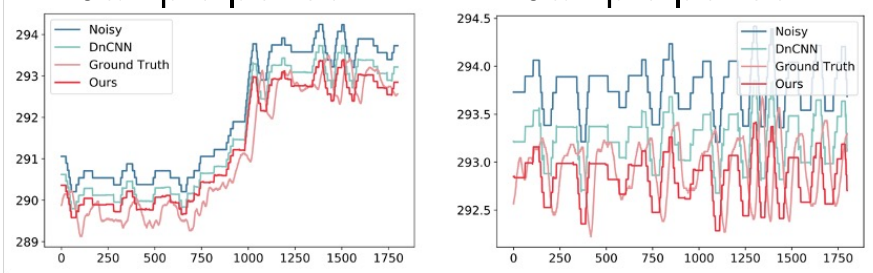
Experiments

HVAC Control

- Reconstruction and physics alignment

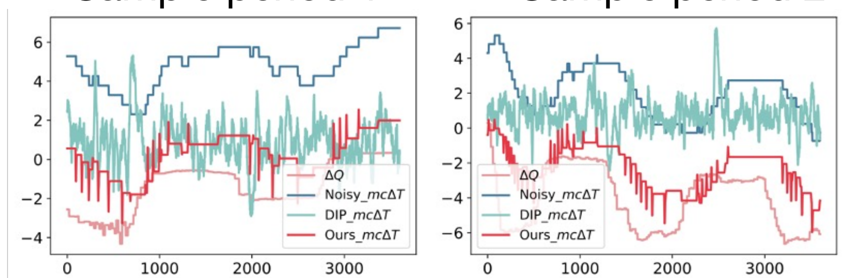
Model	Reconstruction (K)		Physics (K)	
Metrics	MSE	MAE	MSE	MAE
Original	0.9479	0.8841	50.302	6.7184
Gaussian	0.8687	0.8782	49.528	6.7003
DWT	0.8553	0.8689	48.782	6.6471
DnCNN	0.3284	0.4786	50.380	6.7241
TSTNN	2.2980	1.1980	32.012	3.7260
DIP	3.7336	1.5098	29.747	3.7410
N2N	1.1830	0.9522	31.748	3.6990
PILOT	**0.1994**	**0.3454**	**14.081**	**3.1600**

Sample period 1 Sample period 2



Example reconstructions

Sample period 1 Sample period 2



Example physics alignment

Experiments

Ablation Study

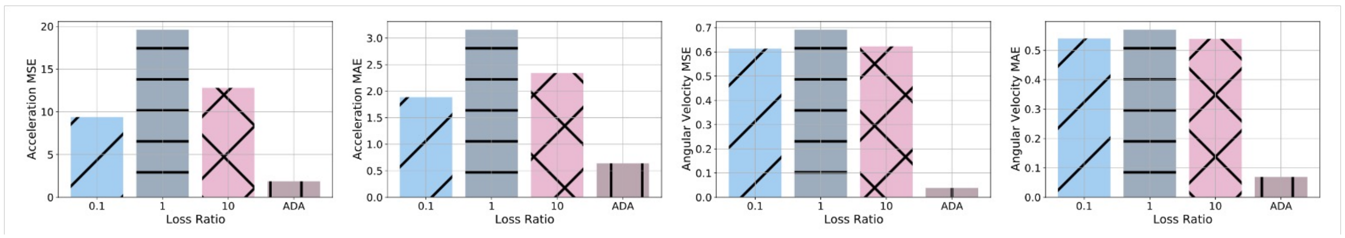
- w/o physics loss
- w/o reconstruction loss
- w/o 2-phase training

Task	Metrics	w/o l_{phy}	w/o l_{rec}	w/o pre-train	PILOT
INS	MSE_a	316.5	259.1	20.70	**1.8695**
	MAE_a	8.342	8.759	2.2930	**0.6372**
	MSE_w	0.3579	0.2845	0.1380	**0.0380**
	MAE_w	0.1899	0.3352	0.1291	**0.0690**
CO_2	MSE_{rec}	0.4641	0.0568	0.0724	**0.0371**
	MAE_{rec}	0.0139	0.0051	0.0074	**0.0047**
	MSE_{phy}	0.0194	0.0021	0.0023	**0.0012**
	MAE_{phy}	0.0781	0.0264	0.0330	**0.0200**
HVAC	MSE_{rec}	0.3314	0.5314	0.2686	**0.1994**
	MAE_{rec}	0.4709	0.6128	0.4670	**0.3454**
	MSE_{phy}	50.22	14.69	14.469	**14.081**
	MAE_{phy}	6.713	3.263	3.203	**3.1600**

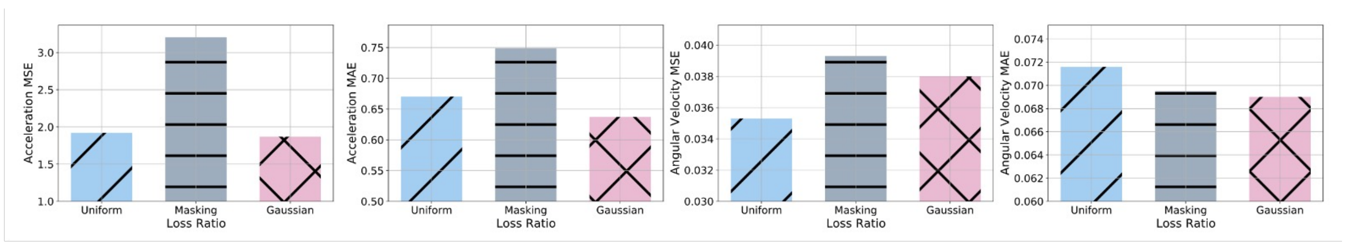
Experiments

Sensitivity Analysis

- Adaptive loss ratio between physics loss and reconstruction loss works the best



- Random noise performs better than zero masking



Experiments

Edge Deployment - Inertial Navigation

- Use Raspberry Pi 4 as an example edge device
- 8-bit training-aware quantization
- Time efficiency: 4 ms to denoise a 1s-sequence (100 readings)
- Memory efficiency (model size)

Metrics	Params	Size	Inference Time	CPU Usage
Efficiency	270K	284 KB	4 ms	25%

Outline

- Motivation
- Method
- Experiments
- Conclusion

Conclusion

- The first physics-informed sensor denoising algorithm
  - Inertial navigation
  - CO_2 monitoring
  - HVAC control system
- State-of-the-art denoising performance with real-world deployment

Thank you!

xiyuanzh@ucsd.edu