

Physics-Informed Data Denoising for Real-Life Sensing Systems

Xiyuan Zhang
University of California, San Diego
xiyuanzh@ucsd.edu

Xiaohan Fu
University of California, San Diego
xhfu@ucsd.edu

Diyan Teng
Qualcomm
diyateng@qti.qualcomm.com

Chengyu Dong
University of California, San Diego
cdong@ucsd.edu

Keerthivasan Vijayakumar
University of California, San Diego
kevijayakumar@ucsd.edu

Jiayun Zhang
University of California, San Diego
jiz069@ucsd.edu

Ranak Roy Chowdhury
University of California, San Diego
rrchowdh@ucsd.edu

Junsheng Han
Qualcomm
junsheng@qti.qualcomm.com

Dezhi Hong*
Amazon
hondezhi@amazon.com

Rashmi Kulkarni
Qualcomm
rashmik@qti.qualcomm.com

Jingbo Shang
University of California, San Diego
jshang@ucsd.edu

Rajesh K. Gupta
University of California, San Diego
rgupta@ucsd.edu

ABSTRACT

Sensors measuring real-life physical processes are ubiquitous in today's interconnected world. These sensors inherently bear noise that often adversely affects the performance and reliability of the systems they support. Classic filtering approaches introduce strong assumption on the time or frequency characteristics of sensory measurements, while learning-based denoising approaches typically rely on using ground truth clean data to train a denoising model, which is often challenging or prohibitive to obtain for many real-world applications. We observe that in many scenarios, the relationships between different sensor measurements (e.g., location and acceleration) are analytically described by laws of physics (e.g., second-order differential equation). By incorporating such physics constraints, we can guide the denoising process to improve performance even in the absence of ground truth data. In light of this, we design a physics-informed denoising model that leverages the inherent algebraic relationships between different measurements governed by the underlying physics. By obviating the need for ground truth clean data, our method offers a practical denoising solution for real-world applications. We conducted experiments in various domains, including inertial navigation, CO₂ monitoring, and HVAC control, and achieved state-of-the-art performance compared with existing denoising methods. Our method can denoise data in real time (4ms for a sequence of 1s) for low-cost noisy sensors and produces results that closely align with those from high-precision, high-cost alternatives, leading to an efficient, cost-effective approach for more accurate sensor-based systems.

*Work unrelated to Amazon.



This work is licensed under a Creative Commons Attribution International 4.0 License.

SenSys '23, November 12–17, 2023, Istanbul, Turkiye
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0414-7/23/11.
<https://doi.org/10.1145/3625687.3625811>

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; • **Computing methodologies** → *Machine learning*; • **Applied computing** → *Physics*.

KEYWORDS

physics-informed machine learning, denoise, inertial navigation, CO₂ monitoring, HVAC control

ACM Reference Format:

Xiyuan Zhang, Xiaohan Fu, Diyan Teng, Chengyu Dong, Keerthivasan Vijayakumar, Jiayun Zhang, Ranak Roy Chowdhury, Junsheng Han, Dezhi Hong, Rashmi Kulkarni, Jingbo Shang, and Rajesh K. Gupta. 2023. Physics-Informed Data Denoising for Real-Life Sensing Systems. In *The 21st ACM Conference on Embedded Networked Sensor Systems (SenSys '23)*, November 12–17, 2023, Istanbul, Turkiye. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3625687.3625811>

1 INTRODUCTION

Sensors measuring various real-life physical processes have permeated our daily lives. These sensors play a crucial role in acquiring a large amount of data in various applications including environmental monitoring, healthcare, smart home and building, and transportation, enabling context inference, pattern recognition and informed downstream decision-making. However, because of factors such as environmental interference, electrical fluctuations, and imprecision of the sensor itself, sensor data are naturally noisy. Such noise degrades data quality and adversely affects performance of downstream applications.

To improve the quality of sensor data, there has been a line of research on noise reduction, from traditional filtering approaches that rely on prior knowledge of signal characteristics in time or frequency domain [11, 38], to more advanced machine learning methods [51, 63]. Existing machine learning-based methods typically assume availability of ground truth clean data in order to train a denoising model in a supervised manner. However, given the inherently noisy nature of sensors, we often do not have access to ground truth *clean* data in real-world applications. In addition to

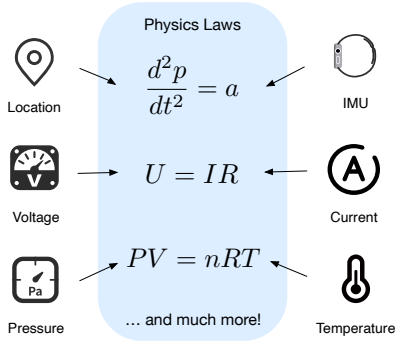


Figure 1: Real-life data captured by sensors are often governed by the laws of physics.

supervised denoising methods, researchers have also developed self-supervised denoising methods, mostly in computer vision applications [26, 34]. These approaches often make simplified assumptions about e.g., noise distributions, which may not accurately reflect real-world sensor data with complex correlations. Moreover, they do not leverage the unique physical characteristics of sensor data distinct from typical image or text data.

In multiple applications, we observe that different measurement channels of sensor data collected from real-life physical processes are often correlated and characterized by equations rooted in well-understood underlying physics. We illustrate three example scenarios in Figure 1. In the first example, the relationship between location and acceleration is captured by the motion law, which equates the second-order derivative of location to acceleration. The second example connects the measurements by drawing upon Ohm’s law – the ratio of voltage and current is a constant, resistance, of the circuit. In the third example, the ideal gas law establishes a link between pressure and temperature measurements. These three examples are merely a sample of the numerous physics principles that govern real-life sensing systems. This paper builds upon the intuition that *such physics-based constraints among measured channels can be used to improve denoising techniques for sensor data*.

To this end, we propose a physics-informed denoising method, PILOT (Physics-Informed Learning for denOising Technology), by incorporating analytical model based on known physics as loss functions during training. Specifically, we introduce physics-derived equations as soft constraints that the denoised output from our model needs to satisfy. We illustrate the model architecture in Figure 4. We intentionally add noise to corrupt the original data collected from multiple sensors. Feeding this corrupted data as input, we train a denoising model to remove the added noise by minimizing the *reconstruction loss* between the model output and the original undistorted input. However, since the original sensor data are inherently noisy, a reconstruction loss alone is insufficient to remove such noise. Therefore, we introduce an additional *physics-based loss* to minimize physics misalignment of denoised output. For the example of inertial navigation, the corresponding physics equations include differential equations between position/orientation and acceleration/angular velocity. Therefore, in this example, the physics constraint would penalize discrepancy between the derivatives of location/orientation and the acceleration/angular velocity.

Denoising processes for other applications follow similarly via a unified mathematical formulation that we will elaborate.

To evaluate our approach, we conducted experiments on different real-world scenarios including inertial navigation, CO₂ monitoring, and HVAC (Heating, Ventilation, and Air Conditioning) control. For inertial navigation, PILOT generates denoised results that are most coherent to physics equations, leading to the best performance in downstream inertial navigation applications on the benchmark OxIOD dataset [6]. We collected data for the other two applications, detailed in Section 5.3 and Section 5.4. For CO₂ monitoring, we have two types of CO₂ sensors—one is low-cost and very noisy, and the other is more accurate but much more expensive. Denoised CO₂ measurements from the low-cost sensors by PILOT closely match those from the much more expensive CO₂ sensors. Consequently, our denoising technique offers significant cost savings. We observed similar advantage for HVAC control system, where our denoised temperature data collected from low-cost noisy sensors best match those collected from industry-grade sensors. In summary, this paper makes the following contributions:

- We propose a novel denoising method, PILOT, trained under the guidance of physics constraints. To our best knowledge, this is the first physics-informed denoising method that supports different sensing applications.
- PILOT offers a practical denoising solution for application scenarios where obtaining ground truth clean data or understanding the underlying noise distribution is challenging, better representing real-world sensing environments.
- We collected two datasets on CO₂ monitoring and HVAC control, with pairs of noisy and accurate sensor data.
- Extensive experiments in three real-world applications demonstrate that PILOT produces results that are closest to the clean data and most coherent to the laws of physics, achieving state-of-the-art denoising performance. We also deployed the model on edge devices and showed real-time denoising capability.

2 RELATED WORK

2.1 Physics-Informed Machine Learning

2.1.1 General Physics-Informed Machine Learning. The concept of Physics Priors refers to our understanding of the inherent principles of the physical world, taking various representations such as differential equations, symmetry constraints, and common-sense knowledge, among others. An emerging field, Physics-Informed Machine Learning (PIML), aims to incorporate physics priors into machine learning models [17, 22, 53]. A line of research has been proposed for such integration, which can be categorized into different directions: incorporating physics priors as loss regularization [15, 23, 41, 42], imposing strict constraints into the model architectures [48, 52, 60] or hybrid approaches [16]. In general, physics-informed machine learning methods enhance efficacy, generalizability, and robustness of the models. However, most works in the PIML domain evaluate on simplified synthetic datasets.

2.1.2 Physics-Informed Machine Learning for Sensing Systems. Several works (such as PIP [61], PACMAN [35], Reducio [4], and PhyAug [33]) employ physics in practical settings. However, most approaches are domain-specific [8, 12, 18, 24], and miss the consider-

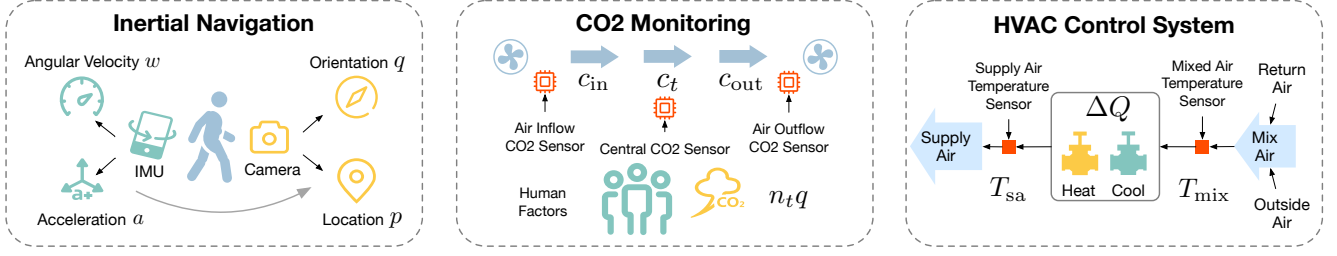


Figure 2: Three example applications of PILOT. Inertial Navigation (left): PILOT denoises inertial measurements (angular velocity and acceleration) and camera measurements (location and orientation) based on the motion law; CO₂ Monitoring (mid): PILOT denoises CO₂ readings leveraging the CO₂ relationship in a room; HVAC Control System (right): PILOT denoises the supply and mix air temperature sensors of an AHU utilizing the heat transfer equation.

ation of physical relationships among different sensor measurement channels that can assist with the denoising process. Multimodal learning methods [21, 32, 54, 64] are also relevant topics as different modalities mutually compensate and offer enriched information, but existing multimodal methods mostly focus on fusing multiple modalities for downstream prediction tasks (e.g., human activity recognition) without addressing the denoising challenge. To address this gap, we present the first physics-informed machine learning model that incorporates these relationships into denoising sensor data, pushing the capability of what is currently achievable in PIML.

2.2 Denoising Methods

Sensor data are often noisy due to intrinsic characteristics such as thermal noise, operating point drifts, or environmental effects. Existing works have proposed methods to denoise sensor data or adapt the model to noisy measurements [13, 27]. The majority of these strategies, however, is custom-tailored to specific domains of application, such as COTS WiFi-based motion sensing [65], image data processing under various weather conditions [7], speech recognition [29], and clinical data analysis [40]. Some researchers have also proposed more general denoising approaches through traditional estimation methods, or time-frequency domain analysis such as discrete wavelet transform [11] and empirical mode decomposition [38]. However, traditional estimation methods such as Kalman smoother rely heavily on the assumption of Gaussian noise and Markovian transition, but real-world physics models such as human motion typically have a higher-order temporal correlation. These methods also rely on oversimplified assumptions that observed variables are conditionally independent over time. More recent deep learning approaches [51, 63] are supervised methods that require ground truth clean data, and they neglect the unique benefits provided by physics-rich real-life sensing systems. Self-supervised denoising methods have been recently proposed to train denoising model without access to ground truth clean data [20, 26, 34, 49, 57]. These methods are mainly designed for the computer vision domain and make assumptions about the noise.

2.3 Real-Life Sensing Systems

There exist numerous physics-rich real-life sensing applications. In this study, we focus on three applications: inertial navigation, CO₂ monitoring, and HVAC systems. Inertial navigation predicts location and orientation based on IMU data, which is prone to er-

roneous pose estimate rapidly over time [44]. Classic approaches incorporate external measurements such as GNSS and camera, to periodically correct the propagated pose value, at the cost of additional hardware and power. Recently, learning-aided approaches become popular [5, 19, 28, 30, 31, 45, 46, 59]. CO₂ monitoring monitors CO₂ level to ensure occupant comfort. Researchers have designed data collection [47] and generation [55] methods for CO₂ data. When modeling CO₂ data, existing works have developed data-driven, differential equation models [56]. Modeling CO₂ levels is beneficial for occupancy prediction and energy savings [2, 14]. HVAC Control System controls heating, ventilation and air conditioning based on temperature and airflow measurements [3, 36]. Apart from traditional model-based methods, recently deep reinforcement learning has also shown promising performance in modeling HVAC systems [25, 58, 62].

3 PRELIMINARY

3.1 Problem Formulation

Clean Data. We denote the unobserved ground truth clean dataset as $\mathcal{D}_X = \{\mathbf{X}_i\}_{i=0}^N$ which consists of N samples. Each sample $\mathbf{X}_i \sim \mathcal{X}$, $\mathbf{X}_i \in \mathbb{R}^{c \times T}$, where \mathcal{X} denotes the clean data space, T denotes the number of timesteps in the sensor data, and c denotes the number of sensor measurement channels. Each sample $\mathbf{X}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{ic}]^\top$, where $\mathbf{x}_{ij} \in \mathbb{R}^T$ represents the j th sensor measurement channel for the i th sample in the ground truth clean dataset.

Noisy Data. We denote our observed noisy sensor dataset as $\mathcal{D}_Y = \{\mathbf{Y}_i\}_{i=0}^N$, composed of N samples. Each sample $\mathbf{Y}_i \sim \mathcal{Y}$, $\mathbf{Y}_i \in \mathbb{R}^{c \times T}$, and $\mathbf{Y}_i = [\mathbf{y}_{i1}, \mathbf{y}_{i2}, \dots, \mathbf{y}_{ic}]^\top$, where \mathcal{Y} denotes the noisy data space, $\mathbf{y}_{ij} \in \mathbb{R}^T$ represents the j th sensor measurement channel for the i th sample in the noisy dataset.

Noise. Each noisy sample \mathbf{Y}_i is corrupted from clean sample \mathbf{X}_i by noise $\epsilon_i \sim \mathcal{E}$, $\epsilon_i \in \mathbb{R}^{c \times T}$, where \mathcal{E} denotes the noise space. Specifically, $\mathbf{Y}_i = \mathbf{X}_i + \epsilon_i$, $i = 1, 2, \dots, N$.

Physics Equations. Equation $g(\cdot)$ describes the relationships between different sensor measurement channels in the ground truth clean data, i.e., $g(\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{ic}) = 0$, $i = 1, 2, \dots, N$.

Denoising Model. The denoising model is parameterized by θ and maps noisy data to clean data by mapping $f(\cdot; \theta) : \mathcal{Y} \rightarrow \mathcal{X} \subset \mathbb{R}^{c \times T}$. We incorporate the physics equation $g(\cdot)$ as a constraint to optimize $f(\cdot; \theta)$ during training.

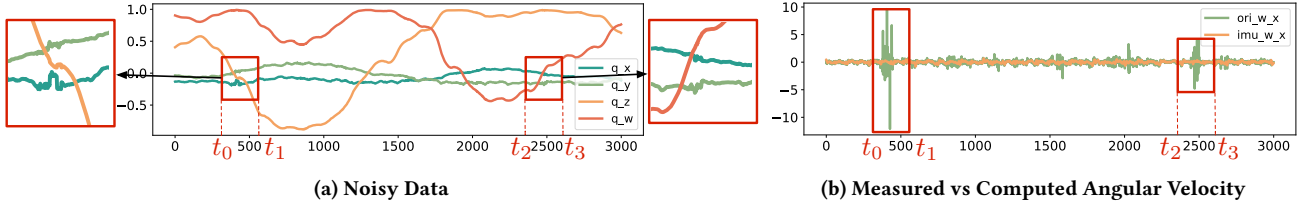


Figure 3: (a) Original orientation measurements. (b) Comparison of gyroscope measurements (*imu_w_x*) and angular velocities derived from orientations (*ori_w_x*) in x-axis. Regions with higher noise level ($t_0 \sim t_1, t_2 \sim t_3$) in the orientation data map to regions with large misalignment between IMU measurements and derived angular velocities. X-axis: timestep, Y-axis: meter.

3.2 Physics Principles in Sensing Systems

In this section, we introduce the underlying physics principles of three example sensing applications: inertial navigation, CO₂ monitoring, and HVAC control. PILOT leverages these physics equations as constraints during training to better guide the denoising process. We evaluate the denoising performance of these three sensing applications in Section 5. The major difference across different applications is the specific form of the physics constraint. Consequently, to adapt our model to new applications, the only requirement is to formulate the equation representing the physics relationship between different sensors in the new application. Thereafter, we can seamlessly incorporate the constraint, treat it as regularization, and maintain a uniform approach across varying applications.

We select these three use cases to demonstrate the different capabilities of the framework to model various physics relationships. The use cases cover different sensor types, domains, and scales. In particular, we choose the three applications to validate that the proposed method is robust across different scales — from single-human interactions to building air circulation models across many zones and users. Further, we seek to cover as many types of sensors (IMU, Camera, CO₂, airflow, temperature, etc.) as possible in these three experiments to demonstrate the potential to generalize to various types of sensors and noise levels. Our method can incorporate different physical laws with varying degrees of physics model complexity. We also picked the applications where physics models are informative but imperfect due to various modeling challenges. The selected applications are known for their dependencies on additional measurement modalities to compensate for the imperfect physics model, such as the visual information for navigation and IMU physics. This motivates the necessity for both learning component- as well as physics-based models.

3.2.1 Application Scenario I: Inertial Navigation. Inertial navigation aims to estimate a moving device’s position and orientation based on IMU sensor measurements, shown in Figure 2 (left). The inertial navigation model takes IMU measurements as input, and predicts locations and orientations. The ground truth location and orientation data are captured by cameras. However, both IMU and camera data carry noise due to factors like jittering or occlusion, posing challenges to training accurate inertial navigation models. If we denote location, orientation (in quaternion form), angular velocity, and acceleration as p, q, w, a respectively, then their relationships can be mathematically expressed by equations $g_1(\cdot), g_2(\cdot)$:

$$g_1(a, p, q) = a - R_q^T \left(\frac{d^2 p}{dt^2} - g_0 \right), \quad (1)$$

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

where R_q, g_0, \otimes represent rotation matrix, gravity constant and quaternion multiplication. As a motivating example, we randomly select 30-second sensor data from the OxIOD dataset [6] and visualize the orientation measurements in Figure 3a. We note two major noise-heavy regions around the 500th timestep and the 2500th timestep (indicated by boxed regions). Applying Equation 2, we derive the angular velocity from the orientation data, and compare it with the angular velocity directly measured by IMU gyroscope in Figure 3b. The misalignment between these two angular velocities appears around the same timesteps of noisy regions in the original orientation data. This highlights the role of known physics to guide the learning process of denoising models and its use in our model.

3.2.2 Application Scenario II: CO₂ Monitoring. CO₂ monitoring is of wide interest in smart building applications. An accurate zone CO₂ reading can enable transmission risk analysis [43], occupancy estimation [9], air quality monitoring [37] for energy-saving or residence safety purposes [14]. Figure 2 (middle) depicts one example placement of three CO₂ sensors proximate to the air intake, outtake, and the center of a standard office room. Assume $c_t, c_0, c_{in}^t, c_{out}^t$ represent the current average CO₂ concentration in the room, initial average CO₂ concentration in the room, CO₂ concentration for the input airflow, CO₂ concentration for the output airflow, respectively. Furthermore, let v, V, q, n_t represent the airflow velocity, room volume, CO₂ emission rate per individual, and the number of occupants in the room. Note that all these extra parameters are known information of the environment. They may change over time, but their values at each time step are known. Then, we can write their relationship in the following equation $g(\cdot)$:

$$g(c_t, c_{in}^t, c_{out}^t) = c_t V - (c_0 V + \sum_t (c_{in}^t v \Delta t) + \sum_t (n_t q \Delta t) - \sum_t (c_{out}^t v \Delta t)) \quad (3)$$

Unfortunately, perfect CO₂ monitoring would require deploying multiple high-resolution and low-noise lab-level CO₂ sensors, which are prohibitively expensive for a typical commercial building consisting of hundreds of zones. By contrast, low-cost CO₂ sensors are readily available on the market with more noise and less accuracy (e.g., CCS881 as we will detail in Sec 5.3). We aim to incorporate the aforementioned physics equation to denoise CO₂ data collected from these low-cost sensors, such that data quality after our denoising algorithm can match the expensive counterparts.

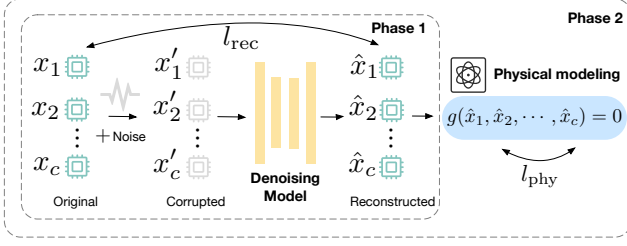


Figure 4: Physics-Informed denoising model framework. During training, we manually add noise to the original sensor data, and minimize a reconstruction loss to remove such noise. We also integrate a physics-based loss to ensure that reconstructed data obey the law of physics.

3.2.3 Application Scenario III: Air Handling in HVAC Control System. HVAC control systems manage critical parameters such as temperature, humidity, and air quality of commercial or residential buildings to ensure thermal comfort and optimal air quality for occupants while maintaining energy efficiency. Air Handling Units (AHUs), the core of an HVAC control system, take in outside air and return air from zones in the building, recondition it via heating or cooling, de-humidification, and then supply it to spaces. Downstream equipment such as Variable Air Volumes (VAVs) may further condition the air based on factors including occupancy, time of day, and specific temperature desired of the space. Figure 2 (right) illustrates an AHU in an example HVAC control system with both heating and cooling functionality. The AHU mixes a certain ratio of return air from the conditioned spaces and the fresh outside air, and reconditions the mixed air through cooling and heating coils and supply air to various spaces. We denote the enthalpy difference across the heating and cooling coils by ΔQ . The air's mass undergoing reconditioning is denoted as m , its specific heat capacity as c , the temperature of the supply air and mix air as T_{sa} and T_{mix} , respectively. We then derive the following heat transfer equation under the assumption of no additional heat source aside from the cooling and heating coils:

$$g(\Delta Q, m, c, T_{sa}, T_{mix}) = \Delta Q - mc(T_{sa} - T_{mix}) \quad (4)$$

Note that air mass m can be acquired through the airflow rate (measured in Cubic Feet per Minute, or CFM) going across the AHU. Furthermore, the air's specific heat capacity, c , is both humidity and temperature dependent.

In practice, both supply air temperature and mix air temperature measurements are subject to noise, which impedes accurate control in the HVAC system. Therefore, we incorporate the above equation to denoise the two temperature measurements, enhancing the accuracy and reliability of the HVAC control system.

4 METHODOLOGY

4.1 Model Framework

Noise Injection. We first follow the setting of typical supervised denoising methods to manually inject noise and train the denoising model to remove the injected noise. Given the unknown nature of real-world noise distributions, we first randomly sample noise from pre-defined distributions as simulations. This process is insufficient to capture the complex noise distributions, so we augment

the model training with physics modeling process illustrated later. When sampling the noise, we use Gaussian distribution with zero mean and variance of σ^2 as an example and compare with other noise distributions in Section 5.6 (Sensitivity Analysis). Mathematically, for each sample Y_i in our noisy dataset, we further sample Gaussian noise \mathbf{n}_i and add it upon Y_i to obtain data Z_i with a higher degree of noise, i.e., $Z_i = Y_i + \mathbf{n}_i$, $\mathbf{n}_i \sim \mathcal{N}(0, \sigma^2)$. Note that our physics modeling component does not pose any constraints or assumptions on the noise type. The inclusion of Gaussian additive noise is purely for simulation purposes to facilitate the training of the denoising autoencoder.

Naive Denoising Model. The naive denoising model performs optimization without modeling physical relationships. The denoising model takes data with manual noise injected Z_i as input, and the training objective is to output data with such manual noise removed, by minimizing the reconstruction loss, i.e.,

$$\arg\min_{\theta} \mathbb{E}_{Y_i \sim \mathcal{Y}, \mathbf{n}_i \sim \mathcal{N}} \ell_{\text{rec}}(f(Y_i + \mathbf{n}_i; \theta), Y_i), \quad (5)$$

$$\ell_{\text{rec}}(f(Y_i + \mathbf{n}_i; \theta), Y_i) = \|f(Y_i + \mathbf{n}_i; \theta) - Y_i\|_2^2. \quad (6)$$

The naive denoising model learns how to remove noise generated from our pre-defined noise distribution, e.g., Gaussian distribution. This is effective if the real noise follows the same noise distribution. However, in practice, the noise distribution is often much more complex and cannot be simply approximated by predefined patterns. As we will theoretically show in Section 4.2, the naive denoising method becomes suboptimal when the real noise distribution deviates from our pre-defined noise distributions. Therefore, we incorporate physics constraint to enhance denoising performance, enabling the model to adapt to more complex, real-world noise distributions.

Physics-Informed Denoising Model. Apart from the reconstruction loss in the naive denoising model, we incorporate another *physics-based loss* to overcome the limitation of the naive model. Recall in Section 3.1 (Problem Formulation) we mentioned that ground truth clean data satisfy physics constraint $g(\mathbf{x}_{i1}, \mathbf{x}_{i2}, \dots, \mathbf{x}_{ic}) = 0$, $i = 1, 2, \dots, N$. In an ideal scenario, if our denoising model can perfectly eliminate all noise, then the resultant output of this optimal denoising model $f(\cdot; \theta^*)$ will also adhere to the physics constraint:

$$g(f(\mathbf{z}_{i1}; \theta^*), f(\mathbf{z}_{i2}; \theta^*), \dots, f(\mathbf{z}_{ic}; \theta^*)) = 0, i = 1, 2, \dots, N, \quad (7)$$

which can be simplified as $g(f(\mathbf{Z}_i; \theta^*)) = 0$ or $g(f(Y_i + \mathbf{n}_i; \theta^*)) = 0$. Therefore, we can exploit the physics constraints intrinsic to an optimal denoising model and formulate the constrained optimization problem as

$$\begin{aligned} &\arg\min_{\theta} \mathbb{E}_{Y_i \sim \mathcal{Y}, \mathbf{n}_i \sim \mathcal{N}} \ell_{\text{rec}}(f(Y_i + \mathbf{n}_i; \theta), Y_i), \\ &\text{s.t. } g(f(Y_i + \mathbf{n}_i; \theta)) = 0. \end{aligned} \quad (8)$$

Enforcing the physics equation as strict constraints makes the model difficult to optimize. Instead, we include the constraint as a *soft regularization term in the loss function*. Specifically, the loss function for the physics-informed denoising model is a combination of reconstruction loss and physics-based loss:

$$\arg\min_{\theta} \mathbb{E}_{Y_i \sim \mathcal{Y}, \mathbf{n}_i \sim \mathcal{N}} \ell(f(Y_i + \mathbf{n}_i; \theta), Y_i), \quad (9)$$

$$\ell = \ell_{\text{rec}} + \lambda \ell_{\text{phy}}, \quad (10)$$

$$\ell_{\text{phy}}(f(Y_i + \mathbf{n}_i; \theta)) = \|g(f(Y_i + \mathbf{n}_i; \theta))\|_2^2. \quad (11)$$

The physics-based loss offers additional prior knowledge about the ground truth data and better guides the learning process of the denoising model, especially when we don't have access to the ground truth distributions. We find in the experiment section that the physics constraint works for both first-order terms and higher-order terms. Here, λ is the ratio that balances the two losses. We find that an adaptive balancing strategy that keeps these two losses always at the same orders of magnitude during training is empirically more effective than setting a fixed ratio, as we will illustrate in Section 5.6.

PILOT Framework. We design a two-phase training framework for the final denoising model. Specifically, we first optimize the model only with reconstruction loss as a pre-training phase. The two-stage training ensures that training is not biased by the physics loss at the beginning, as the initial exclusive focus on optimizing the physics loss can lead to overly simplistic and trivial solutions, such as all-zero predictions. Two-stage training mitigates this by serving as a warm-up period and providing better denoising model initialization. By learning to reconstruct the data, the model better adapts to the underlying data distribution. In the second phase, we combine both reconstruction loss and physics-based loss to optimize the model. The physics-based loss corrects and guides the model for enhanced denoising performance. To choose the denoising model backbone, we keep both efficacy and efficiency in mind, ensuring that denoising models can run on edge devices with limited compute. Therefore, we use one-dimensional Convolutional Neural Networks (CNN) as an example, as they are relatively lightweight with superior capability in extracting sensor features.

4.2 Theoretical Analysis

In this section, inspired by previous works in [20, 66], we analytically show that when the inherent noise of our model input has a non-zero mean, training with naive denoising model would be insufficient to recover the ground truth data distribution.

Corollary 1: Let \mathbf{y} be observed noisy data corrupted from clean data \mathbf{x} by noise ϵ with non-zero mean η : $\mathbf{y} = \mathbf{x} + \epsilon$, $\epsilon \sim \mathcal{E}$. Let \mathbf{z} be manually corrupted data from \mathbf{y} by pre-defined noise \mathbf{n} with zero mean: $\mathbf{z} = \mathbf{y} + \mathbf{n}$, $\mathbf{n} \sim \mathcal{N}$. Assume $\mathbb{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}) = \mathbb{E}_{\mathbf{y}|\mathbf{x}}(\mathbf{y}) = \mathbf{x} + \eta$, and the variance of \mathbf{y} is σ_y^2 . Then the following equation holds true:

$$\begin{aligned} & \mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \epsilon \sim \mathcal{E}, \mathbf{n} \sim \mathcal{N}} \|f(\mathbf{x} + \epsilon + \mathbf{n}; \theta) - \mathbf{x}\|_2^2 \\ &= \mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \epsilon \sim \mathcal{E}, \mathbf{n} \sim \mathcal{N}} \|f(\mathbf{x} + \epsilon + \mathbf{n}; \theta) - (\mathbf{x} + \epsilon)\|_2^2 \\ &\quad - \sigma_y^2 + 2\eta \mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \epsilon \sim \mathcal{E}, \mathbf{n} \sim \mathcal{N}} (f(\mathbf{x} + \epsilon + \mathbf{n}; \theta) - \mathbf{x}). \end{aligned} \quad (12)$$

Since $\eta \neq 0$, optimizing $\mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \epsilon \sim \mathcal{E}, \mathbf{n} \sim \mathcal{N}} \|f(\mathbf{x} + \epsilon + \mathbf{n}; \theta) - (\mathbf{x} + \epsilon)\|_2^2$ is not equivalent to the ideal goal of optimizing against ground truth, i.e., $\mathbb{E}_{\mathbf{x} \sim \mathcal{X}, \epsilon \sim \mathcal{E}, \mathbf{n} \sim \mathcal{N}} \|f(\mathbf{x} + \epsilon + \mathbf{n}; \theta) - \mathbf{x}\|_2^2$. In summary, the corollary states that the naive denoising model's optimization target fails to completely mitigate the inherent noises. Therefore, we incorporate physics-based loss to augment the denoising performance, particularly when underlying noise distributions are inaccessible.

5 EVALUATION

5.1 Experimental Setup

We use a one-dimensional convolutional neural network with four layers as the backbone for our denoising model. The respective

Table 1: Physics alignment results for OxIOD dataset. We bold the best and underline the second best. PILOT aligns the best with both acceleration and angular velocity.

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

kernel sizes for each layer are set to 7, 5, 3, 3. We use Adam optimizer with learning rate $1e^{-4}$ and batch size 16. For each dataset $\mathcal{D}_X = \{\mathbf{X}_i\}_{i=0}^N$, we first compute how well each sample aligns with the physics equation. Specifically, the alignment for the i -th sample is the L2 norm $a_i = \|g(\mathbf{X}_i)\|_2^2$. We select samples with the top 50% smallest a_i as our training set. These samples align more coherently with physics equations and potentially carry less noise. The remaining samples are used as the test set. The applications we have studied are mostly stationary systems, so the model should be able to generalize to new test data in practice. We conduct the experiments in PYTORCH with NVIDIA RTX A6000 (with 48GB memory), AMD EPYC 7452 32-Core Processor, and Ubuntu 18.04.5 LTS. We also implement our method on edge device (Raspberry Pi 4) and evaluate its efficiency in Section 5.7. We tune the hyper-parameters of both PILOT and baselines to minimize loss on the training set, and then evaluate on the test set after hyper-parameter tuning.

We compare PILOT with both statistical methods and recent deep learning denoising methods as follows (including two self-supervised methods DIP and N2N):

- **Gaussian Filter** employs a Gaussian filter with standard deviation σ for the kernel to smooth the sensor data.
- **DWT** [11] decomposes the signal using Discrete Wavelet Transform (DWT) and chooses the most energetic coefficients to reconstruct the denoised signal.
- **DnCNN** [63] is a convolutional neural network-based denoising model that exploits residual learning and batch normalization to boost the denoising performance.
- **TSTNN** [51] is a Transformer-based model for end-to-end speech denoising. It is composed of a feature-mapping encoder, a two-stage Transformer model to extract local and global information, a masking module and a decoder to reconstruct denoised speech.
- **Deep Image Prior (DIP)** [49] exploits the structure of the neural network as priors for denoising data. The method fits the model to the noisy data with early exit, so the network captures dominant patterns while ignoring the random noise.
- **Neighbor2Neighbor (N2N)** [20] applies random neighbor subsampler on noisy data to generate input and target for the network. It also proposes a regularizer as additional loss for performance augmentation.

We evaluate the performance from three perspectives. First of all, if we have collected data of higher quality using other sources

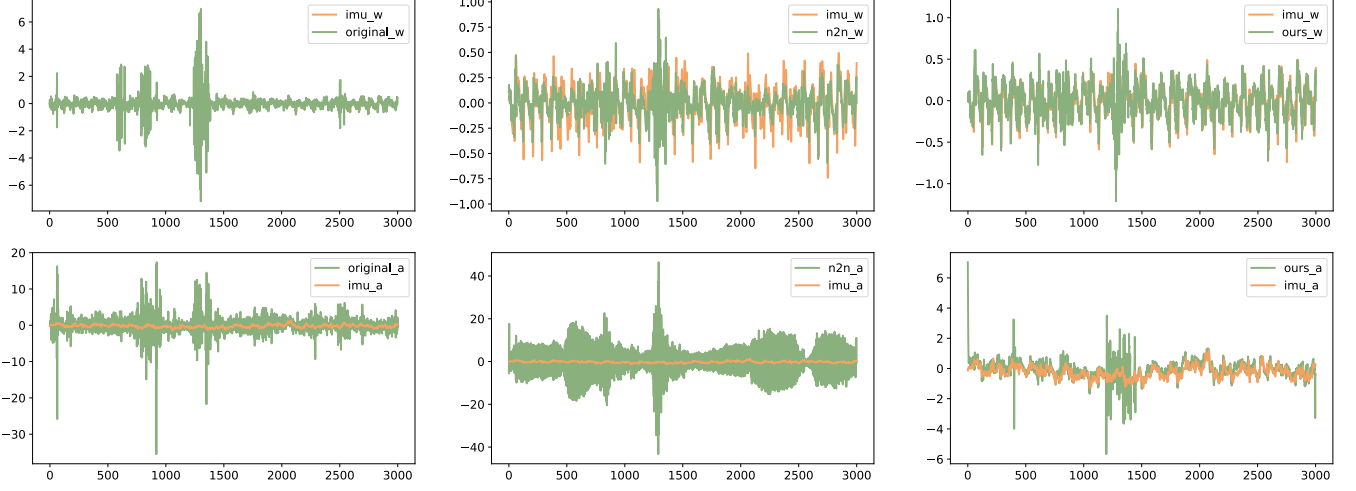


Figure 5: Top: Angular velocity from IMU (imu_w) and from first-order derivative calculation of orientation based on original data (original_w)/N2N(n2n_w)/PILOT (ours_w). Bottom: Acceleration from IMU (imu_a) and from second-order derivative calculation of location based on original data (original_a)/N2N(n2n_a)/PILOT (ours_a). X-axis: timestep, Y-axis: meter.

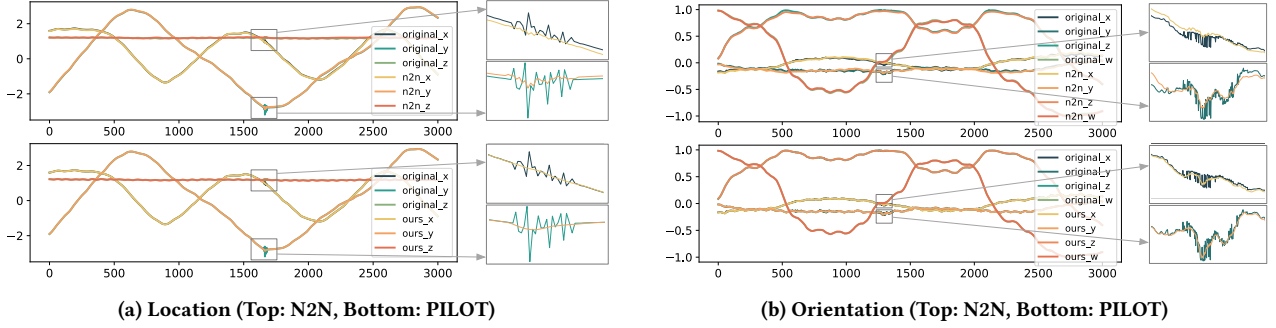


Figure 6: We compare PILOT with the best performing baseline N2N for qualitative denoising performance of (a) location and (b) orientation. We also zoom in for regions with higher noise levels for better illustration. PILOT generates smoother results which also align more closely with the original locations and orientations. X-axis: timestep, Y-axis: meter.

(e.g., more expensive and accurate sensors), we would regard these higher-quality data as approximate ground truth and compute Mean Square Error (MSE) and Mean Absolute Error (MAE) between the denoised data \hat{X} and the higher-quality data X as the first metric, noted as “Reconstruction Performance”. Specifically, Reconstruction $MSE = \|\hat{X} - X\|_2^2$, and Reconstruction $MAE = \|\hat{X} - X\|_1$. The second metric investigates alignment of denoised data with the governing physics equations, noted as “Physics Alignment”. Specifically, Physics $MSE = \|g(\hat{X})\|_2^2$, and Physics $MAE = \|g(\hat{X})\|_1$. Lastly, if the denoised data are further applied in downstream applications (e.g., inertial navigation system), we also evaluate “Downstream Performance” based on the metrics of interest for the corresponding downstream task. Measuring downstream performance helps us understand the full impact of denoising on real-world applications.

5.2 Application I: Inertial Navigation

5.2.1 Dataset. We use OxIOD [6] dataset for inertial navigation experiments. OxIOD collects accelerometer and gyroscope data (100 Hz) mostly by IMUs (InvenSense ICM20600) in iPhone 7 plus. A Vicon motion capture system (10 Bonita B10 cameras) is used to

record the locations and orientations. The total walking distance and recording time of the dataset are 42.5 km and 14.72 h. The data collection protocol is designed to simulate natural pedestrian movement within an indoor environment equipped with motion capture system. When collecting the data, a pedestrian walks naturally inside a room with motion capture system, carrying the phone in hand, in the pocket, in the handbag, or on the trolley.

5.2.2 Physics Alignment. We compare the physics alignment performance in Table 1. Specifically for this application, we compute the misalignment for acceleration and angular velocity. Adapting our experimental approach for the specificities of inertial navigation, we only denoise the location and orientation data, and leave IMU data in its original form. This is because the downstream inertial navigation model’s training requires integration over the IMU, thereby inherently smoothing and denoising the IMU data. Since our model only denoises the location and orientation data, we model the physics relationships between the denoised location, orientation, and the original IMU. More specifically, acceleration MSE in Table 1 is computed as the discrepancy between accelerometer

Table 2: Inertial navigation performance on OxIOD dataset with two inertial navigation models (IONet and RoNIN). We bold the best and underline the second best. When applying the denoised data to downstream inertial navigation task, PILOT yields the best performance given two different inertial navigation models (IONet and RoNIN).

Model	IONet [5]						RoNIN [19]					
	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	<u>0.0093</u>	0.0314	0.3076	0.8194	0.0180	0.0621	<u>0.0090</u>	0.0297	<u>0.2472</u>	<u>0.6337</u>
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	<u>0.0200</u>	0.0235	0.0144	0.0193	<u>0.3001</u>	<u>0.7891</u>	<u>0.0177</u>	0.0213	0.0139	<u>0.0176</u>	0.2476	0.6598
TSTNN	0.2857	0.3250	0.0935	0.2348	<u>0.6496</u>	<u>1.6575</u>	<u>0.2865</u>	0.3253	0.0938	<u>0.2352</u>	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	<u>0.0144</u>	0.0183	<u>0.0191</u>	0.3151	0.8317	0.0224	<u>0.0122</u>	0.0182	<u>0.0176</u>	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

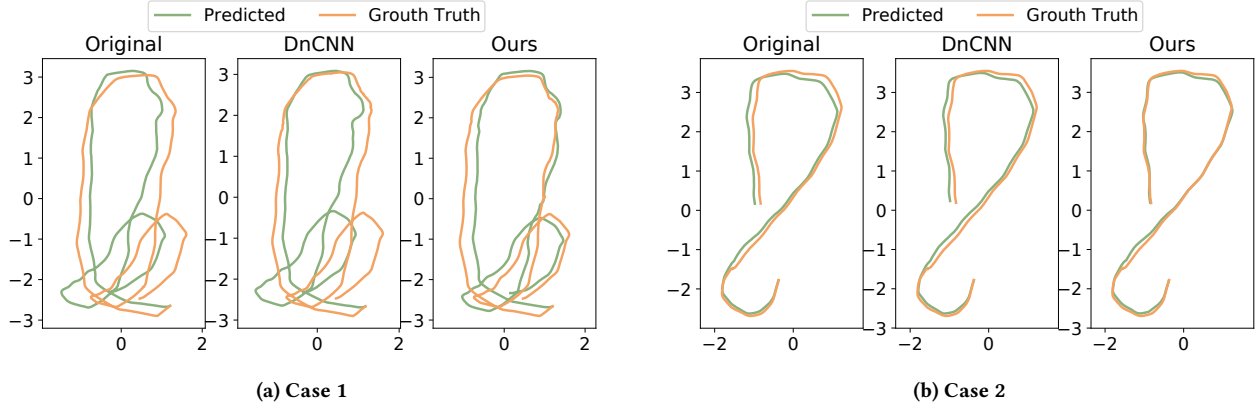


Figure 7: Downstream inertial navigation performance on OxIOD dataset with RoNIN model. RoNIN trained on denoised data of PILOT reconstructs trajectories closest to the ground truth trajectories. X-axis: meter, Y-axis: meter.

data from IMU (a) and the second-order derivative of the denoised location subtracted by gravity constant and multiplied by rotation matrix ($R_q^T(d^2p/dt^2 - g_0)$), as shown in Equation 1. Similarly, angular velocity MSE is computed as the difference between gyroscope data from IMU multiplied by orientation ($q \otimes w/2$) and the first-order derivative of denoised orientation (dq/dt), as in Equation 2.

From Table 1, we can see that while most denoising methods reduce the physics MSE compared with original data without denoising, *PILOT significantly reduces the physics MSE by two orders of magnitude*. Accompanied by these quantitative results, we also offer qualitative illustrations in Figure 5. We present a visual comparison of the direct IMU measurements and derived accelerations and angular velocities. When comparing PILOT with original and denoised data from the best-performing baseline, we observe that PILOT-denoised data provide the best physics alignment for both angular velocity and acceleration.

5.2.3 Reconstruction Performance. For the inertial navigation application, we don't have access to ground truth location and orientation data. Therefore, we qualitatively compare the reconstruction performance of PILOT and the best-performing baseline in Figure 6. We compare the reconstruction performance for both location and orientation, with detailed zoom-ins on regions exhibiting higher

noise levels for more clear illustration. We observe that PILOT effectively reduces the noise in the original data, while simultaneously reconstructing results that show greater alignment and coherence with the original locations and orientations.

5.2.4 Downstream Performance. We also use the denoised data to train downstream inertial navigation model, and compare PILOT and baselines in terms of downstream task performance. We use IMU data, denoised location and orientation data as training pairs for the following inertial navigation models:

- **IONet [5]:** IONet uses deep recurrent neural networks to learn location transforms in polar coordinates from raw IMU data, and construct inertial odometry.
- **RoNIN [19]:** RoNIN regresses velocity given IMU with coordinate frame normalization and robust velocity losses. We measure the downstream performance by the following three metrics, following previous works [19].
- **Velocity:** We compute MSE of predicted velocity (x, y, and z-axis, along with the mean velocity of all three axes).
- **Absolute Translation Error (ATE):** ATE is defined as the Root Mean Squared Error (RMSE) between the entire estimated and ground truth location trajectory.
- **Relative Translation Error (RTE):** RTE is defined as the Root

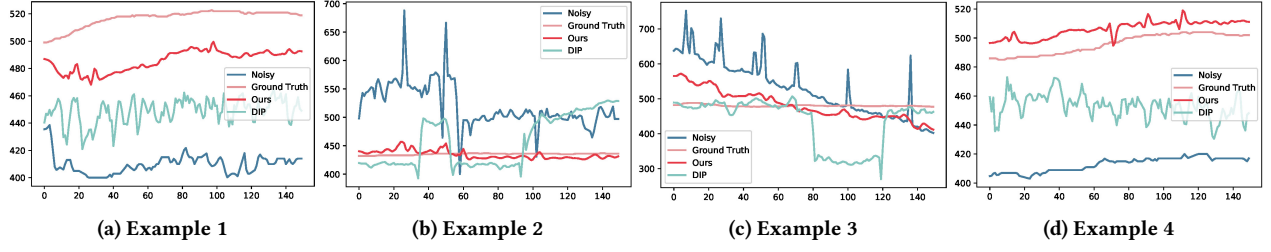


Figure 8: We compare the CO₂ denoising results of PILOT and the best performing baseline DIP. PILOT denoises CCS881 results closest to the ground truth (K30 measurements). X-axis: timestep, Y-axis: ppm.

Mean Squared Error (RMSE) over a fixed time interval. We choose 1 minute in our experiment as in previous works [19].

We compare PILOT with baselines on downstream inertial navigation task using IONet and RoNIN (Table 2) as the downstream models. We can observe that PILOT best predicts the velocity and locations, regardless of the specific inertial navigation model. In Figure 7, we also qualitatively compare PILOT with the best-performing baseline using RoNIN as the inertial navigation model. We can see PILOT’s denoised data offer RoNIN the best trajectory reconstruction performance relative to both the original data and that from the best-performing baseline.

5.3 Application II: CO₂ Monitoring

5.3.1 Dataset Collection. In this experiment, we deployed two pairs of CO₂ sensors (K30 and CCS811) in a typical graduate student office environment. K30 is a highly accurate Non-Dispersive Infra-Red (NDIR) CO₂ sensor manufactured by SenseAir which costs \$100 per unit. It features an accuracy of ± 30 ppm or $\pm 3\%$ of measurements and a high repeatability of ± 20 ppm or $\pm 1\%$ of measurements [50]. The NDIR technology [39] utilizes the unique property of CO₂ molecules — their significant absorption of infrared (IR) light in the vicinity of $4.2 \mu\text{m}$ wavelength. When a gas sample is illuminated with light of this particular wavelength, the concentration of CO₂ can be deduced by examining the fraction of light absorbed. We regard K30 readings as the ground truth of CO₂ concentration.

In comparison, CCS811 [1] is a substantially cheaper Metal-Oxide (MOX) gas sensor, priced at less than \$10 per unit, which computes equivalent CO₂ readings based on hydrogen gas readings. A MOX gas sensor works by measuring and analyzing changes in the conductivity of the gas-sensitive MOX semiconductor layer(s) at vari-

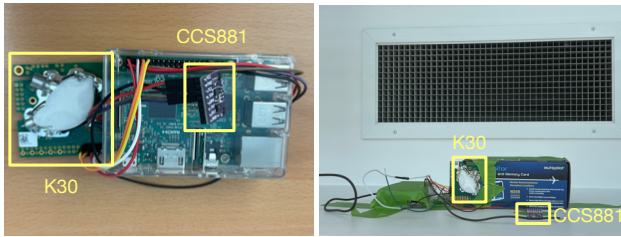
Table 3: Reconstruction and physics alignment results on CO₂ dataset. We bold the best and underline the second best.

Model	Recons (1×10^6 ppm)		Physics (1×10^6 ppm)	
	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	<u>0.0841</u>	<u>0.0006</u>	<u>0.0018</u>	<u>0.0308</u>
N2N	0.4396	0.0018	0.0085	0.0789
PILOT	0.0371	0.0004	0.0012	0.0200

ous gas exposure [10]. It has no verifiable accuracy or repeatability guarantee and is usually not recommended for laboratory use.

For each pair of these two sensors, we connect both of them to a single Raspberry Pi board, which collects data through UART protocol and I2C interface from K30 and CCS811 respectively, at an interval of 4 seconds. The collected readings are transmitted to a remote InfluxDB instance for storage and further analysis. We placed these two pairs of CO₂ sensors as follows. One pair is adjacent to the HVAC exhaust vent and the other pair is at the center of the room, to capture the relevant variables in Equation 3 as illustrated in Figure 9. We set inflow CO₂ concentration in Equation 3 to 440 ppm based on empirical observation.

5.3.2 Reconstruction and Physics Performance. For the application of CO₂ monitoring, data collected from high-precision K30 sensors serve as the benchmark for calculating the reconstruction performance. We also evaluate the physics alignment by how well our denoised data match Equation 3. As shown in Table 3, PILOT outperforms all baselines, showcasing the lowest reconstruction and physics alignment errors. We also qualitatively compare PILOT with the best-performing baseline in Figure 8. We can observe that the denoised data by PILOT align the closest to the ground truth (K30 data) compared with original noisy data as well denoised data from the best-performing baseline.



(a) Deployment on desk

(b) Deployment by vent

Figure 9: We connected a pair of two different CO₂ sensors (K30 and CCS 881 respectively) on a Raspberry Pi. We deployed (a) one pair on the desk in the center of the office and (b) the other pair close to the exhaust vent to monitor CO₂ level.

5.4 Application III: Air Handling in HVAC Control System

5.4.1 Dataset Collection. We collected data points of relevant variables in Equation 4 from an AHU serving a lecture hall on campus. We chose this particular AHU based on the following two criteria. Firstly, this AHU is equipped with both heating and cooling

thermal submeters (Figure 10a), ensuring the availability of ΔQ in Equation 4. Secondly, none of the zones served by this AHU has any reheat unit. In theory, this guarantees that our assumption holds true that heating and cooling coils are the only heat sources. We collected the data through the campus building management system (BMS) for 18 consecutive days, with readings taken at 1-second intervals. We plan to collect more long-term data as a future work.

The mixed air temperature and supply air temperature are measured by copper-thread averaging duct sensors (costing \$150 for each) within the respective supply and exhaust ducts of this AHU, as shown in Figure 10b. These sensors give the average temperature based on the fact that the electrical resistance of copper changes in a predictable way with temperature changes. However, this method is prone to inaccuracy and lacks repeatability due to challenges associated with maintenance and cleaning, as well as the uneven distribution of the thread within the duct. A feasible yet costly alternative solution is to use multiple single-unit temperature sensors evenly distributed throughout the duct. These sensors are more expensive and harder to deploy. To evaluate the denoising methods, we carried this setup by deploying two dual-probe high-accuracy, industry-level temperature sensors in the supply duct, each costing \$200, and took the average of readings from each probe to get a highly reliable ground truth temperature measurement.

5.4.2 Reconstruction and Physics Performance. For the HVAC control system application, we measure the denoising performance by both reconstruction performance and physics alignment. For reconstruction performance, we deploy high-precision temperature sensors and utilize the collected data as ground truth temperature to assess the reconstruction performance of different denoising methods. For physics alignment, we compute the MSE and MAE between ΔQ and $mc\Delta T$ in Equation 4. As shown in Table 4, *PILOT achieves state-of-the-art performance and shows the lowest errors for both reconstruction performance and physics alignment.* We also provide qualitative comparisons to supplement our evaluation. Figure 11 showcases two example reconstructions of supply air temperature

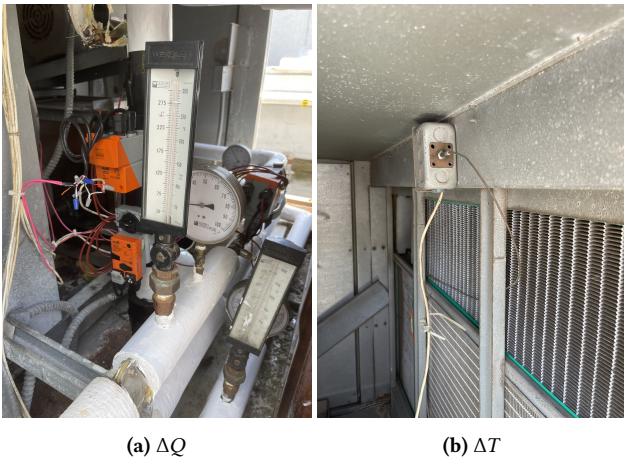


Figure 10: Sensor deployments in HVAC systems. (a) We measure heat difference ΔQ through the water temperature difference. (b) We measure the air temperature differences with a copper thread averaging sensor.

Table 4: Reconstruction and physics alignment on HVAC dataset. We bold the best and underline the second best. PILOT best denoises HVAC data in terms of both reconstruction accuracy as well as physics alignment.

Model	Reconstruction (K)		Physics (K)	
	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	<u>0.3284</u>	<u>0.4786</u>	50.380	6.7241
TSTNN	2.2980	1.1980	32.012	3.7260
DIP	3.7336	1.5098	<u>29.747</u>	3.7410
N2N	1.1830	0.9522	31.748	<u>3.6990</u>
PILOT	0.1994	0.3454	14.081	3.1600

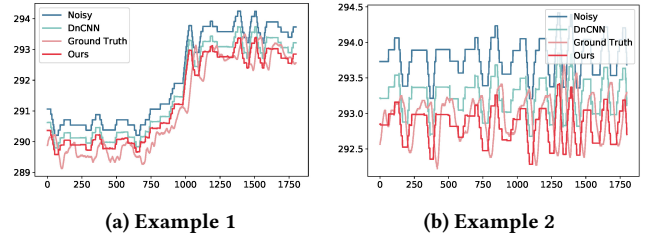


Figure 11: Two example reconstructions for supply air temperature. PILOT's outputs best align with the ground truth compared with original noisy measurements and the best-performing baseline. X-axis: timestep, Y-axis: K.

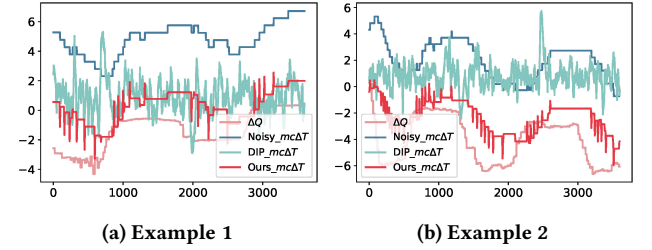


Figure 12: Two example alignment between ΔQ and $mc\Delta T$. PILOT achieves the closest alignment compared with original data and the best baseline. X-axis: timestep, Y-axis: K.

data. Denoised data from PILOT exhibits the best alignment with the ground truth data compared with original data, as well as the best-performing baseline DnCNN. In Figure 12, we also present two examples of physics alignment. We see that denoised data of PILOT best align with the physics equation compared with original noisy data and the best-performing baseline DIP.

5.5 Ablation Study

To study the effects of different components of our model, we separately remove the physics-based loss, reconstruction loss, and pre-training phase, and evaluate PILOT in Table 5. For Inertial Navigation System (INS), we compare the physics alignment of both acceleration (MSE_a , MAE_a) and angular velocity (MSE_w , MAE_w). For CO₂ monitoring and HVAC control, we compare both reconstruc-

Table 5: Ablation Study of PILOT. We bold the best and underline the second best. Removing either physics-based loss, reconstruction loss or pre-training phase would negatively affect the performance, demonstrating the effectiveness of all three components.

Task	Metrics	w/o l_{phy}	w/o l_{rec}	w/o pre-train	PILOT
INS	MSE _a	316.5	259.1	<u>20.70</u>	1.8695
	MAE _a	8.342	8.759	<u>2.2930</u>	0.6372
	MSE _w	0.3579	0.2845	<u>0.1380</u>	0.0380
	MAE _w	0.1899	0.3352	<u>0.1291</u>	0.0690
CO ₂	MSE _{rec}	0.4641	<u>0.0568</u>	0.0724	0.0371
	MAE _{rec}	0.0139	<u>0.0051</u>	0.0074	0.0047
	MSE _{phy}	0.0194	<u>0.0021</u>	0.0023	0.0012
	MAE _{phy}	0.0781	<u>0.0264</u>	0.0330	0.0200
HVAC	MSE _{rec}	0.3314	0.5314	<u>0.2686</u>	0.1994
	MAE _{rec}	0.4709	0.6128	<u>0.4670</u>	0.3454
	MSE _{phy}	50.22	14.69	<u>14.469</u>	14.081
	MAE _{phy}	6.713	3.263	<u>3.203</u>	3.1600

tion (MSE_{rec}, MAE_{rec}) and physics alignment (MSE_{phy}, MAE_{phy}).

We note that by removing the physics-based loss, PILOT degenerates to a naive denoising model, which is not sufficient to capture the complex noise distribution in real-world sensor data. Therefore, compared with PILOT, MSE and MAE of both reconstruction and physics alignment increase after removing the physics-based loss. We also observed that a higher degree of precision in the physics model and higher frequency sensor data sampling (resulting in fewer synchronization errors), lead to larger performance improvements. For example, the relationships in inertial navigation physics are more precisely captured than with CO₂ or temperature systems, and the IMU sensors are sampled at higher frequencies than CO₂ sensors. Correspondingly, we observe greater performance gains from employing physics equations than other applications. Secondly, reconstruction loss is important as it facilitates the model’s ability to learn data distribution. In its absence, the model may potentially generate trivial outputs that, while satisfying physics constraints, neglect the actual data distribution (e.g., outputs of pure zeros). Lastly, the pre-training phase serves as an essential warm-up period to help the model better adapt to the underlying data distribution. To summarize, the physics-based loss, reconstruction loss, and pre-training phase collectively contribute to the overall efficacy of the model, and their presence is crucial for optimal performance.

5.6 Sensitivity Analysis

5.6.1 Balance between Reconstruction Loss and Physics-Based Loss.

In our experiments, we adopt an adaptive strategy to adjust the ratio between reconstruction loss (l_{rec}) and physics-based loss (l_{phy}) during training. Denote the ratio as $\lambda = l_{\text{rec}}/l_{\text{phy}}$. More specifically, for each iteration, we compute the mean value of the reconstruction loss, and adaptively adjust λ such that $(\lambda \cdot l_{\text{phy}})$ aligns with the same order of magnitude as l_{rec} . Consequently, this adaptive approach avoids overemphasizing either the reconstruction loss or physics-based loss throughout the entire training process. To validate the effectiveness of adaptive loss ratios, we compare PILOT with its counterparts trained with fixed loss ratios, varying from 0.1, 1 to

Table 6: Efficiency Analysis on Raspberry Pi 4.

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

10. As shown in Figure 13, adaptive loss ratio (denoted as “ADA”) yields the best performance compared with different fixed ratios.

5.6.2 Manually Injected Noise Types. We also study the effect of different noise types injected for optimizing the reconstruction loss. In our experiments, we use Gaussian noise as an example injected noise. We compare PILOT with its counterparts trained with manually injected uniformly distributed noise, as well as manually injected zero masks. As shown in Figure 14, injecting random noise (Gaussian or Uniform noise) performs better than applying zero masking, as random noise aligns closer to real-world noise distribution compared with zero masking. Moreover, the model is less sensitive to the particular noise types applied, such as Gaussian noise or uniformly distributed noise.

5.7 Efficiency Analysis for Edge Devices

As many denoising methods for sensor data run on edge devices, we explore the time and memory efficiency of PILOT in Table 6 using Raspberry Pi 4 as an example edge device. We deploy the inertial navigation denoising application through the Tensorflow Lite framework on Raspberry Pi 4. For time efficiency, PILOT can denoise a sequence of 100 readings (corresponding to a sequence of 1 second) within 4 milliseconds. To ensure the reliability of our evaluation, this experiment has been repeated a thousand times to obtain the average inference time. Therefore, PILOT is capable of performing sensor data denoising in real time. For memory efficiency, PILOT is a lightweight CNN model with just 270K parameters and 284 KB model size. Moreover, it demonstrates an average CPU utilization of only 25% during the entire inference process. In summary, PILOT is both time and memory efficient for real-time operations on edge devices.

6 DISCUSSIONS AND FUTURE WORK

In this section, we discuss potential future directions of PILOT.

Broader Applicability to Other Sensing Systems We acknowledge that the need for clearly formulated physics relationships is one limitation of our work. However, we note that apart from the three applications we have explored, in practice numerous sensing systems have underlying physics relationships, and we recognize this opportunity for extended exploration in future work. We discuss a few potential use cases as some examples for the model’s wider applicability:

- **Power System.** In a power system, the most classical Ohm’s law describes the relationships between power, current and voltage. Electronic devices measuring current or voltage are often subject to environmental disruptions, such as temperature fluctuations, resulting in measurement noise. Utilizing physics priors like Ohm’s Law, PILOT can offer a promising solution to mitigate these noise-related inaccuracies.
- **Weather Monitoring.** Weather monitoring and forecasting involve calculations with respect to air pressure, temperature, wind

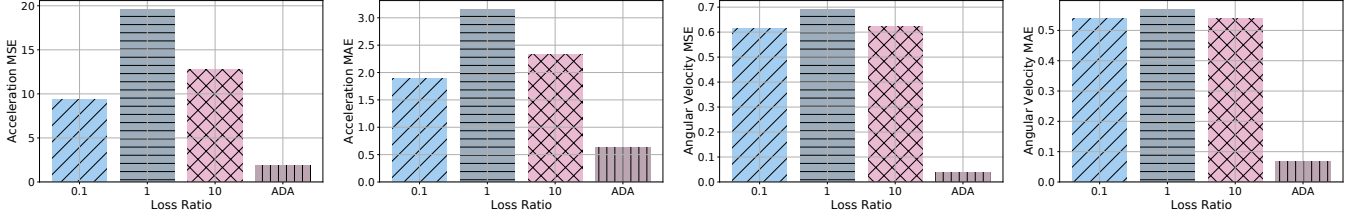


Figure 13: Performance analysis on the ratio of reconstruction loss over physics-based loss. Adaptive loss ratio yields better performance compared with fixed ratios.

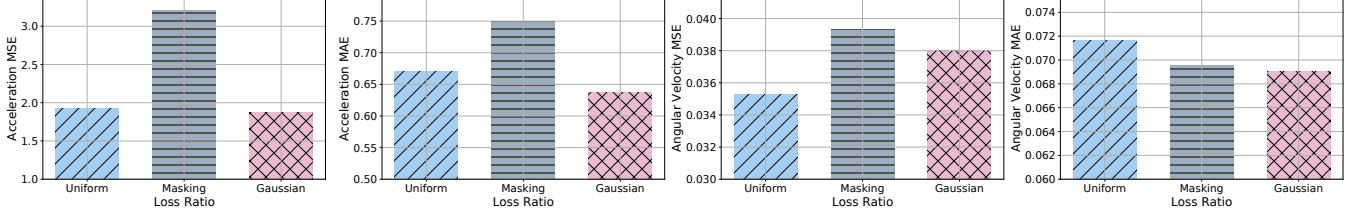


Figure 14: Performance analysis on the types of manually injected noise for the reconstruction process. Applying random noise performs better than zero masking.

speed, etc. These properties can be modeled as fluid dynamics problems, which are often characterized by the Navier-Stokes equations linking velocity and pressure. Environmental sensors responsible for capturing data like wind speed or pressure are susceptible to noise originating from environmental influences or sensor sensitivity. PILOT can leverage physics-based priors such as the Navier-Stokes equations to denoise these collected data, enhancing the accuracy of weather monitoring.

- **Localization in Autonomous Systems.** In autonomous systems, we employ a combination of GPS, Lidar, Radar and IMU sensors to localize autonomous driving cars or drones. These sensor readings can be noisy due to atmospheric effects, multi-path propagation, etc. PILOT is capable of applying the laws of motion (e.g., acceleration is the second derivative of location) to denoise location data collected from these multiple sources.

Uncertainty Quantification Our future work also includes incorporating uncertainty quantification (UQ) into the denoising process. The integration of UQ aims to provide a holistic understanding of noise structure, delivering not just a point estimate for the denoised data but also the model’s confidence in that estimate, avoiding overconfident and potentially inaccurate inferences. The integration may also help quantify the effects of denoising with respect to the precision of the model or synchronization errors. Uncertainty quantification can be achieved by adopting a probabilistic approach to extend the current model architecture to output not just a single denoised signal but a distribution over possible denoised signals.

FPGA Acceleration We have implemented our denoising method on edge devices like Raspberry Pi and demonstrated its time and memory efficiency. To take this step further, we plan to explore hardware acceleration as Field-Programmable Gate Arrays (FPGA) co-processors to explore the possibility and cost of such edge inference. With FPGA’s reprogrammable nature, we have the flexibility of prototyping and testing different configurations and parameters for the best possible performance and efficiency. Furthermore, FPGA allows us to seamlessly transition from prototype to subse-

quent production of Application-Specific Integrated Circuit (ASIC), which enable us to scale up our solution while reducing power consumption and production costs.

7 CONCLUSION

We presented a physics-informed denoising method, PILOT, for real-life sensing systems. We build upon the insight that measurements from different sensors are intrinsically related by known underlying physics principles. This approach allows the model to harness these physics constraints as guidance during training to improve denoising process. This paves the way for a more practical denoising solution, especially given the frequent challenges associated with acquiring ground truth clean data in sensing systems or understanding underlying noise patterns. Extensive experiments show the efficacy of PILOT in removing sensor noise across three representative real-world sensing systems. PILOT produces denoised results for low-cost sensors that align closely with high-precision and high-cost sensors, leading to a cost-effective denoising approach. PILOT is also lightweight and can enable real-time denoising on edge devices.

ACKNOWLEDGMENTS

Our work is supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. Our work is also supported by Qualcomm Innovation Fellowship and is sponsored by NSF CAREER Award 2239440, NSF Proto-OKN Award 2333790, NIH Bridge2AI Center Program under award 1U54HG012510-01, Cisco-UCSD Sponsored Research Project, as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

REFERENCES

- [1] ams OSRAM AG. 2016. CCS811 Datasheet. https://cdn.sparkfun.com/assets/learn_tutorials/1/4/3/CCS811_Datasheet-DS000459.pdf.
- [2] Irvan B Arief-Ang, Flora D Salim, and Margaret Hamilton. 2017. Da-hoc: semi-supervised domain adaptation for room occupancy prediction using co2 sensor data. In *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*. 1–10.
- [3] Alex Beltran and Alberto E Cerpa. 2014. Optimal HVAC building control with occupancy prediction. In *Proceedings of the 1st ACM conference on embedded systems for energy-efficient buildings*. 168–171.
- [4] Zhiwei Cao, Ruihang Wang, Xin Zhou, and Yonggang Wen. 2022. Reducio: model reduction for data center predictive digital twins via physics-guided machine learning. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 1–10.
- [5] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. 2018. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
- [6] Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, and Niki Trigoni. 2018. Oxiod: The dataset for deep inertial odometry. *arXiv preprint arXiv:1809.07491* (2018).
- [7] Mingkan Chen, Jingtao Sun, Kazushige Saga, Tomota Tanjo, and Kento Aida. 2020. An adaptive noise removal tool for iot image processing under influence of weather conditions. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 655–656.
- [8] Xinlei Chen, Xiangxiang Xu, Xinyu Liu, Shijia Pan, Jiayou He, Hae Young Noh, Lin Zhang, and Pei Zhang. 2018. Pga: Physics guided and adaptive approach for mobile fine-grained air pollution estimation. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 1321–1330.
- [9] A. M. Collier-Oxandale, J. Thorson, H. Halliday, J. Milford, and M. Hannigan. 2019. Understanding the ability of low-cost MOx sensors to quantify ambient VOCs. *Atmospheric Measurement Techniques* 12, 3 (2019), 1441–1460. <https://doi.org/10.5194/amt-12-1441-2019>
- [10] A. M. Collier-Oxandale, J. Thorson, H. Halliday, J. Milford, and M. Hannigan. 2019. Understanding the ability of low-cost MOx sensors to quantify ambient VOCs. *Atmospheric Measurement Techniques* 12, 3 (2019), 1441–1460. <https://doi.org/10.5194/amt-12-1441-2019>
- [11] Maria Luisa Lopes De Faria, Carlos Eduardo Cugnasca, and José Roberto Almeida Amazonas. 2017. Insights into IoT data and an innovative DWT-based technique to denoise sensor signals. *IEEE Sensors Journal* 18, 1 (2017), 237–247.
- [12] Joao Diogo Falcao, Prabh Simran S Baweja, Yi Wang, Akkarit Sangpetch, Hae Young Noh, Orathai Sangpetch, and Pei Zhang. 2021. PIWIMS: Physics Informed Warehouse Inventory Monitory via Synthetic Data Generation. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers*. 613–618.
- [13] Matthew Faulkner, Annie H Liu, and Andreas Krause. 2013. A fresh perspective: Learning to sparsify for detection in massive noisy sensor networks. In *Proceedings of the 12th international conference on Information processing in sensor networks*. 7–18.
- [14] Xiaohan Fu, Jason Koh, Francesco Fraternali, Dezhi Hong, and Rajesh Gupta. 2020. Zonal Air Handling in Commercial Buildings. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (Virtual Event, Japan) (BuildSys '20)*. Association for Computing Machinery, New York, NY, USA, 302–303. <https://doi.org/10.1145/3408308.3431113>
- [15] Han Gao, Luning Sun, and Jian-Xun Wang. 2021. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids* 33, 7 (2021), 073603.
- [16] Victor Garcia Satorras, Zeynep Akata, and Max Welling. 2019. Combining generative and discriminative models for hybrid inference. *Advances in Neural Information Processing Systems* 32 (2019).
- [17] Zhongkai Hao, Songming Liu, Yichi Zhang, Chengyang Ying, Yao Feng, Hang Su, and Jun Zhu. 2022. Physics-Informed Machine Learning: A Survey on Problems, Methods and Applications. *arXiv preprint arXiv:2211.08064* (2022).
- [18] Lixing He, Carlos Ruiz, Mostafa Mirshekari, and Shijia Pan. 2020. Scsv2: physics-informed self-configuration sensing through vision and vibration context modeling. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*. 532–537.
- [19] Sachini Herath, Hang Yan, and Yasutaka Furukawa. 2020. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3146–3152.
- [20] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. 2021. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14781–14790.
- [21] Jeya Vikranth Jeyakumar, Liangzhen Lai, Naveen Suda, and Mani Srivastava. 2019. SenseHAR: a robust virtual activity sensor for smartphones and wearables. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 15–28.
- [22] GE Karniadakis, IG Kevrekidis, L Lu, P Perdikaris, S Wang, and L Yang. 2021. Physics-informed machine learning: Nature Reviews Physics. (2021).
- [23] Daniel Kelshaw and Luca Magri. 2022. Physics-Informed Convolutional Neural Networks for Corruption Removal on Dynamical Systems. *arXiv preprint arXiv:2210.16215* (2022).
- [24] Minsung Kim, Salvatore Mandrà, Davide Venturelli, and Kyle Jamieson. 2021. Physics-inspired heuristics for soft MIMO detection in 5G new radio and beyond. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 42–55.
- [25] Kuldeep Kurte, Kadir Amasyali, Jeffrey Munk, and Helia Zandi. 2021. Comparative analysis of model-free and model-based HVAC control for residential demand response. In *Proceedings of the 8th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 309–313.
- [26] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. 2018. Noise2Noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189* (2018).
- [27] Shuheng Li, Jingbo Shang, Rajesh K Gupta, and Dezhong Hong. 2022. SQEE: A Machine Perception Approach to Sensing Quality Evaluation at the Edge by Uncertainty Quantification. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 277–290.
- [28] Miaomiao Liu, Sikai Yang, Wyssanie Chomsin, and Wan Du. 2022. Real-time tracking of smartwatch orientation and location by multitask learning. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 120–133.
- [29] Tiantian Liu, Ming Gao, Feng Lin, Chao Wang, Zhongjie Ba, Jinsong Han, Wenya Xu, and Kui Ren. 2021. Wavevoice: A noise-resistant multi-modal speech recognition system fusing mmwave and audio signals. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 97–110.
- [30] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. 2020. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters* 5, 4 (2020), 5653–5660.
- [31] Yang Liu, Zhenjiang Li, Zhidan Liu, and Kaishun Wu. 2019. Real-time arm skeleton tracking and gesture inference tolerant to missing wearable sensors. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 287–299.
- [32] Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, Peijun Zhao, Yasin Almaliglu, Pedro PB De Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. 2020. milliEgo: single-chip mmWave radar aided egomotion estimation via deep sensor fusion. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 109–122.
- [33] Wenjie Luo, Zhenyu Yan, Qun Song, and Rui Tan. 2021. Phyaug: Physics-directed data augmentation for deep sensing model transfer in cyber-physical systems. In *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021)*. 31–46.
- [34] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. 2020. Noisier2noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12064–12072.
- [35] Srinarayana Nagarathinam, Yashovardhan S Chati, Malini Pooni Venkat, and Arunchandar Vasani. 2022. PACMAN: physics-aware control MANager for HVAC. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 11–20.
- [36] Srinarayana Nagarathinam, Arunchandar Vasani, Venkata Ramakrishna P, Shiva R Iyer, Venkatesh Sarangan, and Anand Sivasubramanian. 2015. Centralized management of HVAC energy in large multi-AHU zones. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. 157–166.
- [37] Wisconsin Department of Health Service. 2023. Carbon Dioxide. <https://www.dhs.wisconsin.gov/chemical/carbondioxide.htm>.
- [38] Olufemi A Omitaomu, Vladimir A Protopopescu, and Auroop R Ganguly. 2011. Empirical mode decomposition technique with conditional mutual information for denoising operational sensor data. *IEEE sensors journal* 11, 10 (2011), 2565–2575.
- [39] Stefan Palzer. 2020. Photoacoustic-based gas sensing: A review. *Sensors (Basel)* 20, 9 (May 2020), 2745.
- [40] Jaeyeon Park, Woojin Nam, Jaewon Choi, Taeyeon Kim, Dukyong Yoon, Sukhoon Lee, Jeongyeup Paek, and JeongGil Ko. 2017. Glasses for the third eye: Improving the quality of clinical data analysis with motion sensor-based data filtering. In *Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems*. 1–14.
- [41] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378 (2019), 686–707.
- [42] Hongyu Ren, Russell Stewart, Jiaming Song, Volodymyr Kuleshov, and Stefano

- Ermon. 2018. Learning with weak supervision from physics and data-driven constraints. *AI Magazine* 39, 1 (2018), 27–38.
- [43] S N Rudnick and D K Milton. 2003. Risk of indoor airborne infection transmission estimated from carbon dioxide concentration. *Indoor Air* 13, 3 (Sept. 2003), 237–245.
- [44] Sheng Shen, Mahanth Gowda, and Romit Roy Choudhury. 2018. Closing the gaps in inertial motion tracking. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 429–444.
- [45] Sheng Shen, He Wang, and Romit Roy Choudhury. 2016. I am a smartwatch and i can track my user's arm. In *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*. 85–96.
- [46] Scott Sun, Dennis Melamed, and Kris Kitani. 2021. Idol: Inertial deep orientation-estimation and localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6128–6137.
- [47] Yifei Sun, Yuxuan Liu, Ziteng Wang, Xiaolei Qu, Dezhi Zheng, and Xinlei Chen. 2022. C-RIDGE: Indoor CO₂ Data Collection System for Large Venues Based on prior Knowledge. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 1077–1082.
- [48] Naoya Takeishi and Alexandros Kalousis. 2021. Physics-integrated variational autoencoders for robust and interpretable generative modeling. *Advances in Neural Information Processing Systems* 34 (2021), 14809–14821.
- [49] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2018. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9446–9454.
- [50] CO₂Meter.com. 2022. K30 Datasheet. https://cdn.shopify.com/s/files/1/0019/5952/files/DS-SenseAir-K30-CO2-Sensor-Revised-061722_96d4f229-643c-45b0-95a6-24efa252490d.pdf?v=1672160950.
- [51] Kai Wang, Bengbeng He, and Wei-Ping Zhu. 2021. TSTNN: Two-stage transformer based neural network for speech enhancement in the time domain. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7098–7102.
- [52] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. 2020. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1457–1466.
- [53] Rui Wang and Rose Yu. 2021. Physics-guided deep learning for dynamical systems: A survey. *arXiv preprint arXiv:2107.01272* (2021).
- [54] Ziqi Wang, Ankur Sarker, Jason Wu, Derek Hua, Gaofeng Dong, Akash Deep Singh, and Mani Srivastava. 2022. Capricorn: Towards Real-time Rich Scene Analysis Using RF-Vision Sensor Fusion. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 334–348.
- [55] Manuel Weber, Christoph Doblander, and Peter Mandl. 2020. Detecting building occupancy with synthetic environmental data. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 324–325.
- [56] Kevin Weekly, Nikolaos Bekiaris-Liberis, Ming Jin, and Alexandre M Bayen. 2015. Modeling and estimation of the humans' effect on the CO₂ dynamics inside a conference room. *IEEE Transactions on Control Systems Technology* 23, 5 (2015), 1770–1781.
- [57] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. 2020. Noisy-as-clean: Learning self-supervised denoising from corrupted image. *IEEE Transactions on Image Processing* 29 (2020), 9316–9329.
- [58] Shichao Xu, Yangyang Fu, Yixuan Wang, Zhuoran Yang, Zheng O'Neill, Zhaoran Wang, and Qi Zhu. 2022. Accelerate online reinforcement learning for building HVAC control with heterogeneous expert guidances. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 89–98.
- [59] Hang Yan, Qi Shan, and Yasutaka Furukawa. 2018. RIDI: Robust IMU double integration. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 621–636.
- [60] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. 2022. Learning physics constrained dynamics using autoencoders. *Advances in Neural Information Processing Systems* 35 (2022), 17157–17172.
- [61] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. 2022. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13167–13178.
- [62] Chi Zhang, Sanmukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. 2019. Building HVAC scheduling using reinforcement learning via neural network based model approximation. In *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 287–296.
- [63] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing* 26, 7 (2017), 3142–3155.
- [64] Xiyuan Zhang, Ranak Roy Chowdhury, Dezhi Hong, Rajesh K Gupta, and Jingbo Shang. 2023. Modeling Label Semantics Improves Activity Recognition. *arXiv preprint arXiv:2301.03462* (2023).
- [65] Jincao Zhu, Youngbin Im, Shivakant Mishra, and Sangtae Ha. 2017. Calibrating time-variant, device-specific phase noise for COTS WiFi devices. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. 1–12.
- [66] Magaiya Zhussip, Shakarim Soltanayev, and Se Young Chun. 2019. Extending Stein's unbiased risk estimator to train deep denoisers with correlated pairs of noisy images. *Advances in neural information processing systems* 32 (2019).