K-PRISM: Kalman-based Physics-informed Residual Innovation with Sensor Multimodality

Anonymous authors
Paper under double-blind review

Abstract

Classical Kalman filtering is optimal under known linear-Gaussian dynamics, but real systems exhibit partially-known dynamics, unknown observation models, and time-varying sensor reliability. Neural-augmented filters such as KalmanNet relax the need for known noise statistics, yet rely on recurrent gain estimation that adapts slowly to abrupt sensor degradation and tends to memorize training-specific spatial corrections. We propose **K-PRISM** (Kalman-based Physics-informed Residual Innovation with Sensor Multimodality), which (i) decomposes the state transition into a known physics prior and a learned residual, (ii) computes a *translationally invariant* gain from instantaneous velocity and innovation features, enabling single-timestep adaptation without spatial overfitting, and (iii) extends to heterogeneous multi-sensor fusion via cross-modal gating. The resulting update generalizes the structure of the classical multi-sensor Kalman update, with Lipschitz-bounded innovations and formal error contraction. Empirically, K-PRISM matches or exceeds KalmanNet's accuracy while using 20 to 200× fewer parameters and eliminating its catastrophic-failure mode. On the Lorenz attractor, it matches steady-state accuracy while recovering from sensor dropout with zero failures; in a controlled two-sensor study under alternating reliability, the cross-modal gate identifies the informative sensor online, beating independent fusion by 5.2 dB. On real-world NCLT localization it converges in every run, whereas our KalmanNet reproduction diverges in 47%, and it transfers to visual-odometry velocity filtering on EuRoC. These results position compact, translationally invariant filtering as a robust alternative to recurrent gain estimation.

1 Introduction

State estimation, the recursive inference of a system's latent state from noisy observations, is a foundational problem across robotics, navigation, target tracking, and signal processing (Revach et al., 2022). The Kalman filter (Kalman, 1960) and its nonlinear extensions provide a principled predict-update recursion that is provably optimal under known linear-Gaussian dynamics. In practice, three properties of real systems erode this optimality. The state-transition model is often only partially known: an approximate physics model exists, but it carries discretization error and unmodeled dynamics (Shi et al., 2019). The observation model may be unknown, particularly when measurements are high-dimensional sensor data with no closed-form mapping to the state. And sensor reliability is heterogeneous and time-varying, as different sensors degrade at different times, abruptly and unpredictably.

Neural-augmented filters and differentiable algorithms (Krishnan et al., 2015; Jonschkowski et al., 2018) address these gaps while preserving the algorithmic priors of classical filters. KalmanNet (Revach et al., 2022) replaces the analytic Kalman gain with a recurrent network that learns the gain from data, retaining the predict-update recursion but removing the requirement for known noise statistics. This design has proven effective and has spawned extensions for split gain estimation, latent observation spaces, and transformer-based gain computation (Choi et al., 2023; Buchnik et al., 2023; Shen et al., 2025). Two structural

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

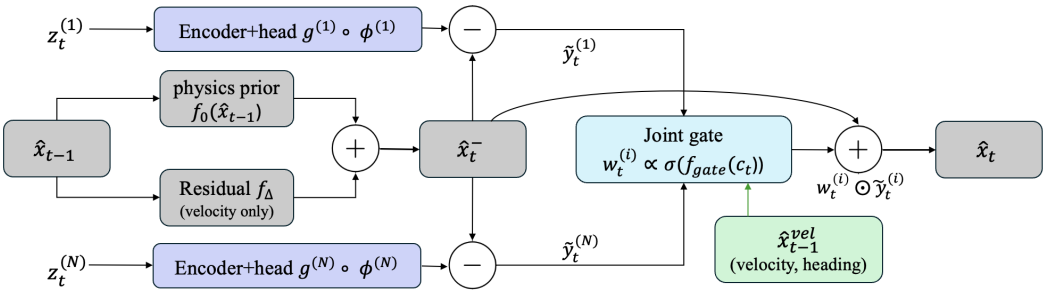

Figure 1: **K-PRISM architecture.** The prediction combines a known physics prior f_0 with a learned residual f_Δ that sees only relative quantities. Each sensor is encoded to a state proposal $\hat{\mathbf{x}}_t^{(i)}$, differenced against the prediction to form an innovation $\tilde{\mathbf{y}}_t^{(i)}$. A joint gate computes per-sensor, per-dimension weights from the combined context and applies them in a Kalman-style update. The gate is conditioned on velocity and innovation but never on absolute position, which yields translational invariance and single-timestep adaptation.

limitations persist. The recurrent gain adapts slowly: an abrupt sensor degradation takes several timesteps to propagate through the hidden state, during which estimates degrade. And a gain conditioned on absolute position can memorize a spatial correction map tied to the training trajectories, generalizing poorly to unseen regions.

We propose K-PRISM (Kalman-based Physics-informed Residual Innovation with Sensor Multimodality), a neural-augmented filtering framework built on three design principles (Figure 1). First, the state transition is decomposed into a known physics prior f_0 and a learned residual f_Δ, exploiting partial model knowledge while compensating for complex unmodeled dynamics. Second, the gain is computed by a feedforward network conditioned only on relative quantities (velocity, heading, and the current innovation), never absolute position; this translational invariance prevents position-dependent memorization and enables single-timestep adaptation, eliminating the recurrent module. Third, for N heterogeneous sensors, a joint gating network computes per-sensor fusion weights from the combined feature context, allowing a degraded modality to be suppressed within a single timestep, analogous to dynamic routing. We provide formal guarantees: the residual update generalizes the structure of the classical multi-sensor Kalman update (Proposition 1), spectral normalization bounds the innovation magnitude (Proposition 2), and the estimation error contracts under mild Lipschitz conditions (Proposition 3).

We validate K-PRISM across synthetic and real-world benchmarks, spanning the chaotic Lorenz attractor, NCLT localization (Carlevaris-Bianco et al., 2016), and EuRoC visual-odometry filtering, where it matches or exceeds KalmanNet at 20 to 200× fewer parameters (Section 4). Our contributions are: (i) a filtering framework that jointly accommodates partially-known dynamics, unknown observation models, and heterogeneous multi-sensor fusion, unifying capabilities that prior KalmanNet variants address individually; (ii) a translationally invariant gain that replaces recurrent covariance tracking with an instantaneous feedforward computation, providing both fast adaptation and improved generalization; and (iii) formal stability guarantees, and evidence that this compact (∼1.4K-parameter) filter eliminates the training-instability and slow-adaptation failure modes of recurrent gains.

2 Background and Related Work

We review the classical Kalman filter and its multi-sensor extension (Sec. 2.1), summarize KalmanNet and its neural-augmented descendants (Sec. 2.2), and identify the open gaps that motivate K-PRISM (Sec. 2.3).

2.1 Classical Kalman Filtering

Consider a discrete-time linear state-space model

$$\mathbf{x}_t = \mathbf{F}\,\mathbf{x}_{t-1} + \mathbf{e}_t, \qquad \mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \tag{1a}$$

$$\mathbf{z}_t = \mathbf{H}\,\mathbf{x}_t + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}), \tag{1b}$$

with latent state $\mathbf{x}_t \in \mathbb{R}^m$, observation $\mathbf{z}_t \in \mathbb{R}^n$, transition matrix $\mathbf{F}$, observation matrix $\mathbf{H}$, and noise covariances $\mathbf{Q}$, $\mathbf{R}$. The Kalman filter (Kalman, 1960) produces the MMSE estimate of $\mathbf{x}_t$ through a predict-update recursion. The prediction propagates the state and covariance, $\hat{\mathbf{x}}_t^- = \mathbf{F}\hat{\mathbf{x}}_{t-1}$ and $\mathbf{P}_t^- = \mathbf{F}\mathbf{P}_{t-1}\mathbf{F}^\top + \mathbf{Q}$. The update then corrects the prediction using the innovation $\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_t^-$ and innovation covariance $\mathbf{S}_t$:

$$\mathbf{S}_t = \mathbf{H}\,\mathbf{P}_t^-\,\mathbf{H}^\top + \mathbf{R}, \tag{2}$$

$$\mathbf{K}_t = \mathbf{P}_t^-\,\mathbf{H}^\top\,\mathbf{S}_t^{-1}, \tag{3}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t\tilde{\mathbf{y}}_t, \tag{4}$$

with $\mathbf{P}_t = (\mathbf{I}_m - \mathbf{K}_t\mathbf{H})\mathbf{P}_t^-$. The Kalman gain $\mathbf{K}_t$ optimally balances the prediction against the observation according to their respective uncertainties. When $N \geq 2$ sensors observe the state, the optimal centralized update fuses all sensors jointly. In information form, the posterior covariance and estimate are

$$\mathbf{P}_t = \Big((\mathbf{P}_t^-)^{-1} + \textstyle\sum_{i=1}^N (\mathbf{H}^{(i)})^\top (\mathbf{R}_t^{(i)})^{-1}\mathbf{H}^{(i)}\Big)^{-1}, \qquad \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \textstyle\sum_{i=1}^N \mathbf{K}_t^{(i)}\,\tilde{\mathbf{y}}_t^{(i)}, \tag{5}$$

with $\tilde{\mathbf{y}}_t^{(i)} = \mathbf{z}_t^{(i)} - \mathbf{H}^{(i)}\hat{\mathbf{x}}_t^-$ and per-sensor gain $\mathbf{K}_t^{(i)} = \mathbf{P}_t(\mathbf{H}^{(i)})^\top(\mathbf{R}_t^{(i)})^{-1}$ (Bar-Shalom et al., 2001). Crucially, each gain uses the *posterior* covariance $\mathbf{P}_t$, which couples the sensors: the corrections cannot be computed independently and summed, since doing so would overcount the shared prior information. Optimality of this update requires $\mathbf{F}$, $\mathbf{H}^{(i)}$, $\mathbf{Q}$, and $\mathbf{R}^{(i)}$ all known. The Extended Kalman Filter (EKF) (Housein et al., 2022) handles nonlinear systems by linearizing the transition and observation functions, but this still requires analytic models that are rarely available when observations are high-dimensional sensor data such as images or RF spectra (Clark et al., 2017).

2.2 Neural-Augmented Kalman Filtering and State Estimation

KalmanNet (Revach et al., 2022) replaces the analytic gain with a Gated Recurrent Unit that maps innovation and state-update features $(\Delta\hat{\mathbf{x}}_t, \Delta\tilde{\mathbf{x}}_t)$ to the gain $\mathbf{K}_t = \mathrm{GRU}_\theta\big(\Delta\hat{\mathbf{x}}_t, \Delta\tilde{\mathbf{x}}_t, \mathbf{g}_{t-1}\big)$, with the recurrent hidden state $\mathbf{g}_{t-1}$ serving as an implicit proxy for the error covariance. The predict-update recursion is otherwise preserved, removing the dependence on known noise statistics.

Subsequent work extends this paradigm along several axes. Split-KalmanNet (Choi et al., 2023) factorizes the gain into two RNNs, one for the predicted covariance and one for the innovation covariance, improving robustness to compound model mismatch. Latent-KalmanNet (Buchnik et al., 2023) introduces a learned encoder that maps high-dimensional observations to a latent space with a postulated linear observation model, relaxing the known-H requirement. KalmanFormer (Shen et al., 2025) replaces the GRU with a Transformer to capture longer-range temporal dependencies, and Recursive KalmanNet (Mortada et al., 2025) adds an explicit covariance branch for uncertainty quantification. Most recently, AM-KNet (Mehrfard et al., 2026) (concurrent work, subsequently withdrawn) introduced sensor-specific modules for radar, lidar, and camera, an early multi-sensor extension in the family. Most relevant to our multi-sensor setting, Song et al. (Song et al., 2024) apply KalmanNet to integrated navigation, fusing GPS, wheel odometry, and IMU; we adopt their benchmark and compare against it directly in Section 4.2. These methods share two properties that limit their applicability to our setting: all but Latent-KalmanNet require a known observation model, and all compute the gain with a recurrent module whose hidden state requires multiple timesteps to adapt after an abrupt change in sensor reliability. Our empirical comparison focuses on the canonical KalmanNet, the most widely used and reproduced neural filter, plus classical baselines; more recent variants (Split-KalmanNet, KalmanFormer, Adaptive KalmanNet) target orthogonal axes and retain the recurrent gain whose cost we analyze, so comparing against them is a natural next step.

Beyond the KalmanNet family, differentiable filters learn components by backpropagating through the recursion (Haarnoja et al., 2016; Jonschkowski et al., 2018; Kloss et al., 2021). Closely related, physics-informed hybrids embed a known dynamical model inside a differentiable Kalman recursion: HKR-Net (Li et al., 2024), for instance, couples a vehicle dynam-

Table 1: Structural comparison of neural-aided Kalman filters. "Partial" denotes a known model plus a learned residual; the last column lists unresolved gaps among C1–C3. †K-PRISM's encoder-head is dimensionality-agnostic; we demonstrate low-dimensional encoders and leave high-dimensional ones to future work.

Model	f / $\mathbf{F}$	h / $\mathbf{H}$	Gain mechanism	Gain structure	Open
Classical KF (Kalman, 1960)	Known	Known	Riccati recursion	Full $\mathbf{K}_t$	C1, C2
KalmanNet (Revach et al., 2022)	Known	Known	GRU on innovations	Full vector	C1, C2, C3
Split-KalmanNet (Choi et al., 2023)	Known	Known	2 RNNs (split $\mathbf{P}$, $\mathbf{S}$)	$\hat{\mathbf{P}}^{-}\mathbf{H}^{\top}\hat{\mathbf{S}}^{-1}$	C1, C2, C3
Latent-KalmanNet (Buchnik et al., 2023)	Approx.	Learned	GRU in latent space	Full vector	C2, C3
KalmanNet-Fusion (Song et al., 2024)	Known	Known	GRU (dead-reckoning prior)	Full vector	C1, C3
KalmanFormer (Shen et al., 2025)	Known	Known	Transformer	Full vector	C1, C2, C3
AM-KNet (Mehrfard et al., 2026)	Known	Known	Sensor-specific RNNs	Per-sensor $\mathbf{K}_t^{(i)}$	C1, C3
K-PRISM (ours)	**Partial**	**Learned**	**MLP on context + innov.**	**Diag. per-sensor**	**None†**

ics model with a differentiable EKF and recurrent decoders for trajectory prediction. Such methods rely on a fully known transition model and a recurrent gain, whereas K-PRISM assumes only a partial transition (physics prior plus learned residual) and computes a feed-forward, translationally invariant gain. Separately, end-to-end approaches (Wang et al., 2021; Teed & Deng, 2021) map raw sensor data directly to state estimates but discard the algorithmic priors of Bayesian filtering, often generalizing worse than structured filters.

2.3 Open Gaps and Motivation for K-PRISM

Table 1 summarizes the structural assumptions of each model. Three compounding challenges remain unaddressed by any single existing approach. (C1) Unknown observation models with partial transition knowledge. An approximate physics model is often available for the transition, but the mapping from observations to the state may have no closed form. While physics-informed learning incorporates nominal models into networks (Shi et al., 2019; Raissi et al., 2019), current differentiable filters rarely exploit partial transition knowledge while also learning the observation pathway. K-PRISM learns the observation mapping through a modular encoder, demonstrated on low-dimensional observations and extensible to higher-dimensional modalities via a domain-appropriate encoder. (C2) Heterogeneous multi-sensor fusion with per-sensor gain. Resolving reliability disagreements between physically distinct sensors calls for a per-sensor gain that can suppress one modality without affecting others. Existing multi-sensor KalmanNet variants use a single unified gain over combined inputs (Song et al., 2024) or assume known dynamics (the withdrawn AM-KNet); a per-sensor, per-dimension gain under partial dynamics remains open. (C3) Instantaneous adaptation to modality failure. Recurrent gains adapt over several timesteps; a gain conditioned directly on the current features and innovation can respond immediately.

K-PRISM addresses all three simultaneously. It decomposes the transition into a known physics prior and a learned residual, exploiting partial model knowledge while compensating for unmodelled dynamics (C1). The observation pathway is learned end-to-end through modality-specific encoders and spectral-normalized heads (the learned-H component of C1). A per-sensor, per-dimension gating mechanism computes fusion weights from the instantaneous feature context and innovation magnitudes, enabling single-timestep adaptation without recurrent covariance tracking (C2 and C3); we validate this cross-modal gating in a controlled two-sensor study (Section 4, Appendix B.1), with real-data validation on single-sensor tasks and joint multi-sensor fusion left to future work.

3 Methodology

3.1 Problem Formulation

Consider a discrete-time system whose latent state $\mathbf{x}_t \in \mathbb{R}^m$ evolves as $\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{e}_t$, where f is a possibly nonlinear transition and $\mathbf{e}_t$ is zero-mean process noise with unknown statistics. At each timestep, $N \geq 1$ sensors produce observations through sensor-specific, potentially high-dimensional and unknown mappings,

$$\mathbf{z}_t^{(i)} = h^{(i)}(\mathbf{x}_t) + \mathbf{v}_t^{(i)}, \quad i \in \{1, \ldots, N\}, \tag{6}$$

with $\mathbf{z}_t^{(i)} \in \mathbb{R}^{n_i}$ and zero-mean measurement noise $\mathbf{v}_t^{(i)}$ whose statistics may be non-Gaussian and time-varying. We operate under three assumptions that distinguish K-PRISM from prior work. (A1) Partially known transition: an approximate physics model f_0 is available but carries discretization error and unmodelled dynamics. (A2) Unknown observation models: no analytic expression for any $h^{(i)}$ is available, as arises with high-dimensional sensor data. (A3) Heterogeneous sensor reliability: noise characteristics differ across sensors and may change abruptly. The objective is to estimate $\hat{\mathbf{x}}_t$ by fusing $\{\mathbf{z}_t^{(i)}\}_{i=1}^N$ through a predict-update recursion that inherits the structure of the Kalman filter while accommodating A1 to A3.

3.2 Single-Sensor K-PRISM

We first derive K-PRISM for a single sensor ($N = 1$), suppressing the superscript (i). This base case already extends KalmanNet by relaxing both the known-H and fully-known-f requirements.

Prediction via physics prior and learned residual. We replace the linear transition $\hat{\mathbf{x}}_t^- = \mathbf{F}\hat{\mathbf{x}}_{t-1}$ with a known physics prior f_0 plus a learned residual f_Δ:

$$\hat{\mathbf{x}}_t^- = f_0(\hat{\mathbf{x}}_{t-1}) + f_\Delta(\hat{\mathbf{x}}_{t-1}^{\mathrm{rel}}), \tag{7}$$

where f_Δ is a lightweight MLP receiving only the position-independent components $\hat{\mathbf{x}}_{t-1}^{\mathrm{rel}}$ (velocity, heading, angular rate, and any control inputs). Restricting f_Δ to relative quantities makes the residual correction translationally invariant: identical dynamics produce identical corrections regardless of absolute position. The prior f_0 may be a known dynamical model (e.g. a Lorenz ODE step), a sensor-driven kinematic model (e.g. dead reckoning), or a generic constant-velocity extrapolation.

Learned observation encoding. Because no analytic $h(\cdot)$ is available (A2), we learn the observation mapping via an encoder $\phi : \mathbb{R}^n \to \mathbb{R}^{d_h}$ and a spectral-normalized head $g : \mathbb{R}^{d_h} \to \mathbb{R}^m$ (Miyato et al., 2018):

$$\mathbf{h}_t = \phi(\mathbf{z}_t), \qquad \hat{\mathbf{x}}_t^{(\mathrm{obs})} = g(\mathbf{h}_t) = \boldsymbol{\Theta}_2\,\mathrm{ReLU}(\boldsymbol{\Theta}_1\mathbf{h}_t) \in \mathbb{R}^m, \tag{8}$$

with $\|\boldsymbol{\Theta}_k\|_2 \le 1$, so g is 1-Lipschitz (Proposition 2). The composition $g \circ \phi$ replaces $\mathbf{H}$: the effective observation matrix is $\mathbf{H}^{\mathrm{eff}} = \mathbf{I}_m$ and the innovation is $\tilde{\mathbf{y}}_t = \hat{\mathbf{x}}_t^{(\mathrm{obs})} - \hat{\mathbf{x}}_t^-$, analogous to $\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_t^-$.

Translationally invariant gain. KalmanNet computes the gain with a GRU that implicitly tracks covariance. K-PRISM instead uses a feedforward network conditioned only on position-independent quantities.

Definition 1 (Translational invariance)**.** A gain $\mathbf{w}_t = \mathcal{G}(\hat{\mathbf{x}}_{t-1}, \tilde{\mathbf{y}}_t)$ is translationally invariant if for any translation $\boldsymbol{\tau}$ of the position components, $\mathcal{G}(\hat{\mathbf{x}}_{t-1} + [\boldsymbol{\tau}; \mathbf{0}], \tilde{\mathbf{y}}_t) = \mathcal{G}(\hat{\mathbf{x}}_{t-1}, \tilde{\mathbf{y}}_t)$.

Two distinct properties are at work: translational invariance, from conditioning on relative quantities, prevents spatial memorization and aids generalization; memorylessness, from replacing the recurrent hidden state with an instantaneous feedforward map, enables single-timestep adaptation. K-PRISM combines both, but they are separable, a recurrent gain can be translationally invariant, and a feedforward gain can be position-dependent. This is enforced architecturally by restricting the gain input:

$$\mathbf{w}_t = \sigma\big(\psi_{\mathrm{gain}}\big([\hat{\mathbf{x}}_{t-1}^{\mathrm{vel}} \,\|\, \tilde{\mathbf{y}}_t]\big)\big) \in (0,1)^m, \tag{9}$$

where $\hat{\mathbf{x}}_{t-1}^{\mathrm{vel}} \subseteq \hat{\mathbf{x}}_{t-1}^{\mathrm{rel}}$ contains only the velocity, heading, and angular-rate components (excluding control inputs) and ψ_{gain} is a two-layer network. Without this constraint, the gain can memorize a spatial correction map that generalizes poorly. By conditioning only on relative quantities, the gain learns a transferable policy: how much to trust the observation given the current velocity and innovation. We validate this in Section 4.2, where the position-dependent variant degrades from 20.64 to 23.03 dB on GPS-only NCLT and from 9.43 to 36.01 m on multi-sensor fusion. The gain bias is initialized to -2.0 ($\sigma(-2) \approx 0.12$),

a conservative prior favoring the physics prediction initially. The posterior estimate is

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{w}_t \odot \tilde{\mathbf{y}}_t. \tag{10}$$

Setting $\mathbf{H} = \mathbf{I}_m$, this recovers the classical update under $\mathbf{K}_t \leftrightarrow \mathrm{diag}(\mathbf{w}_t)$, with the sigmoid constraining the update to a convex combination of prior and observation.

A controlled study (Appendix C) shows this restriction is at worst cost-free: a position-conditioned gain never significantly beats the invariant gain, since the innovation already signals reliability locally; in practice (Sec. 4.2) it is worse, as exploiting position requires a larger, overfitting network.

3.3 Multi-Sensor Extension

The single-sensor estimator does not capture the central difficulty of fusion: when one sensor degrades, the others must compensate within the same timestep. Running N independent filters leaves each blind to the others, and a single filter on concatenated observations cannot assess per-sensor reliability. Drawing on gated multimodal fusion (Arevalo et al., 2017), K-PRISM gives each sensor a dedicated encoder and head but makes the gating decision jointly, so each sensor's weight is informed by all sensors' features (full discussion in Appendix B).

Each sensor i has encoder $\phi^{(i)}$ and spectral-normalized head $g^{(i)}$, producing per-sensor state proposals $\hat{\mathbf{x}}_t^{(i)} = g^{(i)}(\phi^{(i)}(\mathbf{z}_t^{(i)}))$ and innovations $\tilde{\mathbf{y}}_t^{(i)} = \hat{\mathbf{x}}_t^{(i)} - \hat{\mathbf{x}}_t^-$, all in the shared state space $\mathbb{R}^m$. The gating context concatenates position-independent features from all sensors,

$$\mathbf{c}_t = \left[\mathbf{h}_t^{(1)} \,\|\, \cdots \,\|\, \mathbf{h}_t^{(N)} \,\|\, \hat{\mathbf{x}}_{t-1}^{\mathrm{vel}} \,\|\, \tilde{\mathbf{y}}_t^{(1)} \,\|\, \cdots \,\|\, \tilde{\mathbf{y}}_t^{(N)}\right], \tag{11}$$

and a confidence network produces normalized per-sensor, per-dimension weights:

$$\boldsymbol{\alpha}_t^{(i)} = \sigma\Big(f_{\mathrm{gate}}^{(i)}(\mathbf{c}_t)\Big), \qquad \mathbf{w}_t^{(i)} = \frac{\boldsymbol{\alpha}_t^{(i)}}{\sum_{j=1}^{N} \boldsymbol{\alpha}_t^{(j)} + \epsilon}. \tag{12}$$

Here $\epsilon > 0$ is a small constant preventing division by zero. The normalization ensures $\sum_i \mathbf{w}_t^{(i)} \leq \mathbf{1}$ component-wise, bounding the total correction. The posterior estimate is

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \sum_{i=1}^{N} \mathbf{w}_t^{(i)} \odot \tilde{\mathbf{y}}_t^{(i)}. \tag{13}$$

Because the innovations measure discrepancies rather than absolute positions, translational invariance holds for the full multi-sensor mechanism: when sensor j degrades (large $\|\tilde{\mathbf{y}}_t^{(j)}\|$), the joint context lets the gate suppress it while maintaining trust in healthy sensors, within a single timestep.

Proposition 1 (Multi-Sensor Structure as a Learned Convex Combination)**.** Let $\mathbf{H}^{(i)} = \mathbf{I}_m$ for all i. Under the normalization of Eq. 12, the update Eq. 13 is a per-dimension convex combination of prior and sensor proposals,

$$\hat{x}_{t,k} = \Big(1 - \textstyle\sum_i w_{t,k}^{(i)}\Big)\hat{x}_{t,k}^- + \sum_i w_{t,k}^{(i)}\, \hat{x}_{t,k}^{(i)}, \qquad \sum_i w_{t,k}^{(i)} \leq 1, \tag{14}$$

with non-negative coefficients. It shares the additive per-sensor correction structure of the classical update Eq. 5 under $\mathbf{z}_t^{(i)} \leftrightarrow \hat{\mathbf{x}}_t^{(i)}$ and $\mathbf{K}_t^{(i)} \leftrightarrow \mathrm{diag}(\mathbf{w}_t^{(i)})$, but replaces the covariance-derived gain with a learned normalized weight whose convex-combination form bounds the total correction, unlike an unnormalized sum of independent gains.

K-PRISM thus does not reproduce the optimal centralized MMSE fusion of Eq. 5, which needs the noise covariances our setting lacks; it instead learns a bounded convex combination adapting to non-Gaussian, time-varying reliability without an explicit noise model (Appendix D).

3.4 Stability Guarantees

We state two guarantees; proofs are in Appendix A.

Proposition 2 (Lipschitz-Bounded Innovations)**.** Let $g^{(i)}$ be a two-layer head with spectral-normalized weights ($\|\boldsymbol{\Theta}_k^{(i)}\|_2 \leq 1$); then $g^{(i)}$ is 1-Lipschitz. Consequently the correction

Table 2: Lorenz steady-state MSE [dB] (5 seeds, lower is better).

Method	MSE [dB]
Raw observations	-20.03
EKF	-25.16
K-PRISM	-26.31 ± 0.06
KalmanNet	-27.20 ± 0.18

Table 3: Lorenz dropout recovery over 30 sequences: steps to 2× pre-dropout error and catastrophic failure rate (lower is better).

Method	Recovery	Fail %
EKF	3.9 ± 3.7	0%
K-PRISM	$\mathbf{3.9 \pm 3.0}$	**0**%
KalmanNet	15.4 ± 23.4	3.3%

$\mathbf{w}_t^{(i)} \odot \tilde{\mathbf{y}}_t^{(i)}$ is bounded in the encoder feature $\mathbf{h}_t^{(i)}$: a bounded change in features yields a proportionally bounded change in the correction, so it cannot respond discontinuously to a feature perturbation.

Proposition 3 (Contraction of Estimation Error)**.** Let $\mathbf{e}_t = \mathbf{x}_t - \hat{\mathbf{x}}_t$ and let the composite transition be L_f-Lipschitz (Assumption 1). By Eq. 12, $\sum_i w_{t,k}^{(i)} \in [0,1]$; suppose in the informative regime it is bounded below, $\min_k \sum_i w_{t,k}^{(i)} \geq \gamma_{\min} > 0$. Then $\|\mathbf{e}_t\|_2 \leq \rho \|\mathbf{e}_{t-1}\|_2 + \delta_t$ with $\rho = L_f(1 - \gamma_{\min})$ and δ_t bounded by the sensor-noise and encoder-approximation magnitude. Contraction ($\rho < 1$) holds whenever $\gamma_{\min} > 1 - 1/L_f$; for a non-expansive prior ($L_f \leq 1$) any $\gamma_{\min} > 0$ suffices. When the gate suppresses all sensors ($\sum_i w_{t,k}^{(i)} \to 0$) the bound reduces to open-loop $\|\mathbf{e}_t\|_2 \lesssim L_f \|\mathbf{e}_{t-1}\|_2 + \delta_t$, contractive only if $L_f < 1$. Contraction is thus a property of the informative regime with a positive gain floor; during dropout, stability rests on the prior, consistent with the transient error growth in Figure 2.

4 Experimental Results

4.1 Synthetic Benchmark: Lorenz Attractor

We first validate K-PRISM on the chaotic Lorenz attractor (Lorenz, 2017), a standard benchmark for nonlinear filtering and a rigorous testbed for data-driven sequence models evaluating distribution shifts (Gilpin, 2021; Revach et al., 2022). The state $\mathbf{x}_t = [x, y, z]^\top$ evolves under the Lorenz ODE ($\sigma = 10$, $\rho = 28$, $\beta = 8/3$), integrated by RK4 and decimated to create discretization mismatch. Observations are the full state corrupted by Gaussian noise at -20 dB. K-PRISM uses a single RK4 step as the physics prior f_0, a learned residual f_Δ, and the instantaneous gain of Eq. 9; KalmanNet uses the same known dynamics with its recurrent gain. We report MSE normalized by the state variance, in dB.

Steady-state tracking. Table 2 reports accuracy over 5 seeds with clean observations. KalmanNet's recurrent gain holds a slight edge, as its accumulated covariance suits stationary conditions; K-PRISM attains comparable accuracy with a purely feedforward gain.

Robustness to sensor dropout. While the recurrent gain aids steady-state accuracy, it introduces a reliability cost under abrupt sensor degradation, a known vulnerability of recurrent architectures facing corrupted modalities (Baltrušaitis et al., 2018). We inject catastrophic noise (8× amplitude) over timesteps $[50, 100)$ and measure recovery, the number of steps for the position error to return to twice its pre-dropout level, over 30 sequences (Table 3). On the median sequence, all three filters recover comparably. However, KalmanNet's recovery is highly variable (std 23.4 steps) and occasionally fails to recover within the observation window (3.3%, 1 of 30), as its recurrent hidden state must be flushed of the corrupted observations before tracking resumes. In contrast, both K-PRISM and the EKF, whose gains depend only on instantaneous quantities, recover reliably in every sequence (0% failure, std ≤ 3.7).

Figure 2 shows both the outcome and the mechanism: after clean observations resume, KalmanNet's error decays more slowly than K-PRISM's or the EKF's (left), reflecting hidden-state lag; K-PRISM's learned gain dips during dropout to down-weight corrupted observations, then returns immediately (right), a single-timestep adaptation unavailable to a recurrent gain.

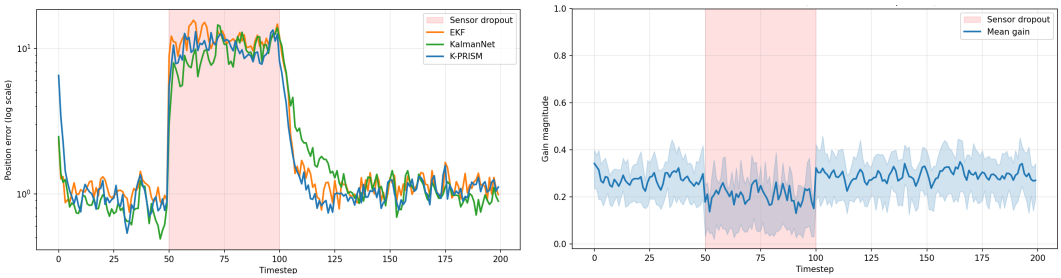

Figure 2: **Left:** worst-case (90th-percentile) position error over 30 dropout sequences. After the dropout window (shaded), KalmanNet (green) decays more slowly than K-PRISM (blue) and the EKF (orange). **Right:** K-PRISM's learned gain (mean over 30 sequences) suppresses the corrupted observations during the dropout window and recovers immediately afterward.

Table 4: MSE [dB] on NCLT GPS-only (lower is better). Mean±std over 10 seeds.

Method	**MSE [dB]**	**Params**
Vanilla RNN (Revach et al., 2022)	40.21	—
EKF (Revach et al., 2022)	25.39	—
KalmanNet (Revach et al., 2022)	22.20	∼29K
K-PRISM	$\mathbf{20.64 \pm 0.37}$	**∼1.4K**

4.2 Real-World Localization: NCLT Dataset

We evaluate K-PRISM on the NCLT dataset (Carlevaris-Bianco et al., 2016), a benchmark for autonomous vehicle localization with 27 trajectories in indoor/outdoor environments.

4.2.1 GPS-Only Localization

We instantiate the translationally invariant gain of Section 3.2 with a constant-velocity prior. To prevent the network from memorizing position-dependent corrections, a known vulnerability when neural networks process absolute spatial coordinates (Liu et al., 2018), both the gain and the residual are conditioned exclusively on relative quantities. The full model requires only ∼1,400 parameters. Following (Revach et al., 2022), we use session `2012-01-22` at 1 Hz with a constant-velocity model ($\mathbf{x}_t = [p_x, p_y, v_x, v_y]^\top$, $\mathbf{y}_t = [p_x^{\text{gps}}, p_y^{\text{gps}}]^\top$), an 85/10/5% train/val/test split, and overlapping training sequences ($T = 200$, stride 50). Table 4 reports MSE in dB ($10 \log_{10}(\text{MSE})$, lower is better) over 10 seeds. K-PRISM achieves 20.64±0.37 dB, outperforming KalmanNet (22.20 dB) by 1.56 dB with 21× fewer parameters. Furthermore, in this data-limited regime, the structural prior is decisive: unstructured sequence models (LSTMs, causally-masked Transformers) overfit and trail K-PRISM even at matched capacity (Appendix F.3).

4.2.2 Multi-Sensor Fusion

We also evaluate on the multi-sensor setting of (Song et al., 2024). We reproduce KalmanNet with the authors' public code[1] (details in Appendix F.2). K-PRISM achieves 9.43 ± 0.88 m (best: 8.20 m). The published KalmanNet result on this split is 8.01 m (Song et al., 2024), which we could not reproduce with the authors' public code: under its default configuration the baseline averages 13.79 m and diverges in 47% of runs (Appendix F.2). We therefore frame K-PRISM's advantage not as beating the best published mean, which it does not, but as convergence stability (0% vs. 47%) and parameter efficiency (∼500 vs. 107K); the reproduced baseline used the default unregularized script, whereas K-PRISM used cosine annealing, weight decay, and gradient clipping. During indoor segments GPS is zeroed via the dataset's preprocessed validity mask rather than by the learned gate, so this benchmark

[1] `https://github.com/SongJgit/KalmanNet4SensorFusion`

evaluates fusion under known GPS availability; autonomous online detection of GPS loss from innovations alone is demonstrated separately in Appendix B.1.

Table 5: EuRoC Machine Hall: velocity RMSE [m/s] (3 seeds, lower is better). "KalmanNet+res" adds K-PRISM's residual to KalmanNet to isolate the gain. Relative position error (Appendix G) follows the same ordering.

Method	**MH_01**	**MH_02**	**MH_03**	**MH_04**	**MH_05**
Raw VO	0.372	0.642	0.953	0.930	1.153
VO-EKF	0.370	0.638	0.942	0.886	1.140
KalmanNet (Revach et al., 2022)	0.365	0.630	0.877	1.525	1.149
KalmanNet+res	0.276	0.436	0.832	0.798	1.033
K-PRISM	**0.254**	**0.400**	**0.820**	**0.798**	**1.032**

4.3 Real-World Visual Odometry Velocity Filtering: EuRoC

To test K-PRISM on a second real-world domain, we evaluate on the EuRoC MAV benchmark (Burri et al., 2016), which provides synchronized stereo imagery and IMU with millimeter-accurate ground truth. A stereo VO front-end (ORB features (Rublee et al., 2011), triangulation, frame-to-frame PnP) produces camera motion; frame-to-frame VO drifts in absolute position but is reliable short-term, so we use it as a *velocity* observation. This is visual-odometry velocity filtering, not full VIO: the IMU is not integrated into propagation or update, so we benchmark against a kinematic (constant-velocity) EKF. The state is $[\mathbf{p}, \mathbf{v}] \in \mathbb{R}^6$ and we report velocity RMSE over 3 seeds (full details in Appendix G).

K-PRISM achieves the lowest error on all five Machine Hall sequences (Table 5), outperforming KalmanNet, the EKF, and raw VO; KalmanNet additionally diverges on MH_04, consistent with the recurrent-gain instability seen on NCLT. The residual drives most of the improvement, with the invariant gain adding a smaller further gain on the easy sequences.

5 Limitations

First, translational invariance assumes position-independent reliability; systems with genuinely position-dependent reliability (e.g., GPS multipath) may benefit from controlled spatial conditioning, a trade-off we leave open. Second, without a reasonable physics prior f_0 the framework reduces to learning the full transition, forfeiting the inductive bias behind its sample efficiency. Third, extending the gating mechanism to asynchronous, multi-rate sensors remains future work. Fourth, the Lipschitz condition for error contraction (Proposition 3) is encouraged via weight decay rather than structurally enforced. Fifth, K-PRISM outputs a point estimate without an explicit posterior covariance, which downstream tasks such as planning and SLAM often require; an auxiliary uncertainty head is a natural extension. Sixth, because the gain is memoryless it maintains no representation of accumulated uncertainty during extended outages, so after a long blackout a large drift-induced innovation is not readily distinguished from an outlier, and the contraction guarantee applies only in the informative regime.

6 Conclusion

We introduced K-PRISM, a neural-augmented filtering framework for partially-known dynamics, unknown observation models, and time-varying sensor reliability. Its translationally invariant gain, computed from velocity and innovation, prevents spatial memorization and enables single-timestep adaptation without recurrent modules. We showed this generalizes the structure of the multi-sensor Kalman update, with Lipschitz-bounded innovations and formal error contraction. Empirically, across the chaotic Lorenz attractor, NCLT localization, and EuRoC visual-odometry filtering, K-PRISM matches or exceeds KalmanNet's accuracy while using 20 to 200× fewer parameters and removing its two central failure modes: slow recovery from abrupt degradation and unstable training. These results position compact, translationally invariant filtering as a robust, parameter-efficient alternative to recurrent gain estimation.

Use of Large Language Models

We used an LLM to **improve the readability** of portions of the manuscript through copy-editing and to **format tables and references**. We also used an LLM for **literature-search assistance**, whose outputs we independently verified before citing. In addition, we used an LLM to **review, debug, and refine our code and for reproducibility auditing**: the authors wrote the experiment code, and the LLM assisted in checking and improving it, cross-checking the reported results against re-run outputs, and identifying inconsistencies between the manuscript and the released code, which we then corrected and verified against the code. We did not use generative AI tools to develop the method, formulate or prove the theoretical results, generate the synthetic datasets, or design the experimental methodology; these are the authors' own work. We have reviewed all AI-assisted work: all code was written by the authors and verified and tested for correctness by the authors, all empirical claims were confirmed by direct execution of the released code, and all AI-assisted text and citations were checked by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

Reproducibility Statement

We have taken several steps to ensure our results are reproducible. All code required to reproduce every experiment is provided as anonymized supplementary material, with per-table run commands, a pinned dependency list, and pointers to the official public sources for the NCLT (Carlevaris-Bianco et al., 2016), EuRoC (Burri et al., 2016), and multi-sensor fusion (Song et al., 2024) datasets (we do not redistribute the data). The full model architecture is specified in Section 3 (Eqs. 7–13), and complete training and preprocessing details such as optimizer, learning-rate schedule, epochs, early-stopping criteria, sequence lengths, and data splits are given in Appendices E, F.6, and G, with the exact per-experiment settings matching the released code. All reported numbers are means over multiple seeds (5 or 10, depending on the experiment, stated in each table), and our KalmanNet comparison is reproduced from the authors' public code under its documented default configuration (Appendix F.2), including a per-seed breakdown of its convergence behavior. The two stability guarantees (Propositions 2 and 3) are stated with explicit assumptions and proved in Appendix A.

Ethics Statement

This work develops a general-purpose state-estimation method and is evaluated solely on publicly available benchmark datasets (NCLT (Carlevaris-Bianco et al., 2016), EuRoC (Burri et al., 2016), and the integrated-navigation data of (Song et al., 2024)) that contain no personally identifiable information and no human-subject data; we use them under their respective public licenses and terms. We foresee no direct harmful applications beyond the general dual-use considerations common to localization and navigation research. Our empirical claims are reported with multiple seeds and, where we reproduce prior work, we do so transparently under its default configuration and report outcomes we could and could not replicate. We declare no conflicts of interest.

References

John Arevalo, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A González. Gated multimodal units for information fusion. *arXiv preprint arXiv:1702.01992*, 2017.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

Itay Buchnik, Guy Revach, Damiano Steger, Ruud JG Van Sloun, Tirza Routtenberg, and Nir Shlezinger. Latent-kalmannet: Learned kalman filtering for tracking from high-dimensional signals. *IEEE Transactions on Signal Processing*, 72:352–367, 2023.

Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9):1023–1035, 2016.

Geon Choi, Jeonghun Park, Nir Shlezinger, Yonina C Eldar, and Namyoon Lee. Split-kalmannet: A robust model-based deep learning approach for state estimation. *IEEE transactions on vehicular technology*, 72(9):12326–12331, 2023.

Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

William Gilpin. Chaos as an interpretable benchmark for forecasting and data-driven modelling. *arXiv preprint arXiv:2110.05266*, 2021.

Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop kf: Learning discriminative deterministic state estimators. *Advances in neural information processing systems*, 29, 2016.

Alaa Aldeen Housein, Gao Xingyu, Weiming Li, and Yang Huang. Extended kalman filter sensor fusion in practice for mobile robot localization. *International Journal of Advanced Computer Science and Applications*, 13(2), 2022.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. *arXiv preprint arXiv:1805.11122*, 2018.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Alina Kloss, Georg Martius, and Jeannette Bohg. How to train your differentiable filter. *Autonomous Robots*, 45(4):561–578, 2021.

Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Zhenni Li, Hui Sun, Dong Xiao, and Hongfei Xie. Hybrid kalman recurrent neural network for vehicle trajectory prediction. *IEEE Transactions on Instrumentation and Measurement*, 73:1–14, 2024.

Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *Advances in neural information processing systems*, 31, 2018.

Edward N Lorenz. Deterministic nonperiodic flow 1. In *Universality in Chaos, 2nd edition*, pp. 367–378. Routledge, 2017.

Arian Mehrfard, Bharanidhar Duraisamy, Stefan Haag, Florian Geiss, and Mirko Mählisch. Adaptive learned state estimation based on kalmannet. *arXiv preprint arXiv:2604.02441*, 2026.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Hassan Mortada, Cyril Falcon, Yanis Kahil, Mathéo Clavaud, and Jean-Philippe Michel. Recursive kalmannet: Deep learning-augmented kalman filtering for state estimation with consistent uncertainty quantification. In *2025 33rd European Signal Processing Conference (EUSIPCO)*, pp. 885–889. IEEE, 2025.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud JG Van Sloun, and Yonina C Eldar. Kalmannet: Neural network aided kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pp. 2564–2571. IEEE, 2011.

Siyuan Shen, Jichen Chen, Guanfeng Yu, Zhengjun Zhai, and Pujie Han. Kalmanformer: using transformer to model the kalman gain in kalman filters. *Frontiers in Neurorobotics*, 18:1460255, 2025.

Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *2019 international conference on robotics and automation (icra)*, pp. 9784–9790. IEEE, 2019.

Jian Song, Wei Mei, Yunfeng Xu, Qiang Fu, and Lina Bu. Practical implementation of kalmannet for accurate data fusion in integrated navigation. *IEEE Signal Processing Letters*, 31:1890–1894, 2024.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. In *Conference on Robot Learning*, pp. 1761–1772. PMLR, 2021.

A Proofs and Stability Details

A.1 Proof of Proposition 2 (Lipschitz-Bounded Innovations)

Let $g^{(i)} : \mathbb{R}^{d_h} \to \mathbb{R}^m$ be the two-layer sensor head with both weight matrices spectral-normalized, $\|\mathbf{\Theta}_k^{(i)}\|_2 \le 1$ for $k \in \{1, 2\}$. By submultiplicativity of the spectral norm and non-expansiveness of ReLU,

$$\left\|g^{(i)}(\mathbf{h}_1) - g^{(i)}(\mathbf{h}_2)\right\|_2 \le \|\mathbf{\Theta}_2^{(i)}\|_2 \, \|\mathbf{\Theta}_1^{(i)}\|_2 \, \|\mathbf{h}_1 - \mathbf{h}_2\|_2 \le \|\mathbf{h}_1 - \mathbf{h}_2\|_2, \tag{15}$$

so $g^{(i)}$ is Lipschitz with $L^{(i)} \le 1$.

Corollary (Bounded State Correction). Since $\mathbf{w}_t^{(i)} \in (0, 1)^m$ and $\|\tilde{\mathbf{y}}_t^{(i)}\|_2 \le \|g^{(i)}(\mathbf{h}_t^{(i)})\|_2 + \|\hat{\mathbf{x}}_t^-\|_2$, the per-sensor correction obeys

$$\left\|\mathbf{w}_t^{(i)} \odot \tilde{\mathbf{y}}_t^{(i)}\right\|_2 \le \left\|\tilde{\mathbf{y}}_t^{(i)}\right\|_2 \le \left\|\mathbf{h}_t^{(i)}\right\|_2 + \left\|\hat{\mathbf{x}}_t^-\right\|_2. \tag{16}$$

The correction is therefore bounded in terms of the encoder feature norm $\|\mathbf{h}_t^{(i)}\|_2$ and the prior norm. We note that the encoder $\phi^{(i)}$ is not spectral-normalized, so this guarantee holds in feature space: it bounds the head's response to changes in the features $\mathbf{h}_t^{(i)}$, not the composite response to the raw observation $\mathbf{z}_t^{(i)}$, since $\phi^{(i)}$ may amplify an outlier before the head. Spectral-normalizing $\phi^{(i)}$ (or adding a bounded activation) would extend the guarantee to the raw observation; we leave this as a design option.

A.2 Proof of Proposition 3 (Error Contraction)

Assumption 1. The composite transition $f_0 + f_\Delta$ is L_f-Lipschitz: $\|f_0(\mathbf{a}) + f_\Delta(\mathbf{a}^{\text{vel}}) - f_0(\mathbf{b}) - f_\Delta(\mathbf{b}^{\text{vel}})\|_2 \le L_f \|\mathbf{a} - \mathbf{b}\|_2$, where $\mathbf{a}^{\text{vel}}$ denotes the velocity/heading components of $\mathbf{a}$. For stable dynamics $L_f \le 1$; chaotic systems (e.g. Lorenz) may have $L_f > 1$.

Because f_Δ operates on a lower-dimensional subspace (velocity rather than full state), this condition is easier to satisfy in practice than Lipschitz continuity over the full state space, and is encouraged by weight decay on f_Δ.

Proof. Let $\mathbf{e}_t^- = \mathbf{x}_t - \hat{\mathbf{x}}_t^-$. By Assumption 1 and the triangle inequality, $\|\mathbf{e}_t^-\|_2 \le L_f \|\mathbf{e}_{t-1}\|_2 + \|\boldsymbol{\eta}_t\|_2$, where $\boldsymbol{\eta}_t$ collects process noise and model mismatch. Writing $\mathbf{W}_t = \sum_i \operatorname{diag}(\mathbf{w}_t^{(i)})$, the update Eq. 13 gives

$$\mathbf{e}_t = (\mathbf{I}_m - \mathbf{W}_t)\, \mathbf{e}_t^- - \sum_{i=1}^{N} \mathbf{w}_t^{(i)} \odot \left(\hat{\mathbf{x}}_t^{(i)} - \mathbf{x}_t\right). \tag{17}$$

Because Eq. 12 normalizes the weights so that each diagonal entry $W_{t,kk} = \sum_i w_{t,k}^{(i)} \in [0, 1]$, the diagonal matrix $\mathbf{I}_m - \mathbf{W}_t$ has non-negative entries and

$$\|\mathbf{I}_m - \mathbf{W}_t\|_2 = \max_k (1 - W_{t,kk}) = 1 - \min_k W_{t,kk} \le 1 - \gamma_{\min}. \tag{18}$$

(The bound $W_{t,kk} \le 1$ from normalization is essential here; without it the retention factor is not controlled.) Taking norms and using the a priori bound,

$$\|\mathbf{e}_t\|_2 \le (1 - \gamma_{\min})\|\mathbf{e}_t^-\|_2 + \delta_t \le \underbrace{L_f(1 - \gamma_{\min})}_{\rho} \|\mathbf{e}_{t-1}\|_2 + \delta_t, \tag{19}$$

where $\delta_t \le (1-\gamma_{\min})\|\boldsymbol{\eta}_t\|_2 + \sup_x \sum_i \|\mathbf{w}_t^{(i)} \odot (g^{(i)}(\phi^{(i)}(h^{(i)}(x))) - x)\|_2$ collects process noise and the per-sensor encoder-approximation error. Hence $\rho < 1$ iff $\gamma_{\min} > 1 - 1/L_f$. The lower bound $\gamma_{\min}$ is what drives contraction: a minimum fraction of every state dimension must be corrected each step. During dropout the gate drives $W_{t,kk} \to 0$, so $\|\mathbf{I}_m - \mathbf{W}_t\|_2 \to 1$ and the error evolves open-loop. □

B Why Joint Multi-Sensor Gating is Necessary

A natural question is whether multi-sensor fusion can be handled by combining independent single-sensor estimators. We consider three candidate architectures. **(1) Independent filters, late combination:** run N single-sensor K-PRISM instances in parallel and combine their outputs with a learned layer. Each filter computes its gain from its own features only, so when sensor j degrades, the filters for $i \neq j$ are unaware and cannot compensate within the current timestep. **(2) Single filter on concatenated observations:** concatenate all observations and learn one unified gain. This loses per-sensor reliability assessment, as there is no per-sensor weight to suppress a single degraded modality without affecting healthy ones. **(3) Joint gating (K-PRISM):** each sensor has a dedicated encoder and head, but the confidence network receives features from all sensors when computing each sensor's weight, so $\mathbf{w}_t^{(1)}$ is informed by the state of sensor 2 and vice versa.

The advantage of (3) is cross-modal conditioning: each sensor's reliability is assessed in the context of all others. When sensor 1 degrades, the gate sees both its anomalous features and sensor 2's healthy features and redistributes weight within one timestep. Architectures (1) and (2) cannot, the former because each filter is blind to the others, the latter because there is no per-sensor decomposition to redistribute. Joint gating also provides a structural guarantee absent from the alternatives: because the weights are normalized across sensors (Eq. 12), the total correction magnitude is bounded by construction, as required for the contraction result in Proposition 3.

Table 6: Two-sensor Lorenz under alternating reliability (5 seeds, MSE [dB], lower is better). All variants share the same prior, residual, encoders, and heads; only the gain computation differs. Parameter counts here reflect the two-sensor Lorenz configuration with full per-sensor encoders, larger than the ∼1,400-parameter GPS-only and ∼500-parameter fusion NCLT models.

Fusion architecture	**Params**	**MSE [dB]**
Independent filters	14.9K	-13.81 ± 1.47
Concatenated	7.7K	-17.56 ± 0.32
Joint gating (K-PRISM)	14.8K	$\mathbf{-19.02 \pm 0.38}$

B.1 Controlled Study: Alternating Reliability on the Lorenz Attractor

To isolate the value of K-PRISM's cross-modal joint gating, we construct a controlled two-sensor benchmark on the Lorenz attractor with alternating reliability: at each moment exactly one of the two sensors is reliable (-18 dB) while the other is nearly uninformative ($+2$ dB), and the roles swap every 20 steps. There is no redundancy, so the filter must identify which sensor to trust at each timestep purely from the observed innovations.

We compare the three fusion architectures described above. All variants share an identical physics prior (a single RK4 step), identical learned residual, and identical per-sensor encoders and spectral-normalized heads; only the gain computation differs. Training uses Adam (lr 10^{-3}, cosine annealing, 150 epochs) on 50 training and 10 validation sequences of length $T = 100$. Table 6 reports test MSE over 5 seeds on 30 held-out sequences of length $T = 200$.

Joint gating outperforms concatenation by 1.5 dB and independent filters by 5.2 dB, both well beyond the seed variation. The ordering is intuitive: independent filters are forced to average in the uninformative sensor at every step, concatenation can partially suppress it through the shared encoder but has no per-sensor weight to route around it, and only joint gating can place weight on whichever sensor is currently reliable.

Figure 3 visualizes this: the learned per-sensor weight tracks the alternating reliable sensor, rising for each sensor during the intervals when it is informative. This confirms that the joint gate learns to identify sensor reliability online from innovations alone, the capability that single-sensor and naive multi-sensor baselines lack.

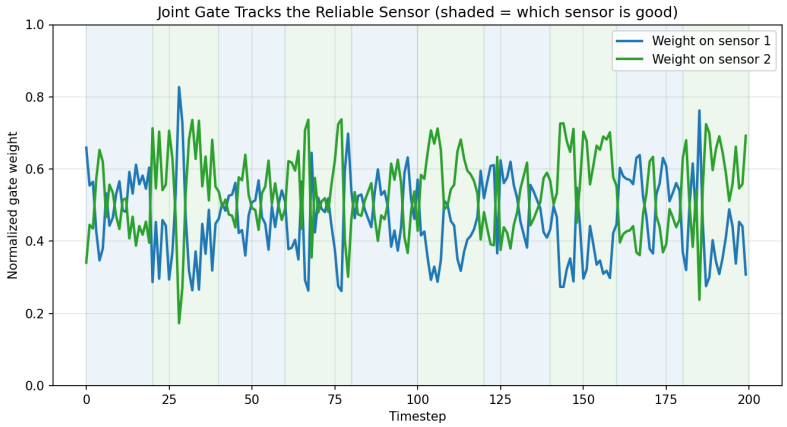

Figure 3: Joint gate weights on a held-out alternating-reliability sequence. Shaded regions indicate the reliable sensor (blue: sensor 1, green: sensor 2), swapping every 20 steps. The gate weight for each sensor rises when that sensor is reliable, showing that joint gating identifies sensor reliability online from innovations alone.

Table 7: Position-conditioned vs. invariant gain on the controlled tracking task (test RMSE, 8 seeds, lower is better). Across both unreliability regimes and all levels of position-dependence α, the two gains are statistically indistinguishable (differences within one standard deviation), so absolute position provides no benefit.

Regime	α	**Invariant gain**	**Position-conditioned**
Noise	0.00	0.207 ± 0.002	0.207 ± 0.002
	0.25	1.093 ± 0.009	1.089 ± 0.007
	0.50	1.690 ± 0.038	1.685 ± 0.042
	0.75	2.131 ± 0.095	2.114 ± 0.083
	1.00	2.470 ± 0.115	2.455 ± 0.129
Bias	0.00	0.207 ± 0.002	0.207 ± 0.002
	0.25	0.61 ± 0.02	0.61 ± 0.02
	0.50	1.086 ± 0.054	1.117 ± 0.055
	0.75	1.474 ± 0.024	1.463 ± 0.030
	1.00	1.830 ± 0.043	1.824 ± 0.039

C When Does Translational Invariance Help?

Translational invariance is K-PRISM's central design choice, so we characterize when conditioning the gain on absolute position could help and whether it does. We construct a controlled 2D tracking task in a bounded arena where sensor reliability depends on absolute position to a tunable degree $\alpha \in [0, 1]$: at α=0 reliability is uniform, and at α=1 it is a strong, fixed spatial map. We compare two otherwise-identical filters (same constant-velocity prior, learned residual, encoder, and head) that differ only in the gain input: the invariant gain conditions on velocity and innovation, while the position-conditioned gain additionally receives absolute position and can in principle learn the spatial map. Test trajectories start from unseen positions, so the position-conditioned gain must generalize the map rather than memorize training paths.

We study two forms of position-dependent unreliability. In the *noise* regime, in-region readings are high-variance (up to a 40× noise-std contrast); in the *bias* regime, in-region readings carry a low-noise systematic offset, so the raw measurement looks locally plausible. These probe the two ways a sensor can be unreliable: visibly inconsistent, or confidently wrong.

Table 7 reports test RMSE across α for both regimes (8 seeds). In every setting, conditioning the gain on absolute position does not significantly outperform the invariant gain:

all differences fall within one standard deviation. The invariant gain matches the position-conditioned gain even when reliability is strongly and genuinely position-dependent. The mechanism is that the innovation, evaluated against the physics-informed prediction, already provides a sufficient local signal of unreliability: a high-variance reading produces a large innovation directly, and a systematically biased reading is surfaced indirectly as the prediction and the biased measurement diverge over successive steps. Absolute position is therefore redundant for gain computation in these settings. This is consistent with the NCLT ablation (Sec. 4.2), where the position-conditioned variant is not merely redundant but actively worse (23.03 vs. 20.64 dB), because exploiting position requires a larger network that overfits the training trajectories. Crucially, the two gains here are parameter-matched, so this controlled result isolates the effect of coordinate conditioning from capacity: even at equal size, absolute position provides no benefit. The NCLT degradation therefore reflects both redundant conditioning and the overfitting of the larger position-dependent network. We conclude that translational invariance is a robust and cost-free inductive bias across the position-dependent reliability structures examined here: it discards information that a learned gain cannot productively exploit, while reducing parameters and improving generalization.

D Extended Remarks

Why a learned gain outperforms the classical gain. The classical gain is MMSE-optimal when the noise covariances $\mathbf{P}_t^-$ and $\mathbf{R}_t^{(i)}$ are known and Gaussian. In our setting both are unavailable, and the noise may be structured, spatially correlated, or non-stationary. The learned gain $\mathrm{diag}(\mathbf{w}_t^{(i)})$, computed from the feature context and instantaneous innovations, adapts to these conditions without an explicit noise model.

Per-dimension gain beyond KalmanNet. The classical gain is a full dense matrix that differentiates corrections across state dimensions through cross-correlations in $\mathbf{P}_t^-$ and $\mathbf{R}_t^{(i)}$, but computing it requires both covariances. KalmanNet uses a single RNN to approximate a unified gain for one sensor. K-PRISM extends this to N heterogeneous sensors with a separate gain each, and its diagonal structure resolves confidence per state dimension, which matters when partial degradation affects only a subset of state variables.

E Training Details

Handling sensor degradation. Rather than synthetic modality masking, the multi-sensor model exploits the dataset's real GPS validity mask (Appendix F.1): during indoor segments where GPS is unavailable, the corresponding gain is set to zero, forcing reliance on wheel and IMU dead-reckoning. Standard dropout ($p = 0.1$) is applied in the residual, encoder, and gain networks throughout.

Gain and residual initialization. The gain output bias is initialized to -2.0, giving $\sigma(-2) \approx 0.12$, a conservative prior consistent with the empirically optimal gain magnitude of roughly 0.05 to 0.10 (Sec. F.5). The residual network's final layer is scaled by 0.01 with zero bias, so the physics prior dominates at initialization.

Composite loss. The framework supports an optional composite objective $\mathcal{L} = \mathcal{L}_{\text{state}} + \lambda_P \mathcal{L}_{\text{physics}} + \lambda_S \mathcal{L}_{\text{smooth}}$, where $\mathcal{L}_{\text{physics}}$ is a domain constraint and $\mathcal{L}_{\text{smooth}}$ penalizes jitter. In all experiments reported here we use pure state MSE ($\lambda_P = \lambda_S = 0$); the EuRoC model additionally weights the position term at 0.5 relative to velocity. We include the composite form to document supported extensions.

F Additional NCLT Experiments

F.1 Multi-Sensor Fusion Details

We evaluate K-PRISM on the multi-sensor setting of Song et al. (Song et al., 2024), which fuses GPS, wheel encoders, and IMU using the state vector $\mathbf{x}_t = [p_x, p_y, v_x, v_y, \theta, \omega]^\top$. We use the authors' public code and preprocessed data (22/2/3 train/val/test trajectories)[2]. The prediction step uses wheel-encoder velocities and IMU heading via the dead-reckoning model of (Song et al., 2024), with a learned residual correction. GPS corrections are applied when available; during indoor segments the preprocessed GPS validity mask sets the gain to zero. Table 8 reports results over 10 seeds.

Table 8: Test RMSE [m] on NCLT multi-sensor fusion. Lower is better.

Method	**Params**	**Mean RMSE**	**Best**	**Fail%**
EKF (Song et al., 2024)	–	13.10	–	0%
KalmanNet (reproduced)	107K	13.79 ± 1.52	12.12	47%
KalmanNet (reported) (Song et al., 2024)	107K	8.01[†]	–	–
K-PRISM	**536**	$\mathbf{9.43 \pm 0.88}$	**8.20**	**0%**

[†] Not reproduced with the authors' public code; see Sec. F.2.

K-PRISM achieves 9.43 ± 0.88 m (best 8.20 m), outperforming the reproduced KalmanNet (13.79 m) by 32% with 200× fewer parameters and zero training failures. Its best run approaches the 8.01 m originally reported by (Song et al., 2024), which we were unable to reproduce with the authors' public code (Sec. F.2).

F.2 KalmanNet Reproducibility Analysis

To ensure a fair comparison, we reproduced KalmanNet using the public code of (Song et al., 2024) with its default configuration: training sub-sequence length 50, sliding-window size 4, and hidden-state detachment every 2 steps (their `train_seq_len`, `slide_win_size`, and `detach_step`), with batch size 256, 50 epochs, and learning rate 10^{-3}. This 50-step training window matches the one used for K-PRISM fusion (Appendix F.6), so both methods are unrolled over sequences of the same length. Table 9 reports per-seed test RMSE over 15 runs.

Table 9: KalmanNet reproduction: per-seed test RMSE [m] on NCLT multi-sensor (15 seeds).

Seed	**2013-04-05**	**2012-11-16**	**2012-11-04**	**Avg**	**Status**
1	7.39	12.71	17.11	12.40	✓
2	9.40	9.68	18.24	13.23	✓
3	8.30	9.26	16.75	12.18	✓
5	6.71	6.09	18.62	12.12	✓
7	6.98	9.45	19.06	13.12	✓
8	8.83	13.52	19.23	14.70	✓
9	6.54	8.03	23.24	14.93	✓
10	11.91	16.74	20.24	16.84	✓
6	9.31	187.79	43.74	–	Diverged
11	6.10	17.34	82.35	–	Diverged
13, 15	∞	∞	∞	–	Inf. loss
4, 12, 14	–	–	–	–	No ckpt
Stable seeds (8/15):				$\mathbf{13.79 \pm 1.52}$	

Of 15 runs, 3 produced no checkpoint, 2 returned infinite loss, and 2 partially diverged, a 47% failure rate. This instability is consistent with the gradient-explosion issues discussed

[2] `https://github.com/SongJgit/KalmanNet4SensorFusion`

in (Song et al., 2024), which motivated their TBPTT strategy. The gap between the reproduced mean (13.79 m) and the reported 8.01 m suggests the latter reflects hyperparameter optimization or seed selection beyond the default configuration. In contrast, K-PRISM converged in all 10 runs (std 0.88 m), demonstrating substantially more stable training.

Table 10: Full ablation on NCLT GPS-only (10 seeds, MSE [dB], lower is better).

Variant	**Params**	**MSE [dB]**
K-PRISM	∼1.4K	20.64 ± 0.37
w/o adaptive gain (scalar)	294	20.35 ± 0.29
w/o learned gain (fixed 0.5)	292	20.82 ± 0.26
w/o invariance (pos.-dep.)	11.3K	23.03 ± 0.67
w/o physics (Pure MLP)	17.9K	22.16 ± 0.66
KalmanNet (Revach et al., 2022)	∼29K	22.20
Vanilla RNN (Revach et al., 2022)	—	40.21

F.3 GPS-Only Ablation

Table 10 extends the GPS-only ablation study with all structural variants and reference baselines. The results confirm several core design principles of K-PRISM. First, every Kalman-structured variant significantly outperforms the Vanilla RNN (40.21 dB), and removing the physics prior entirely (Pure MLP) degrades performance to 22.16 dB, confirming that preserving the structural priors of the Kalman filter is vital. Second, removing translational invariance ("w/o invariance") degrades performance by 2.39 dB due to spatial overfitting, validating our choice to condition the gain solely on relative quantities.

Finally, we evaluate the gain mechanism. Replacing the adaptive gain with a fixed scalar (learned per-dimension but constant over time) removes the ability to modulate trust per-timestep. Because sensor reliability in this specific GPS-only task is approximately stationary, this scalar gain marginally outperforms the dynamic invariant gain (20.35 vs. 20.64 dB), while fixing the gain arbitrarily ("fixed 0.5") degrades performance. Importantly, the slight advantage of the scalar gain reverses entirely in heterogeneous environments: on the multi-sensor fusion task, the scalar gain degrades performance substantially compared to the dynamic gain (14.45 vs. 9.43 m; see Appendix F.4), confirming that dynamic gating remains essential for robust multi-sensor tracking.

Table 11: Unstructured sequence-model baselines on NCLT GPS-only (10 seeds, MSE [dB], lower is better). Three capacities per family; K-PRISM and KalmanNet shown for reference on identical data and splits.

Method	**MSE [dB]**	**Params**
LSTM (small)	31.90 ± 1.09	1.3K
LSTM (medium)	29.86 ± 0.97	10K
LSTM (large)	39.04 ± 1.05	200K
Transformer (small)	42.10 ± 9.01	13K
Transformer (medium)	49.15 ± 0.68	100K
Transformer (large)	50.58 ± 1.07	596K
KalmanNet	22.20	29K
K-PRISM	$\mathbf{20.64 \pm 0.37}$	**∼1.4K**

Comparison with unstructured sequence models. A natural question is whether K-PRISM's physics-informed structure is necessary, or whether a general-purpose sequence model trained end-to-end would suffice. We compare against LSTM and Transformer baselines that map the raw observation sequence directly to the state sequence, with no physics prior, no known dynamics, and no Kalman structure (the Transformer is causally masked so it remains a valid online estimator). To avoid understating the baselines, we sweep three capacities per family and report the best. In the single-session NCLT regime (roughly

80 training sequences), the best-tuned LSTM reaches 29.86 dB and the best Transformer 42.10 dB (Table 11), both far above K-PRISM's 20.64 dB. The gap persists at matched capacity: an LSTM with the same parameter budget as K-PRISM (∼1.4K) attains only 31.90 dB. The unstructured models overfit the limited trajectories and fail to generalize to the held-out portion of the session, whereas the physics prior and the compact invariant gain let K-PRISM estimate accurately from the same data. This reflects the sample-efficiency of encoding known dynamics rather than any deficiency of the architectures, which are known to excel given far larger datasets; it indicates that in the data-limited regimes typical of state estimation, structural priors are decisive.

We note the position-conditioned variant here uses more parameters than K-PRISM, so its degradation could in principle reflect capacity rather than coordinate conditioning. The controlled study of Appendix C rules this out: there, the invariant and position-conditioned gains are parameter-matched, and the position-conditioned gain still shows no advantage across all noise and bias levels. The degradation on NCLT, therefore, reflects spatial overfitting, not merely excess capacity.

F.4 Multi-Sensor Ablation

Table 12 ablates K-PRISM on the multi-sensor task. Here the adaptive, invariant gain is decisive: K-PRISM (9.43 m) outperforms the scalar variant (14.45 m) by 5 m, because it modulates GPS trust per timestep based on innovation magnitude during indoor/outdoor transitions. The fixed gain ($w = 0.5$) performs poorly (21.91 m) since trusting GPS at 50% injects substantial noise from consumer-grade measurements. The position-dependent variant overfits (36.01 m), and Pure MLP fails entirely, confirming the importance of the invariant, physics-informed structure.

Table 12: Ablation on NCLT multi-sensor fusion (RMSE [m], lower is better).

Variant	**Params**	**Seeds**	**Mean**	**Std**
K-PRISM	536	10	**9.43**	**0.88**
w/o adaptive gain (scalar)	552	10	14.45	0.56
w/o learned gain (fixed 0.5)	550	5	21.91	0.64
w/o invariance (pos.-dep.)	11.6K	5	36.01	5.45
w/o physics (Pure MLP)	19.0K	5	–	Failed
KalmanNet (reproduced)	107K	8*	13.79	1.52

* 8 stable out of 15 runs.

F.5 Learned Gain Analysis

To interpret the gain, we examine the scalar variant, whose single per-dimension value is directly readable. Table 13 reports the learned scalar gains across 10 seeds on the multi-sensor task.

Table 13: Learned scalar gain values on NCLT multi-sensor (10 seeds).

	w_{p_x}	w_{p_y}
Mean ± Std	0.048 ± 0.003	0.095 ± 0.020
Range	[0.044, 0.052]	[0.066, 0.130]

The gains converge consistently to $w_{p_x} \approx 0.05$ and $w_{p_y} \approx 0.10$, indicating that the optimal strategy trusts wheel-encoder dead reckoning 90 to 95% and applies only small GPS corrections, consistent with the relatively high noise of consumer-grade GPS compared to wheel odometry. The asymmetry ($w_{p_y} \approx 2w_{p_x}$) may reflect directional differences in GPS accuracy due to satellite geometry. The full K-PRISM gain is not a single constant but adapts around these values per timestep, which is what enables its 5 m improvement over the scalar variant.

F.6 NCLT Training and Preprocessing

GPS-only. Adam (lr 10^{-4}, weight decay 10^{-4}) with a ReduceLROnPlateau schedule (factor 0.5, patience 20, minimum lr 10^{-6}) over up to 300 epochs, dropout $p=0.1$ in the residual, encoder, and gain networks, gradient clipping at norm 1.0, and early stopping (patience 100) on validation MSE. Training sequences have length $T=200$ with stride 50, yielding roughly 80 overlapping sequences.

Multi-sensor. Sequential truncated BPTT over a 50-step training window across the 22 training trajectories (average length ∼5000 steps), with the state detached at each window boundary. Adam (lr 10^{-4}, weight decay 10^{-4}) with a ReduceLROnPlateau schedule (factor 0.5, patience 15, minimum lr 10^{-6}) over up to 200 epochs and early stopping (patience 30).

Preprocessing. For GPS-only, NCLT ground truth is in a SLAM-derived local frame while GPS is geographic; we convert GPS to local meters, swap axes to the NED convention, and apply Procrustes alignment to register the frames. Following (Revach et al., 2022), readings exceeding 50 m error are removed (retaining ∼95% of timesteps). For multi-sensor, we use the preprocessed data of (Song et al., 2024) directly (GPS, wheel, IMU at 1 Hz with validity masks).

G Visual Odometry Velocity Filtering on EuRoC: Details

Dataset and task. The EuRoC MAV dataset (Burri et al., 2016) records a micro-aerial vehicle with a synchronized stereo camera (20 Hz) and IMU (200 Hz), with ground truth from a Leica laser tracker. We use the five Machine Hall sequences (MH_01 to MH_05), spanning easy, medium, and difficult motion. We resample to a common 50 Hz grid and split each sequence 70/15/15% (train/val/test) by time.

Visual-odometry front-end. We compute stereo VO with a standard pipeline: ORB features in both views, left-right matching and triangulation to obtain 3D points, then frame-to-frame PnP-RANSAC for relative pose, accumulated into a trajectory. This produces honest, drifting odometry with no loop closure or bundle adjustment. Because the absolute trajectory drifts (particularly in depth) while frame-to-frame displacement remains informative, we difference the VO positions to form a velocity observation and align it to the world frame by a rotation and scalar scale estimated on the first 40% of each sequence.

Models. All methods share the constant-velocity prediction and the VO-velocity observation; they differ only in the correction. The EKF uses a fixed Riccati gain with measurement variance set to the empirical VO velocity variance. KalmanNet uses a GRU gain on its standard innovation and state-update features. KalmanNet+res adds K-PRISM's learned residual to KalmanNet, leaving the GRU gain unchanged, to isolate the contribution of the gain mechanism from that of the residual. K-PRISM uses the residual and the translationally invariant feedforward gain. Training uses Adam (lr 10^{-3}, weight decay 10^{-4}), cosine annealing, 100 epochs, on 300-step windows; we report the mean over 3 seeds.

Table 14: EuRoC Machine Hall: relative position error [m] over 50-step windows (3 seeds, lower is better).

Method	MH_01	MH_02	MH_03	MH_04	MH_05
Raw VO	0.283	0.585	0.794	0.735	1.074
VO-EKF	0.284	0.583	0.788	0.738	1.075
KalmanNet (Revach et al., 2022)	0.280	0.573	0.733	0.966	1.085
KalmanNet+res	0.220	0.397	0.680	0.669	0.969
K-PRISM	**0.204**	**0.364**	**0.665**	**0.669**	**0.968**

Relative position error. Table 14 reports relative position error over 50-step windows, the standard drift metric for odometry without an absolute reference. The ordering matches the velocity RMSE in the main text: K-PRISM is best on every sequence, the residual drives

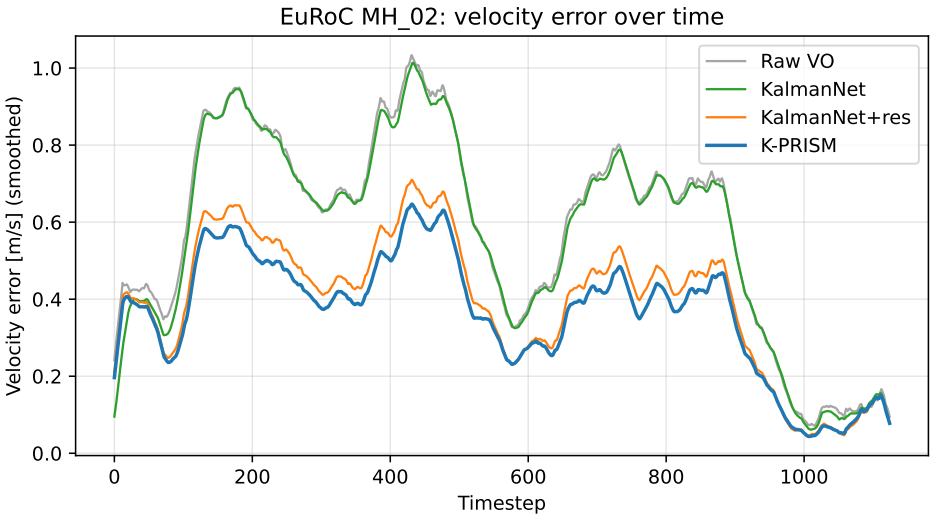

Figure 4: Velocity error over time on EuRoC MH_02 (smoothed with a 25-step moving average for legibility). Raw VO and KalmanNet track closely, since a constant-velocity model observed through a velocity measurement leaves little for the recurrent gain to exploit. Adding the physics-prior residual (KalmanNet+res) lowers the error substantially, and K-PRISM lowers it slightly further through the invariant gain. The ordering is consistent across the sequence, and the gap between the residual-augmented models and the others illustrates that the residual accounts for most of the improvement, with the invariant gain contributing a smaller additional benefit.

most of the improvement, and the invariant gain adds a small further benefit on the easy sequences.

Per-timestep behavior. Beyond the aggregate metrics, Figure 4 shows velocity error over time for one representative easy sequence (MH_02), complementing the aggregate results in Table 5. Two patterns are visible. First, raw VO and KalmanNet lie almost on top of each other throughout: with a constant-velocity prediction and a direct velocity observation, the recurrent gain has little structure to exploit and largely passes the observation through. Second, the two models that include the learned residual (KalmanNet+res and K-PRISM) separate clearly from the other two across the entire sequence, and K-PRISM sits marginally below KalmanNet+res. This visual ordering matches the component attribution in the main text: the physics-prior residual is responsible for most of K-PRISM's advantage on EuRoC, while the translationally invariant gain provides a smaller, consistent additional gain. The error for all methods rises and falls together with the difficulty of the underlying motion, confirming that the shared observation quality bounds every method and that the differences between them come from how each uses the physics prior and computes the gain.

Front-end limitations. The simple frame-to-frame VO degrades under the aggressive motion of MH_04 and MH_05, where feature tracking is intermittently lost and raw VO velocity error rises above 1.1 m/s. All fusion methods are bounded by this observation quality, which is why absolute errors grow and relative gains compress on the harder sequences. K-PRISM still improves over the raw input and over classical fusion throughout. A stronger VO or full SLAM front-end would lower the absolute errors; the comparison among fusion methods on identical observations would be unaffected. KalmanNet's divergence on MH_04 (velocity RMSE 1.53, exceeding the raw input) further illustrates the recurrent-gain instability documented on NCLT (Sec. F.2); adding the physics-prior residual removes this instability.