

FW-NKF: Frequency-Weighted Neural Kalman Filters

Annan Harun Dogan*, Berken Utku Demirel, Christian Holz

Abstract—Robust state estimation is central to robotic autonomy, yet classical Kalman filters struggle with frequency-dependent disturbances and model mismatch such as sensor vibrations, electromagnetic interference, and periodic noise. Although Deep Kalman Filter (DKF) variants extend the Extended Kalman Filtering (EKF) framework by learning latent transitions, they lack explicit mechanisms to suppress band-limited noise components that typically corrupt sensor measurements in real-world scenarios. We introduce the Frequency-Weighted Neural Kalman Filter (FW-NKF), a unified hybrid approach that embeds a causal spectral-shaping operator into the Kalman measurement residual and jointly learns observation, and transition networks. By adapting both the filter spectrum and the latent state representation, FW-NKF attenuates the noise-dominated frequency bands while capturing complex residual structures. We conduct extensive experiments on four heterogeneous benchmarks, including chaotic systems such as multi-dimensional Lorenz systems and full-body inertial pose estimation, and find a reduction in localization error of up to 10% as well as marked improvements in orientation accuracy. Our ablation studies confirm that frequency weighting and deep latent-state modeling contribute to overall performance.

I. INTRODUCTION

Real-world sensor measurements, including drone inertial measurement units (IMUs), wheel encoders, and wearable IMUs, are often corrupted by band-specific disturbances such as high-frequency vibration jitter, low-frequency bias drift, and periodic electromagnetic interference [1]. Downstream tasks such as trajectory planning, flight stabilization, and avatar animation in VR/AR demand state estimates that are both accurate and temporally coherent [2], [3], yet even moderate spectral contamination can degrade quality [4].

State estimation often relies on noisy IMU data, where integration drift causes small biases in acceleration or angular rate to accumulate over time. These small biases can quickly become large position or orientation errors, especially when external corrections such as GPS or visual markers are unavailable [5]. For example, in ground vehicles, this drift can result in missed turns or navigation failures. For micro-drones operating with tight agility constraints, even minor pose errors can destabilize flight, preventing successful obstacle avoidance or trajectory tracking. In human motion capture, drift in IMU-based pose estimation causes joint angle inconsistencies and body posture deviations, degrading the realism and accuracy of avatar representations in virtual reality [6].

The classical Kalman filter attains minimum mean-squared error (MMSE) optimality only when process and measurement noise are white, Gaussian, and temporally uncorrelated [7],

[8], [9]—conditions Maybeck famously called “fairy tales for undergraduates” [10]. When noise is instead colored, periodic, or band-limited [11], [12], [13], the filter’s fixed covariance model can neither capture nor suppress these artifacts, causing estimation errors to accumulate across time steps [14], [15]. High-frequency IMU jitter [16] and low-frequency bias drift [17] are two canonical examples that scalar Q/R tuning cannot resolve. EKF and UKF variants [18] relax the linearity requirement but inherit the same spectral blind spot, as their gain computation remains agnostic to the frequency content of the innovation.

In this work, we introduce the Frequency-Weighted Neural Kalman Filter (FW-NKF), a frequency-aware extension of Kalman filtering designed for state estimation. Instead of relying on a fixed linear observation matrix, we use a learnable neural network that maps latent states to denoised sensor measurements, supervised by ground truth states. This network implicitly learns to reconstruct noise-free sensor signals to enable a more accurate integration of the true state trajectory. To introduce frequency-specific error, we apply a Fourier transform to the estimated noise-free and observed signals and minimize the mean squared error in the frequency domain to encourage the filter to suppress spectral components most responsible for drift and noise. In addition, we introduce a learnable frequency-domain filter applied to the Kalman innovation term to selectively attenuate disruptive frequency bands during the correction step. By integrating frequency awareness into both the observation model and update mechanism, FW-NKF achieves improved state estimation accuracy across diverse tasks while maintaining the computational efficiency compared to the non-linear filters. Our contributions can be summarized:

- A hybrid filtering framework that embeds a causal, frequency-weighted operator within a Kalman filter to learn and attenuate system-specific noise concentrated in particular frequency bands.
- A spectral-domain training objective that minimizes a frequency-weighted error between predicted and denoised sensor signals to suppress noise-dominated frequency components. This is motivated by the Cramér–Rao lower bound, since reducing effective measurement noise can lower the minimum achievable estimator variance under standard regularity assumptions.
- Extensive evaluation on four heterogeneous benchmarks—pendulum, Lorenz chaotic systems, a micro aerial vehicle (MAV) and human pose estimation—achieving up to 10% reduction in state estimation while maintaining a competitive computational overhead.

* Corresponding author. All authors are with the Department of Computer Science, ETH Zürich, Switzerland¹. Code is available at <https://github.com/eth-siplab/frequency-weighted-neural-kalman-filters>

II. RELATED WORK

a) Classical and Adaptive Kalman Filtering: The Kalman filter [19] remains the standard recursive estimator for linear Gaussian systems owing to its closed-form predict–update cycle and minimum-variance guarantees [20], [21]. When dynamics are nonlinear, the Extended Kalman Filter (EKF) [22] linearizes the current estimate, while the Unscented Kalman Filter (UKF) [18] propagates deterministic sigma points to capture second-order statistics [23], [24]. Both families, along with innovation-based covariance adaptation and Interacting Multiple Model (IMM) switching [25], [26], adjust gains to accommodate maneuvering targets or drifting noise statistics. Nevertheless, all these variants treat measurement noise as spectrally flat, lacking a mechanism to selectively attenuate frequency-localized disturbances—the gap our work addresses. KalmanNet [27] replaces the analytic Kalman gain with a learned GRU mapping, achieving competitive accuracy without explicit noise statistics. Subsequent extensions add Bayesian weight posteriors for uncertainty calibration [28] or recursive gain structures for long-horizon stability [29]. Variational state-space models [30], [31] further generalize by jointly learning nonlinear transitions and emissions, and particle-filter hybrids extend this to non-Gaussian posteriors [21]. While these methods relax linearity and Gaussianity constraints, none incorporate spectral supervision or frequency-selective innovation filtering.

b) Frequency-Weighted Filtering: Frequency-shaped Kalman filters enhance robustness by selectively emphasizing informative spectral bands. By applying a causal frequency-shaping operator to the innovation sequence, these methods minimize weighted estimation variances in targeted frequency ranges, effectively suppressing band-limited noise [32]. Designing suitable shaping functions requires prior knowledge of sensor noise or estimation of noise characteristics.

c) Application-Specific Filtering and Localization: Many real-world estimation problems, ranging from autonomous navigation to human motion tracking, rely on noisy sensory data prone to frequency-dependent disturbances [33]. Benchmarks UIP-DP pose tracking collections demonstrate these challenges across diverse platforms, environments, and sensor modalities [34]. While application-specific methods and task-oriented neural networks have been developed to address domain-specific dynamics [35], they often overlook the structured nature of sensor noise, particularly its spectral characteristics. Our frequency-aware filtering approach can complement these existing methods by explicitly attenuating noise-dominated frequency components, improving state estimation accuracy without handcrafted filters.

III. PRELIMINARIES

A. Notations

We use lowercase letters (x) to denote scalar quantities, and bold lowercase letters (\mathbf{x}) to represent vectors, such as temporal sequences, while bold uppercase letters (\mathbf{X}) are used for matrices. The discrete Fourier transform is denoted as $\mathcal{F}(\cdot)$, yielding a complex variable as \mathcal{F}_x .

B. Linear Gaussian State-Space Models

Our framework builds on the discrete-time linear Gaussian state-space model [19], [20]:

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k), \quad (1)$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k), \quad (2)$$

with state $\mathbf{x}_k \in \mathbb{R}^n$, control $\mathbf{u}_k \in \mathbb{R}^p$, and observation $\mathbf{y}_k \in \mathbb{R}^m$. $\mathbf{F}_k \in \mathbb{R}^{n \times n}$ and $\mathbf{B}_k \in \mathbb{R}^{n \times p}$ govern state evolution, $\mathbf{Q}_k \in \mathbb{R}^{n \times n}$ models process uncertainty, $\mathbf{H}_k \in \mathbb{R}^{m \times n}$ projects states onto measurements, and $\mathbf{R}_k \in \mathbb{R}^{m \times m}$ encodes sensor noise [21].

C. Classical Kalman Filter

Given measurements $\mathbf{y}_{1:t}$, the Kalman filter recursively computes the posterior \mathbf{x}_t that minimizes the mean-squared estimation error under the model above [19], [20]:

a) Prediction Step:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F} \hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B} \mathbf{u}_t, \quad (3)$$

$$\mathbf{P}_{t|t-1} = \mathbf{F} \mathbf{P}_{t-1|t-1} \mathbf{F}^\top + \mathbf{Q}. \quad (4)$$

b) Update Step:

$$\Delta \mathbf{y}_t = \mathbf{y}_t - \mathbf{H} \hat{\mathbf{x}}_{t|t-1}, \quad (5)$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}^\top (\mathbf{H} \mathbf{P}_{t|t-1} \mathbf{H}^\top + \mathbf{R})^{-1}, \quad (6)$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \Delta \mathbf{y}_t, \quad \mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_{t|t-1}. \quad (7)$$

The subscript $t|t-1$ indicates the prior (before incorporating \mathbf{y}_t) and $t|t$ the posterior. Eqs. (3)–(7) alternate between propagating the state forward and correcting it via the innovation $\Delta \mathbf{y}_t$ scaled by the Kalman gain \mathbf{K}_t .

D. Motivation

Kalman filters assume stationary noise covariances \mathbf{Q} and \mathbf{R} , which are suboptimal when sensor noise exhibits frequency-dependent characteristics [32]. In practice, sensor measurements contain structured disturbances such as low-frequency drift, high-frequency jitter, or periodic environmental noise, none of which are captured by the white noise assumption [19]. These spectral artifacts can lead to suboptimal state estimation, as the Kalman gain cannot distinguish between informative signal components and frequency-localized noise. We propose frequency-weighted neural Kalman filter to introduce spectral awareness into both the observation model and the update step. By combining an observation network trained with spectral-domain supervision and a causal filter applied to the innovation term, FW-NKF selectively suppresses noise-dominated frequencies while preserving task-relevant information. This approach enables accurate state estimation in environments where classical filters fall short, particularly for sensor modalities like IMUs, where frequency-specific disturbances are common [12].

IV. METHOD

In this section, we present two key enhancements to the Kalman filter, which are a frequency-domain reconstruction loss for training (Algorithm 1) and a learnable IIR innovation filtering mechanism (Algorithm 2). These components address

limitations in conventional Kalman filter implementations when dealing with systems that have distinct frequency characteristics with non-Gaussian noise.

A. Incorporating Frequency Loss into the Kalman Filter

The frequency-domain reconstruction loss introduces a critical improvement over traditional time-domain losses by comparing signals in the spectral domain rather than point-by-point in time. Specifically, we minimize the difference between the magnitude spectra of predicted and ground-truth observations, obtained via the Fourier Transform. This shifts the focus from waveform alignment to frequency content, which is often more relevant in systems affected by structured, non-Gaussian noise—such as IMU signals [33], [12]. We define the spectral loss during training as in Equation 8.

$$\mathcal{L}_\Phi = \left\| \|\mathcal{F}(\hat{\mathbf{y}}_t)\| - \|\mathcal{F}(\tilde{\mathbf{y}}_t)\| \right\|_1 \quad (8)$$

where $\hat{\mathbf{y}}_t = H(\hat{\mathbf{x}}_{t|t-1})$ is the predicted observation from the estimated state, and $\tilde{\mathbf{y}}_t = H(\mathbf{x}_t)$ is the reference noiseless observation derived from the ground truth. Although noiseless readings are not directly available, we approximate them by applying the observation model to the ground truth states during training. Under the supervision, the observation model $H(\cdot)$ is trained to approximate noise-free observations, making $\tilde{\mathbf{y}}_t$ a valid target for guiding spectral reconstruction.

Unlike time-domain minimum square error, this loss is phase-invariant and robust to temporal misalignments. The use of the ℓ_1 norm further encourages sparsity in spectral differences, ensuring the model focuses on aligning dominant frequency components while ignoring minor discrepancies. Algorithm 1 summarizes the implementation of incorporating the frequency loss to the original filter.

Algorithm 1 Integration of spectral loss into the Kalman filter training

- 1: **procedure** SPECTRALLOSS($\{\mathbf{x}_{t|t}\}_{t=1}^T$: filtered states, $\{\hat{\mathbf{y}}_t\}_{t=1}^T$: predicted observations, $\{\mathbf{x}_t\}_{t=1}^T$: true states)
 - 2: $\mathcal{L}_{\text{state}} \leftarrow \frac{1}{T} \sum_{t=1}^T \|\hat{\mathbf{x}}_{t|t} - \mathbf{x}_t\|_2^2$ \triangleright State estimation loss
 - 3: $\tilde{\mathbf{y}}_t \leftarrow H(\mathbf{x}_t)$ \triangleright Estimated noiseless observation from ground truth state
 - 4: $\mathcal{F}_{\tilde{\mathbf{y}}_t} \leftarrow \|\mathcal{F}\{\tilde{\mathbf{y}}_t\}\|$ \triangleright Magnitude spectrum of estimated noiseless observation
 - 5: $\mathcal{F}_{\hat{\mathbf{y}}_t} \leftarrow \|\mathcal{F}\{\hat{\mathbf{y}}_t\}\|$ \triangleright Magnitude spectrum of predicted observation
 - 6: $\mathcal{L}_\Phi \leftarrow \frac{1}{\omega} \sum_{\omega=0}^{\pi/2} \|\mathcal{F}_{\hat{\mathbf{y}}_t} - \mathcal{F}_{\tilde{\mathbf{y}}_t}\|_1$ \triangleright Spectral-domain reconstruction loss
 - 7: $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{state}} + \lambda_\Phi \cdot \mathcal{L}_\Phi$
 - 8: **return** $\mathcal{L}_{\text{total}}$
 - 9: **end procedure**
-

The proposed frequency-domain loss extends traditional time-domain objectives by comparing the magnitude spectra of predicted and noiseless observations via the Fourier transform. This emphasizes spectral structure rather than point-by-point waveform differences. It is especially useful

when frequency matters more than exact time alignment, such as in IMUs with vibration and interference, and in settings where noise is approximately non-Gaussian [12], [33].

B. Learnable IIR Innovation Filtering for Kalman Estimation

We apply a learnable causal IIR filter directly to the innovation $\Delta \mathbf{y}_t$ defined in (5), where $\Delta \mathbf{y}_t = \mathbf{y}_t - \mathbf{H} \hat{\mathbf{x}}_{t|t-1}$. Unless stated otherwise, the same filter is shared across measurement channels and acts componentwise.

a) *Z-transform convention and delay.*: For a discrete vector sequence $\mathbf{u}_t \in \mathbb{R}^m$, its one-sided Z-transform is $\mathbf{U}(z) = \sum_{t=0}^{\infty} \mathbf{u}_t z^{-t}$, with complex indeterminate z . Here z^{-1} denotes one-sample delay; more generally z^{-i} is an i -step delay. When the same scalar IIR is applied to each channel, this is equivalent to a diagonal transfer $\Phi(z) = \Phi(z) \mathbf{I}_m$.

b) *Filter parameterization.*: We parameterize a causal IIR with learnable coefficients $\{b_i\}_{i=0}^p$ and $\{a_j\}_{j=1}^q$:

$$\Phi(z) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{\sum_{i=0}^p b_i z^{-i}}{1 + \sum_{j=1}^q a_j z^{-j}}. \quad (9)$$

c) *Time-domain realization and KF correction.*: Instead of the single-line ARMA recursion, we use the equivalent cascaded realization with an internal buffer $\mathbf{s}_t \in \mathbb{R}^m$:

$$\mathbf{s}_t = - \sum_{j=1}^q a_j \mathbf{s}_{t-j} + \Delta \mathbf{y}_t, \quad (10)$$

$$\widetilde{\Delta \mathbf{y}}_t = \sum_{i=0}^p b_i \mathbf{s}_{t-i}. \quad (11)$$

The filtered innovation $\widetilde{\Delta \mathbf{y}}_t = \Phi * \Delta \mathbf{y}_t$ replaces the raw innovation in the Kalman update as in Equation 7.

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \widetilde{\Delta \mathbf{y}}_t, \quad (12)$$

Stability is enforced by constraining the poles of $A(z^{-1})$ to lie strictly inside the unit circle (e.g., via a stable-root reparameterization of $\{a_j\}$); this preserves a well-posed recursion and, with the Joseph form, maintains positive semidefiniteness of $\mathbf{P}_{t|t}$ under finite precision.

C. End-to-End Integration in FW-NKF

The two components enter the forward pass at complementary points without altering the Kalman structure. During *training*, the observation pathway is supervised in the spectral domain via Algorithm 1, which compares $\hat{\mathbf{y}}_t = H(\hat{\mathbf{x}}_{t|t-1})$ to the noiseless target $\tilde{\mathbf{y}}_t = H(\mathbf{x}_t)$ through the loss in (8). During *inference and training*, the correction step replaces the raw innovation $\Delta \mathbf{y}_t$ from (5) with the frequency-weighted innovation produced by the learnable IIR in (10)–(11), equivalently $\widetilde{\Delta \mathbf{y}}_t = [\Phi * \Delta \mathbf{y}]_t$. The time-varying dynamics and noise terms $(\mathbf{A}_t, \mathbf{b}_t, \mathbf{Q}_t, \mathbf{R}_t)$ are generated by a GRU-conditioned parameterization, while the IIR coefficients $\{a_j\}, \{b_i\}$ are learned with the stability constraint on $A(z^{-1})$ as specified in Section IV-B. When measurements are missing or unreliable, the optional mask \mathbf{m}_t gates the update.

Proposition IV.1 formalizes a theoretical justification for our method. It states that spectral filtering can improve state

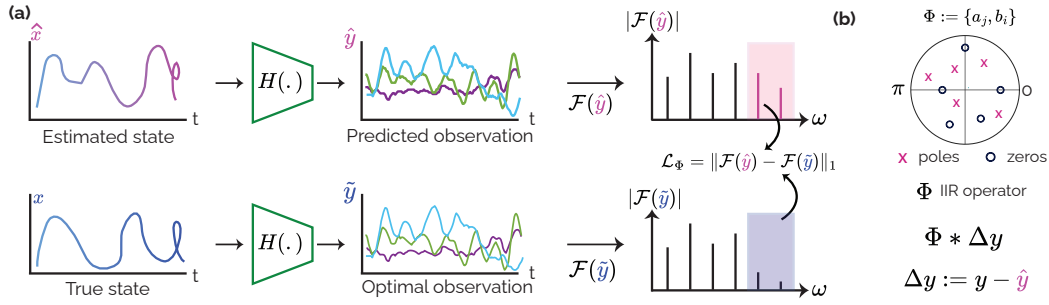


Fig. 1: Overview of our method with spectral supervision. **(a)** The observation model $H(\cdot)$ predicts sensor measurements in two ways: one from the ground truth state x (supervised) and one from the estimated state \hat{x} . This dual prediction enables to explicitly guide the observation model in reconstructing noise-free sensor signals, encouraging frequency-selective denoising. **(b)** The pole-zero plot of the learnable IIR filter Φ , parameterized by coefficients $\{a_j, b_j\}$, which are optimized during training. This filter is applied causally to the innovation term (the difference between observed sensor data y and predicted observation \hat{y}), selectively attenuating frequency components associated with measurement noise before the Kalman update.

Algorithm 2 Adaptive Frequency-Weighted Innovation

- 1: **procedure** IIR-INNOVATION($y_t, \hat{x}_{t|t-1}$, coeffs. $\{b_i\}_{i=0}^p, \{a_j\}_{j=1}^q$)
 - 2: **Inputs:** observation $y_t \in \mathbb{R}^m$; predicted state $\hat{x}_{t|t-1} \in \mathbb{R}^n$; observation matrix $H \in \mathbb{R}^{m \times n}$.
 - 3: **Coeffs:** feedforward $\{b_i\}_{i=0}^p$, feedback $\{a_j\}_{j=1}^q$.
 - 4: **Persistent internal state:** lagged IIR states $\{s_{t-j}\}_{j=1}^q$ (initialized to $\mathbf{0}$ at start).
 - 5: $\Delta y_t \leftarrow y_t - H \hat{x}_{t|t-1}$ \triangleright Innovation from (5)
 - 6: $s_t \leftarrow -\sum_{j=1}^q a_j s_{t-j} + \Delta y_t$ \triangleright Pole recursion, (10)
 - 7: $\widetilde{\Delta y}_t \leftarrow \sum_{i=0}^p b_i s_{t-i}$ \triangleright FIR stage (11)
 - 8: **State update:** shift and store s_t as the newest lag (discard the oldest).
 - 9: **Output:** $\widetilde{\Delta y}_t$ \triangleright Use in KF: $\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \widetilde{\Delta y}_t$
 - 10: **Cost:** $O((p+q)m)$ per step for componentwise filtering with a shared Φ .
 - 11: **end procedure**
-

estimation when it suppresses noise in frequency bands that are weakly informative about the state, without removing state-dependent content.

In the measurement model $y = h(x) + v$, the noise v is often structured and frequency-dependent. Applying a (causal) linear filter Φ yields the transformed observation $\tilde{y} = \Phi * y$, which both reshapes the measurement model and filters the noise. When this transformation increases the information content of the observation about the state, the achievable estimation error decreases by the Cramér–Rao bound.

This result supports our approach of learning Φ . By training Φ to attenuate noise-dominated frequency components while preserving state-informative components, we improve the signal-to-noise ratio of the measurements, which is reflected in better state estimation in our experiments.

Proposition IV.1 (Spectral Filtering). *Let $y = h(x) + v$, where $x \in \mathbb{R}^n$, h is smooth, and v is zero-mean noise with covariance $R = \mathbb{E}[vv^\top]$. Let Φ be any (causal) linear filter and $\tilde{y} = \Phi y = \Phi h(x) + \tilde{v}$ with $\tilde{v} = \Phi v$ and $\tilde{R} = \Phi R \Phi^\top$.*

Algorithm 3 FW-NKF Prediction & Update (forward step)

- 1: **procedure** FORWARD STEP(y_t : observation, m_t : mask (optional), u_t : control (optional))
 - 2: *Given cached \hat{x}_{t-1}, P_{t-1} , encoder hidden h_{t-1} , IIR lags $\{s_{t-j}\}_{j=1}^q$, fixed H*
 - 3: $z_t \leftarrow [\hat{x}_{t-1}, y_t]$ \triangleright Append u_t if present
 - 4: $(h_t) \leftarrow \text{GRU}(z_t, h_{t-1})$
 - 5: $A_t \leftarrow I + \Delta A_t(h_t)$ \triangleright Low-rank or diagonal gate
 - 6: $b_t \leftarrow f_b(h_t)$; $Q_t \leftarrow f_Q(h_t)$; $R_t \leftarrow f_R(h_t)$
 - 7: $\hat{x}_{t|t-1} \leftarrow A_t \hat{x}_{t-1} + b_t$
 - 8: $P_{t|t-1} \leftarrow A_t P_{t-1} A_t^\top + Q_t$
 - 9: $S_t \leftarrow H P_{t|t-1} H^\top + R_t$
 - 10: $K_t \leftarrow P_{t|t-1} H^\top S_t^{-1}$ \triangleright Cholesky-inverse
 - 11: $\Delta y_t \leftarrow y_t - H \hat{x}_{t|t-1}$ \triangleright Innovation
 - 12: $s_t \leftarrow -\sum_{j=1}^q a_j s_{t-j} + \Delta y_t$ \triangleright IIR all-pole stage
 - 13: $\widetilde{\Delta y}_t \leftarrow \sum_{i=0}^p b_i s_{t-i}$ \triangleright IIR FIR stage
 - 14: $\hat{x}'_{t|t} \leftarrow \hat{x}_{t|t-1} + K_t \widetilde{\Delta y}_t$
 - 15: $P'_{t|t} \leftarrow (I - K_t H) P_{t|t-1} (I - K_t H)^\top + K_t R_t K_t^\top$ \triangleright Joseph form
 - 16: $\hat{x}_{t|t} \leftarrow m_t \hat{x}'_{t|t} + (1 - m_t) \hat{x}_{t|t-1}$ \triangleright Mask update
 - 17: $P_{t|t} \leftarrow m_t P'_{t|t} + (1 - m_t) P_{t|t-1}$
 - 18: update lags $\{s_{t-j}\} \leftarrow$ shift-in s_t and discard oldest \triangleright Roll IIR state
 - 19: cache $\hat{x}_t \leftarrow \hat{x}_{t|t}, P_t \leftarrow P_{t|t}, h_t$ \triangleright For the next step
 - 20: **return** $\hat{x}_{t|t}, P_{t|t}, h_t, s_t$
 - 21: **end procedure**
-

When the Fisher information in the filtered observation is no smaller than in the original observation, i.e., $I_{\tilde{y}}(x) \succeq I_y(x)$. Then, by the Cramér–Rao bound, any unbiased estimator based on \tilde{y} has covariance no larger than one based on y .

Proof: The Cramér–Rao bound states that for any unbiased estimator \hat{x} based on an observation z , $\text{Cov}(\hat{x}) \succeq I_z(x)^{-1}$ (under standard regularity conditions). If $I_{\tilde{y}}(x) \succeq I_y(x)$, then $I_{\tilde{y}}(x)^{-1} \preceq I_y(x)^{-1}$, which yields the stated ordering on the covariance lower bounds. \square

V. EXPERIMENTS

We conducted experiments on four datasets spanning distinct applications to demonstrate the versatility of our method. Below, we provide a brief overview of each dataset.

A. Datasets

1) Chaotic systems:

a) *Lorenz*: We simulate the Lorenz–63 system

$$\nabla \mathbf{x} = \sigma(\mathbf{y} - \mathbf{x}), \quad \nabla \mathbf{y} = \mathbf{x}(\rho - z) - \mathbf{y}, \quad \nabla z = \mathbf{x}\mathbf{y} - \beta z$$

with the standard parameters $\sigma=10$, $\rho=28$, $\beta=8/3$. Each trajectory has length $T=100$ steps with $\Delta t=0.01$ s (total 1 s). Initial states are drawn i.i.d. from $\mathcal{N}(0, 5^2 \mathbf{I}_3)$. We integrate the continuous dynamics using `odeint` and form latent states $\mathbf{x}_t \in \mathbb{R}^3$. To stress the filter, we add small process perturbations to the *simulated* path before sensing:

$$\mathbf{x}_t^{\text{proc}} = \mathbf{x}_t + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, 0.1^2 \mathbf{I}_3). \quad (13)$$

Observations are either partial or full. (i) **Partial**: $\mathbf{y}_t = [\mathbf{x}_{t,1}^{\text{proc}}, \mathbf{x}_{t,2}^{\text{proc}}]^\top + \boldsymbol{\eta}_t \in \mathbb{R}^2$; (ii) **full**: $\mathbf{y}_t = \mathbf{x}_t^{\text{proc}} + \boldsymbol{\eta}_t \in \mathbb{R}^3$, with $\boldsymbol{\eta}_t \sim \mathcal{N}(0, 0.5^2 \mathbf{I})$. We report results for the partial setting (harder). For linear-filter baselines, we use fixed discrete approximation of F and H (linearized around an equilibrium) as in Equation 14.

$$\mathbf{F} = \mathbf{I}_3 + \Delta t \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{bmatrix}, \quad (14a)$$

$$\mathbf{H}_{\text{partial}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{H}_{\text{full}} = \mathbf{I}_3. \quad (14b)$$

We generate $N=1000$ trajectories and use an 80/10/10 split (train/val/test) with fixed seeds for reproducibility.

b) *Pendulum*.: We consider a damped, forced nonlinear pendulum with state $\mathbf{x}_t = [\theta_t, \omega_t]^\top$ and dynamics

$$\nabla \theta = \omega, \quad \nabla \omega = -(g/L) \sin \theta - (b/m)\omega + u_t/m, \quad (15)$$

with $g=9.81$, $L=1.0$, $b=0.1$, $m=1.0$. We discretize via forward Euler at $\Delta t=0.02$ s for $T=100$ steps (2 s). Controls are zero-mean Gaussian $u_t \sim \mathcal{N}(0, 0.5^2)$. Process noise is injected additively after the Euler step:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta t f(\mathbf{x}_t, u_t) + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, 0.05^2 \mathbf{I}_2); \quad (16)$$

angles are wrapped to $(-\pi, \pi]$. Observations are Cartesian tip positions with Gaussian noise:

$$\mathbf{y}_t = \begin{bmatrix} L \sin \theta_t \\ -L \cos \theta_t \end{bmatrix} + \boldsymbol{\eta}_t, \quad \boldsymbol{\eta}_t \sim \mathcal{N}(0, 0.1^2 \mathbf{I}_2). \quad (17)$$

For methods [27], [28] that require analytic solutions, we use the approximate analytic Jacobians implemented as,

$$\mathbf{F} = \begin{bmatrix} 1 - \sigma \Delta t & \sigma \Delta t & 0 \\ \rho \Delta t & 1 - \Delta t & 0 \\ 0 & 0 & 1 - \beta \Delta t \end{bmatrix} \quad (18)$$

$$\mathbf{H}_{\text{partial}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{H}_{\text{full}} = \mathbf{I}_3. \quad (19)$$

with $\bar{\theta}$ the linearization point. We generate $N=30000$ trajectories and use the same 80/10/10 split.

Usage in our Kalman models. In both datasets the training target is the clean latent state \mathbf{x}_t (pre-noise Lorenz states and post-process-noise pendulum states as returned by the generator). Measurement models and linearization matrices are provided to classical filters; learned Kalman variants receive the same observations and, where specified, learn adaptive \mathbf{Q}_t , \mathbf{R}_t (and, for ablations, frequency-aware variants). All hyperparameters above exactly match the released defaults.

2) Real-world:

a) *UIP-DB*: UIP-DB contains 200 minutes of synchronized IMU and UWB range data from six wearable sensors alongside optical motion-capture ground truth across 25 motion types performed by 10 participants, designed for full-body inertial pose estimation with ranging constraints [36].

The UIP-DB provides a challenging and realistic benchmark for Kalman filtering. Full-body pose estimation from wearable IMUs is subject to drift due to sensor bias, while UWB ranging measurements are corrupted by high-frequency multipath effects. Kalman filters are a natural fit for this setting, as they fuse heterogeneous measurements with noise statistics into a coherent state estimate.

Importantly, the complementary characteristics of IMU and UWB measurements motivate frequency-aware modeling. Low-frequency content often reflects slow pose evolution, while higher-frequency content captures rapid motion and noise. A frequency-weighted Kalman approach adjusts the filter’s trust across bands, down-weighting noise-dominated components while preserving fast transitions.

b) *EuRoC MAV*: The EuRoC MAV dataset [37] provides synchronized IMU measurements at 200 Hz and motion-capture ground truth from a hexacopter flying in indoor environments. We use 10 sequences (MH_02–05, V1_01–03, V2_01–03) and formulate an IMU-based state estimation task where the observation is the 6D IMU (3-axis gyroscope + 3-axis accelerometer), $\mathbf{y}_t \in \mathbb{R}^6$, and the state comprises position, velocity, and quaternion orientation, $\mathbf{x}_t \in \mathbb{R}^{10}$. We segment each sequence into overlapping windows of length $T=200$ with stride 50, yielding 4,266 samples, and apply an 80/10/10 train/val/test split. For all Kalman filter variants, the observation model is a learned linear mapping $\mathbf{H} \in \mathbb{R}^{6 \times 10}$.

B. Evaluation Metrics

We evaluate our method across diverse tasks, focusing on state estimation accuracy. For all tasks, we report the state estimation error in ℓ_1 and ℓ_2 term. The primary metric is the root-mean-square error (RMSE) between the estimated $\hat{\mathbf{x}}_t$ and the ground truth state \mathbf{x}_t , computed as in Equation 20.

$$\begin{aligned} \text{MSE} &= \frac{1}{T} \sum_t \|\mathbf{e}_t\|_2^2, & \text{RMSE} &= \sqrt{\text{MSE}}, \\ \text{MAE} &= \frac{1}{T} \sum_t \|\mathbf{e}_t\|_1, & \text{NRMSE} &= \frac{\text{RMSE}}{\sqrt{\frac{1}{T} \sum_t \|\mathbf{x}_t\|_2^2}}, \end{aligned} \quad (20)$$

where $\mathbf{e}_t \triangleq \hat{\mathbf{x}}_t - \mathbf{x}_t$.

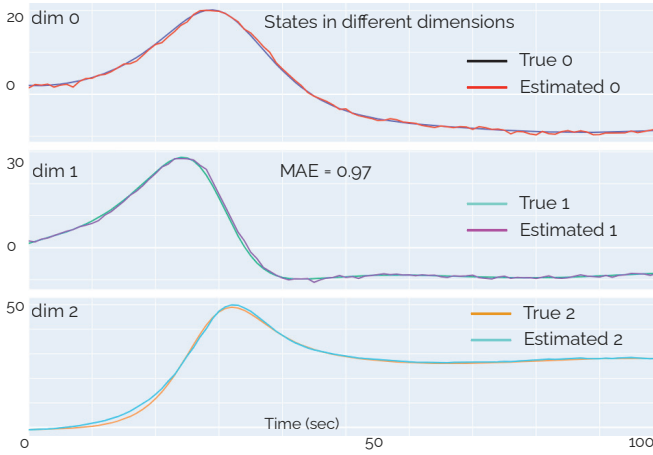


Fig. 2: The state estimation performance of our method illustrated in 3-dimensional (x, y, z) Lorenz system where the estimations follow the true state in all dimensions with low mean absolute error. All y -axes are dimensionless.

TABLE I: Performance comparison of different Kalman filtering methods on Pendulum dataset with varying weights.

| Method | λ_Φ | Performance Metrics | | | | |
|------------------------------|----------------|---------------------|------------------|-------------------|--------------------|---------------------------|
| | | MAE \downarrow | MSE \downarrow | RMSE \downarrow | NRMSE \downarrow | R ² \uparrow |
| <i>Deep Learning Methods</i> | | | | | | |
| FW-NKF (ours) | 0.0 | 0.75±0.08 | 1.24±0.28 | 1.10±0.12 | 0.51±0.05 | 0.74±0.05 |
| FW-NKF (ours) | 0.01 | 0.70±0.07 | 1.10±0.23 | 1.04±0.10 | 0.48±0.05 | 0.77±0.04 |
| FW-NKF (ours) | 0.1 | 0.99±0.14 | 2.03±0.63 | 1.42±0.21 | 0.65±0.09 | 0.59±0.12 |
| BayesKNet [28] | 0.0 | 1.19±0.06 | 2.93±0.32 | 1.71±0.09 | 0.79±0.04 | 0.40±0.06 |
| BayesKNet [28] | 0.01 | 1.37±0.07 | 3.88±0.40 | 1.97±0.10 | 0.91±0.04 | 0.21±0.08 |
| BayesKNet [28] | 0.1 | 1.38±0.07 | 3.96±0.41 | 1.99±0.10 | 0.92±0.04 | 0.20±0.08 |
| Recursive KNet [29] | 0.0 | 1.03±0.08 | 2.22±0.38 | 1.48±0.12 | 0.68±0.05 | 0.55±0.07 |
| Recursive KNet [29] | 0.01 | 0.98±0.08 | 2.00±0.35 | 1.41±0.12 | 0.65±0.05 | 0.59±0.07 |
| Recursive KNet [29] | 0.1 | 1.10±0.10 | 2.52±0.48 | 1.58±0.15 | 0.73±0.06 | 0.49±0.09 |
| <i>Neural Kalman Methods</i> | | | | | | |
| Kalman Net [27] | 0.0 | 1.22±0.06 | 3.08±0.34 | 1.75±0.09 | 0.81±0.04 | 0.37±0.06 |
| Kalman Net [27] | 0.01 | 1.26±0.07 | 3.33±0.39 | 1.82±0.10 | 0.84±0.05 | 0.32±0.07 |
| Kalman Net [27] | 0.1 | 1.30±0.07 | 3.50±0.42 | 1.87±0.11 | 0.86±0.05 | 0.29±0.08 |
| Recurrent KNet [38] | 0.0 | 1.97±0.12 | 8.04±1.05 | 2.83±0.18 | 1.31±0.08 | -0.62±0.21 |
| Recurrent KNet [38] | 0.01 | 2.03±0.13 | 8.54±1.12 | 2.92±0.19 | 1.35±0.08 | -0.72±0.22 |
| Recurrent KNet [38] | 0.1 | 2.10±0.14 | 9.15±1.21 | 3.02±0.20 | 1.40±0.09 | -0.84±0.24 |
| <i>Classical Methods</i> | | | | | | |
| Classical KF [19] | 0.0 | 1.67±0.10 | 5.80±0.73 | 2.40±0.15 | 1.11±0.07 | -0.16±0.14 |
| Classical KF [19] | 0.01 | 1.86±0.11 | 7.20±0.92 | 2.68±0.17 | 1.24±0.08 | -0.45±0.18 |
| Classical KF [19] | 0.1 | 1.87±0.11 | 7.28±0.93 | 2.70±0.17 | 1.25±0.08 | -0.46±0.18 |
| Autoreg KF [39] | 0.0 | 1.78±0.09 | 5.28±0.46 | 2.29±0.10 | 1.00±0.00 | 0.01±0.01 |
| Autoreg KF [39] | 0.01 | 1.87±0.10 | 5.81±0.51 | 2.41±0.10 | 1.05±0.01 | -0.17±0.02 |
| Autoreg KF [39] | 0.1 | 1.96±0.10 | 6.40±0.56 | 2.53±0.11 | 1.10±0.01 | -0.29±0.02 |
| Gain (%) | | 28.57% | 45.00% | 26.24% | 26.15% | 30.51% |

In addition to per-frame errors, we also report the Average Trajectory Error (ATE), which measures the average Euclidean distance between the estimated trajectory and the ground truth trajectory over time. To capture worst-case deviations, we further report the maximum ATE, defined as the largest pointwise trajectory error observed during the sequence.

All methods are evaluated using the same ground truth references and synchronized time frames. Standard deviations across random seeds are omitted from the tables when they fall below a significance threshold, indicating negligible variance.

C. Baselines

Our framework is designed to be modular and can be integrated with a wide range of Kalman filtering approaches.

TABLE II: Performance comparison of different Kalman filtering methods on Lorenz dataset with varying weights.

| Method | λ_Φ | Performance Metrics | | | | |
|------------------------------|----------------|---------------------|------------------|-------------------|--------------------|---------------------------|
| | | MAE \downarrow | MSE \downarrow | RMSE \downarrow | NRMSE \downarrow | R ² \uparrow |
| <i>Deep Learning Methods</i> | | | | | | |
| FW-NKF (ours) | 0.0 | 0.62±0.10 | 0.81±0.26 | 0.87±0.14 | 0.05±0.09 | 0.97±0.01 |
| FW-NKF (ours) | 0.01 | 0.50±0.15 | 0.58±0.36 | 0.71±0.22 | 0.04±0.01 | 0.97±0.01 |
| FW-NKF (ours) | 0.1 | 8.18±1.04 | 119.82±30.51 | 10.94±1.38 | 0.66±0.08 | 0.54±0.11 |
| BayesKNet [28] | 0.0 | 4.49±1.03 | 35.84±16.56 | 5.94±1.36 | 0.36±0.08 | 0.86±0.06 |
| BayesKNet [28] | 0.01 | 7.51±1.33 | 103.09±36.46 | 10.11±1.76 | 0.61±0.10 | 0.61±0.13 |
| BayesKNet [28] | 0.1 | 7.32±1.02 | 98.20±27.96 | 9.87±1.38 | 0.60±0.08 | 0.63±0.10 |
| Recursive KNet [29] | 0.0 | 5.17±0.34 | 49.49±6.65 | 7.02±0.47 | 0.42±0.02 | 0.81±0.02 |
| Recursive KNet [29] | 0.01 | 4.24±0.24 | 33.33±3.77 | 5.76±0.32 | 0.35±0.02 | 0.87±0.01 |
| Recursive KNet [29] | 0.1 | 4.85±0.18 | 43.03±3.36 | 6.55±0.25 | 0.40±0.01 | 0.83±0.01 |
| <i>Neural Kalman Methods</i> | | | | | | |
| Kalman Net [27] | 0.0 | 5.14±0.24 | 48.70±4.64 | 6.97±0.33 | 0.42±0.02 | 0.81±0.02 |
| Kalman Net [27] | 0.01 | 6.11±0.23 | 68.67±5.16 | 8.28±0.31 | 0.50±0.01 | 0.74±0.02 |
| Kalman Net [27] | 0.1 | 6.65±0.24 | 81.33±6.04 | 9.01±0.33 | 0.55±0.02 | 0.69±0.02 |
| Recurrent KNet [38] | 0.0 | 8.61±0.43 | 136.08±13.55 | 11.66±0.58 | 0.71±0.03 | 0.48±0.05 |
| Recurrent KNet [38] | 0.01 | 9.60±0.49 | 169.32±17.32 | 13.01±0.66 | 0.79±0.04 | 0.36±0.06 |
| Recurrent KNet [38] | 0.1 | 13.58±0.60 | 338.60±29.87 | 18.40±0.81 | 1.12±0.05 | -0.27±0.11 |
| <i>Classical Methods</i> | | | | | | |
| Classical KF [19] | 0.0 | 8.43±0.41 | 130.61±12.87 | 11.43±0.56 | 0.69±0.03 | 0.50±0.05 |
| Classical KF [19] | 0.01 | 8.77±0.42 | 141.47±13.70 | 11.89±0.57 | 0.72±0.03 | 0.46±0.05 |
| Classical KF [19] | 0.1 | 10.11±0.50 | 187.82±18.83 | 13.70±0.68 | 0.83±0.04 | 0.29±0.07 |
| Autoreg KF [39] | 0.0 | 13.20±1.49 | 311.83±27.20 | 17.64±0.77 | 1.07±0.05 | -0.16±0.10 |
| Autoreg KF [39] | 0.01 | 13.23±1.50 | 313.06±27.33 | 17.68±0.78 | 1.08±0.05 | -0.16±0.10 |
| Autoreg KF [39] | 0.1 | 13.29±1.51 | 315.52±27.60 | 17.76±0.78 | 1.08±0.05 | -0.17±0.10 |
| Gain (%) | | 88.21% | 98.26% | 87.67% | 88.57% | 11.49% |

To demonstrate its versatility, we also evaluate against six representative baselines that span classical, autoregressive, and neural Kalman filters.

BayesKNet [28] places variational posteriors over the weight matrices of a gain-predicting network, yielding both a state estimate and an epistemic uncertainty at each step.

Recursive KNet [29] feeds the previous Kalman gain back into the gain-prediction network at each time step to model gain dynamics and improving stability over long horizons.

KalmanNet [27] parameterize the Kalman gain as a GRU that ingests the innovation and prior covariance features, sidestepping the need for known Q and R .

Recurrent KNet [38] augments the gain network with an LSTM cell whose hidden state carries information from earlier time steps, capturing long-range temporal correlations.

Classical KF [19] uses the analytic predict–update equations with fixed F, H, Q, R matrices, serving as the MMSE-optimal reference under correct model assumptions.

Autoregressive KF [39] replaces the first-order Markov transition with a higher-order autoregressive process model, allowing richer temporal structure without neural components.

Application of spectral loss to baselines. The frequency-domain reconstruction loss \mathcal{L}_Φ (Eq. 8) is a *training objective*, not an architectural modification. It operates by projecting estimated and ground-truth states through the observation model H and comparing their magnitude spectra. This makes it applicable to any differentiable Kalman filter whose parameters are optimized via gradient descent, regardless of whether those parameters are neural network weights or classical system matrices.

Concretely, Classical KF and Autoregressive KF parameterize the transition matrix F , observation matrix H , and noise covariances Q, R as learnable tensors. When $\lambda_\Phi > 0$, the spectral loss provides an additional gradient signal that

TABLE III: Performance comparison of Kalman filtering methods for state estimation on UIP-DB using IMU and UWB range measurements.

| Method | λ_Φ | Acceleration | | Rotation | Inter Sensor Distance | |
|------------------------------|----------------|------------------|-------------------|------------------|-----------------------|------------------|
| | | RMSE ↓ | MAE ↓ | ρ ↓ | RMSE ↓ | MAE ↓ |
| <i>Deep Learning Methods</i> | | | | | | |
| FW-NKF (ours) | 0.0 | 4.37±0.24 | 13.46±0.83 | 0.42±0.01 | 1.41±0.07 | 6.80±0.34 |
| FW-NKF (ours) | 0.01 | 4.15±0.20 | 12.78±0.66 | 0.40±0.01 | 1.34±0.09 | 6.45±0.42 |
| FW-NKF (ours) | 0.1 | 4.73±0.33 | 14.57±1.15 | 0.45±0.02 | 1.53±0.11 | 7.35±0.57 |
| FW-NKF (ours) | 1.0 | 5.90±0.42 | 18.18±1.39 | 0.56±0.03 | 1.90±0.14 | 9.16±0.72 |
| Recursive KNet [29] | 0.0 | 5.46±0.39 | 16.82±1.26 | 0.52±0.03 | 1.76±0.13 | 8.48±0.66 |
| Recursive KNet [29] | 0.01 | 5.18±0.31 | 15.95±1.00 | 0.49±0.02 | 1.67±0.11 | 8.04±0.50 |
| Recursive KNet [29] | 0.1 | 5.90±0.42 | 18.18±1.39 | 0.56±0.03 | 1.90±0.14 | 9.16±0.72 |
| BayesKNet [28] | 0.0 | 6.28±0.45 | 19.33±1.45 | 0.60±0.04 | 2.02±0.15 | 9.75±0.76 |
| BayesKNet [28] | 0.01 | 5.95±0.36 | 18.33±1.15 | 0.57±0.03 | 1.92±0.12 | 9.24±0.58 |
| BayesKNet [28] | 0.1 | 6.78±0.49 | 20.89±1.56 | 0.65±0.04 | 2.19±0.16 | 10.53±0.83 |
| <i>Neural Kalman Methods</i> | | | | | | |
| Kalman Net [27] | 0.0 | 7.21±0.52 | 22.20±1.66 | 0.69±0.04 | 2.32±0.17 | 11.20±0.88 |
| Kalman Net [27] | 0.01 | 6.83±0.41 | 21.05±1.32 | 0.65±0.04 | 2.20±0.14 | 10.60±0.66 |
| Kalman Net [27] | 0.1 | 7.79±0.56 | 24.00±1.80 | 0.74±0.05 | 2.51±0.18 | 12.10±0.95 |
| Recurrent KNet [38] | 0.0 | 10.06±0.72 | 31.00±2.32 | 0.96±0.06 | 3.24±0.23 | 15.64±1.23 |
| Recurrent KNet [38] | 0.01 | 9.53±0.57 | 29.36±1.84 | 0.91±0.05 | 3.07±0.20 | 14.81±0.93 |
| Recurrent KNet [38] | 0.1 | 10.87±0.78 | 33.49±2.51 | 1.04±0.07 | 3.51±0.27 | 16.89±1.33 |
| <i>Classical Methods</i> | | | | | | |
| Classical KF [19] | 0.0 | 9.05±0.65 | 27.89±2.09 | 0.86±0.05 | 2.92±0.22 | 14.06±1.11 |
| Classical KF [19] | 0.01 | 8.58±0.52 | 26.43±1.66 | 0.82±0.05 | 2.77±0.18 | 13.31±0.83 |
| Classical KF [19] | 0.1 | 9.78±0.70 | 30.14±2.26 | 0.93±0.06 | 3.15±0.24 | 15.20±1.20 |
| Autoreg KF [39] | 0.0 | 10.66±0.77 | 32.85±2.46 | 1.02±0.06 | 3.44±0.26 | 16.57±1.31 |
| Autoreg KF [39] | 0.01 | 10.10±0.61 | 31.12±1.95 | 0.96±0.06 | 3.26±0.21 | 15.69±0.98 |
| Autoreg KF [39] | 0.1 | 11.52±0.83 | 35.51±2.66 | 1.10±0.07 | 3.72±0.28 | 17.91±1.41 |
| Gain (%) | | 19.88% | 19.87% | 18.37% | 19.76% | 19.78% |

TABLE IV: Runtime and model size comparison. Inference measured on a single NVIDIA RTX 6000 (batch=1, seq=100, averaged over 200 runs after 50 warm-up passes).

| Method | #Params | ms/step |
|------------------------------|---------|---------|
| <i>Deep Learning Methods</i> | | |
| FW-NKF (ours) | 607,023 | 0.686 |
| BayesKNet [28] | 106,279 | 2.882 |
| Recursive KNet [29] | 7,351 | 1.222 |
| <i>Neural Kalman Methods</i> | | |
| Kalman Net [27] | 15,476 | 0.742 |
| Recurrent KNet [38] | 25,421 | 0.570 |
| <i>Classical Methods</i> | | |
| Classical KF [19] | 28 | 0.276 |
| Autoreg KF [39] | 61 | 0.272 |

encourages these matrices to better preserve the frequency content of the true observations. For neural baselines (Kalman-Net, BayesKNet, Recursive KNet, Recurrent KNet), the same loss regularizes their parameterizations. We vary λ_Φ across methods to ensure a consistent and fair comparison of how each strategy benefits from frequency-domain supervision.

VI. DISCUSSION

A. Performance Analysis Across Benchmarks

The results across Tables I–III suggest that the benefits of FW-NKF depend on the spectral structure of the underlying estimation problem. On the pendulum benchmark (Table I), FW-NKF achieved the lowest MAE (0.708 ± 0.08) relative to BayesKNet (1.193 ± 0.065), corresponding to a 40.7% reduction in error. This improvement is consistent with the role of the frequency-domain reconstruction loss, which reduces high-frequency fluctuations in the observed Cartesian

TABLE V: Performance comparison of Kalman filtering methods on EuRoC MAV IMU-based state estimation. Recursive KNet is excluded due to memory constraints (>47 GB).

| Method | λ_Φ | Performance Metrics | | | | |
|------------------------------|----------------|---------------------|------------------|------------------|------------------|------------------|
| | | MAE ↓ | MSE ↓ | RMSE ↓ | NRMSE ↓ | R ² ↑ |
| <i>Deep Learning Methods</i> | | | | | | |
| FW-NKF (ours) | 0.0 | 0.12±0.00 | 0.03±0.00 | 0.19±0.00 | 0.11±0.01 | 0.99±0.00 |
| FW-NKF (ours) | 0.01 | 0.12±0.01 | 0.04±0.00 | 0.19±0.01 | 0.11±0.00 | 0.99±0.00 |
| FW-NKF (ours) | 0.1 | 0.13±0.01 | 0.04±0.00 | 0.20±0.01 | 0.11±0.00 | 0.99±0.00 |
| BayesKNet [28] | 0.0 | 0.75±0.04 | 2.36±0.36 | 1.53±0.11 | 0.86±0.01 | 0.26±0.02 |
| BayesKNet [28] | 0.01 | 0.75±0.04 | 2.36±0.36 | 1.53±0.11 | 0.86±0.01 | 0.26±0.02 |
| BayesKNet [28] | 0.1 | 0.75±0.04 | 2.36±0.36 | 1.53±0.11 | 0.86±0.01 | 0.26±0.02 |
| <i>Neural Kalman Methods</i> | | | | | | |
| Kalman Net [27] | 0.0 | 0.68±0.03 | 1.81±0.24 | 1.34±0.09 | 0.75±0.06 | 0.43±0.09 |
| Kalman Net [27] | 0.01 | 0.68±0.03 | 1.81±0.24 | 1.34±0.09 | 0.75±0.06 | 0.43±0.09 |
| Kalman Net [27] | 0.1 | 0.68±0.03 | 1.81±0.24 | 1.34±0.09 | 0.75±0.06 | 0.43±0.09 |
| Recurrent KNet [38] | 0.0 | 0.73±0.02 | 2.26±0.29 | 1.50±0.09 | 0.85±0.01 | 0.28±0.02 |
| Recurrent KNet [38] | 0.01 | 0.73±0.02 | 2.26±0.29 | 1.50±0.09 | 0.85±0.01 | 0.28±0.02 |
| Recurrent KNet [38] | 0.1 | 0.73±0.02 | 2.26±0.29 | 1.50±0.09 | 0.85±0.01 | 0.28±0.02 |
| <i>Classical Methods</i> | | | | | | |
| Classical KF [19] | 0.0 | 0.82 | 2.42 | 1.56 | 0.92 | 0.15 |
| Classical KF [19] | 0.01 | 0.85 | 2.13 | 1.46 | 0.87 | 0.25 |
| Classical KF [19] | 0.1 | 1.35±0.70 | 6.72±5.46 | 2.30±1.19 | 1.23±0.58 | -0.86±1.43 |
| Autoreg KF [39] | 0.0 | 0.72±0.06 | 1.98±0.57 | 1.39±0.19 | 0.78±0.06 | 0.39±0.10 |
| Autoreg KF [39] | 0.01 | 0.72±0.06 | 1.98±0.57 | 1.39±0.19 | 0.78±0.06 | 0.39±0.10 |
| Autoreg KF [39] | 0.1 | 0.72±0.06 | 1.98±0.57 | 1.39±0.19 | 0.78±0.06 | 0.39±0.10 |
| Gain (%) | | 82.35% | 98.34% | 85.82% | 85.33% | 130.23% |

tip position while preserving the lower-frequency dynamics that are relevant for state tracking.

The gains are even larger on the Lorenz system (Table II), where FW-NKF reduced MAE from 4.24 ± 0.24 for Recursive KNet to 0.50 ± 0.15 . One possible explanation is that chaotic systems are especially sensitive to measurement corruption, so suppressing noise-dominated spectral components can improve both observation quality and downstream state propagation. More broadly, these results suggest that frequency-aware innovation modeling is particularly useful when measurement noise overlaps with nonlinear or unstable dynamics, where small observation errors can rapidly accumulate over time.

B. Limitations of Baselines

Classical KF performance degrades across datasets, consistent with violations of the white Gaussian noise assumption. Neural Kalman baselines improve on the classical filter, but still underperform FW-NKF, suggesting that learned filtering alone is insufficient when noise is frequency-dependent.

BayesKNet is more variable across benchmarks, performing competitively on pendulum but degrading substantially on Lorenz. This pattern suggests that variational uncertainty estimation alone does not provide enough structure to handle spectrally complex and unstable dynamics.

C. Component Contribution Analysis

a) *Frequency-Domain Loss*: The spectral reconstruction loss (Eq. 8) demonstrates clear benefits across datasets. In UIP-DB (Table III), FW-NKF achieves 4.15 ± 0.20 RMSE for acceleration estimation versus Recursive KNet’s 5.18 ± 0.31 —a 19.9% improvement resulting from guiding the observation network $H(\cdot)$ toward noise-free IMU reconstruction.

b) *IIR Innovation Filtering*: The learnable IIR innovation filter improves noise suppression. On UIP-DB, orientation MAE drops to 1.34 ± 0.09 compared to 2.77 ± 0.18 for

the classical KF, consistent with the learned coefficients attenuating dominated noise bands before the Kalman update.

c) Frequency Weight Selection: Performance is best at moderate frequency weights ($\lambda = 0.01\text{--}0.05$), indicating that mild spectral guidance is sufficient. Larger λ values reduce performance, as seen on the pendulum benchmark where MAE increases from 0.708 ± 0.075 at $\lambda = 0.01$ to 1.205 ± 0.179 at $\lambda = 5.0$. This suggests that excessive spectral emphasis can interfere with the temporal objectives of sequential state estimation.

VII. CONCLUSION

We presented FW-NKF, a frequency-aware neural Kalman filtering framework that improves state estimation by shaping the innovation signal and supervising observation learning in the spectral domain. Across chaotic systems and real-world sensor-fusion benchmarks, the results show that explicitly modeling frequency-dependent noise can yield more accurate and consistent estimates than conventional Kalman variants. This work highlights that state estimation should not treat all measurement error as spectrally uniform. In real-world, disturbances often concentrate in specific bands, and exploiting that structure improves robustness in downstream tasks such as navigation, motion tracking, and control. The fact that moderate spectral weighting also benefits multiple differentiable Kalman baselines suggests that frequency-aware objectives may provide a general direction for building filters that better match the statistics of real sensor data.

REFERENCES

- [1] T. Haarnoja, V. Pong, A. Zhou, M. Dalal, P. Abbeel, and S. Levine, "Composable deep reinforcement learning for robotic manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6244–6251.
- [2] X. Yi, Y. Zhou, and F. Xu, "TransPose: Real-time 3d human translation and pose estimation with six inertial sensors," *ACM Transactions on Graphics (TOG)*, 2021.
- [3] S. Gong and et al., "SpiralNet++: A fast and highly efficient mesh convolution operator," in *IEEE/CVF International Conference on Computer Vision Workshops (ICCV-W)*, 2019.
- [4] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas, "HuMoR: 3d human motion model for robust pose estimation," in *International Conference on Computer Vision (ICCV)*, 2021.
- [5] C. Chen and et al., "IONet: Learning to cure the curse of drift in inertial odometry," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [6] K. Zhou, B. L. Bhatnagar, J. E. Lenssen, and G. Pons-Moll, "TOCH: Spatio-temporal object-to-hand correspondence for motion refinement," in *European Conference on Computer Vision (ECCV)*, 2022.
- [7] R. G. Krishnan, U. Shalit, and D. Sontag, "Deep Kalman filters," *arXiv preprint arXiv:1511.05121*, 2015.
- [8] —, "Structured inference networks for nonlinear state space models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.
- [9] A. Alemi and et al., "Learning nonlinear dynamics in efficient, balanced spiking networks using local plasticity rules," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [10] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press, 1982, vol. 2.
- [11] E. Hazan, H. Lee, K. Singh, C. Zhang, and Y. Zhang, "Spectral filtering for general linear dynamical systems," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [12] M. O'Connor, H. Millares, P. Banerjee, G. Gorospe, and N. Endres, "Vibration-based health characterization of multiple IMUs in UAV applications," Mar. 2020. [Online]. Available: <https://ntrs.nasa.gov/citations/20205000223>
- [13] D. Middleton, "Statistical-physical models of electromagnetic interference," *IEEE Transactions on Electromagnetic Compatibility*, 1977.
- [14] S. Gu, S. Levine, I. Sutskever, and A. Mnih, "MuProp: Unbiased backpropagation for stochastic neural networks," in *International Conference on Learning Representations*, 2016.
- [15] A. Kloss, G. Martius, and J. Bohg, "How to train your differentiable filter," *Autonomous Robots*, vol. 45, pp. 561–578, 2021.
- [16] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013, vol. 3.
- [17] A. Corenflos and et al., "Differentiable particle filtering via entropy-regularized optimal transport," in *Proceedings of the 38th International Conference on Machine Learning*, 2021.
- [18] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.
- [19] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME—Journal of Basic Engineering*, 1960.
- [20] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Department of Computer Science, Tech. Rep., 1995.
- [21] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [22] G. Einicke and L. White, "Robust extended kalman filtering," *IEEE Transactions on Signal Processing*, 1999.
- [23] S. Konatowski and et al., "Comparison of estimation accuracy of ekf, ukf and pf filters," *Annual of Navigation*, 2016.
- [24] D. Freirich and et al., "Perceptual Kalman filters: Online state estimation under a perfect perceptual-quality constraint," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- [25] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Wiley, 2001.
- [26] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proceedings of SPIE – Signal Processing, Sensor Fusion, and Target Recognition VI*, 1997.
- [27] G. Revach and et al., "KalmanNet: Neural network aided Kalman filtering for partially known dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.
- [28] Y. Dahan and et al., "Uncertainty quantification in deep learning based Kalman filters," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [29] H. Mortada and et al., "Recursive KalmanNet: Deep learning-augmented Kalman filtering for state estimation with consistent uncertainty quantification," *arXiv preprint arXiv:2506.11639*, 2025.
- [30] R. G. Krishnan, U. Shalit, and D. Sontag, "Deep kalman filters," *arXiv preprint arXiv:1511.05121*, 2015.
- [31] M. Karl and et al., "Deep variational bayes filters: Unsupervised learning of state space models from raw data," in *International Conference on Learning Representations (ICLR)*, 2017.
- [32] G. A. Einicke, "Optimal and robust noncausal filter formulations," *IEEE Transactions on Signal Processing*, pp. 1069–1077, 2006.
- [33] M. Looney, "Analyzing frequency response of inertial MEMS in stabilization systems," *Analog Dialogue*, vol. 46, no. 3, 2012. [Online]. Available: <https://www.analog.com/en/resources/analog-dialogue/articles/analyzing-frequency-response-of-inertial-mems.html>
- [34] C. Carlevaris-Bianco et al., "University of michigan north campus long-term vision and lidar dataset," *International Journal of Robotics Research*, 2016.
- [35] M. Orsag and et al., "State estimation, robust control and obstacle avoidance for multicopter in cluttered environments: Euroc experience and results," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.
- [36] R. Armani, C. Qian, J. Jiang, and C. Holz, "Ultra inertial poser: Scalable motion capture and tracking from sparse inertial sensors and ultra-wideband ranging," in *ACM SIGGRAPH 2024 Conference Papers*.
- [37] M. Burri et al., "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, pp. 1157–1163, 2016.
- [38] P. Becker and et al., "Recurrent Kalman networks: Factorized inference in high-dimensional deep feature spaces," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.
- [39] Y. Dahan, "Autoregressive Kalman filter," 2022, software available at <https://github.com/yonatandn/AutoRegKF>. [Online]. Available: <https://github.com/yonatandn/AutoRegKF>