RESEARCH ARTICLE

Check for updates

VNNPF: A Variational Neural Network with Planar Flow for Robust IMU-GPS Fusion and Trajectory Estimation

Heran Fu[1] and **Xuebo Jin**[1,*]

[1] School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China

Abstract

Accurate state estimation for dynamic targets is essential in fields such as target tracking, navigation, and autonomous driving. However, traditional estimation models struggle to handle the nonlinear motion patterns and sensor noise prevalent in real-world environments. To address these challenges, this paper proposes a novel end-to-end estimation model named Variational Neural Network with Planar Flow (VNNPF). The model integrates a Bayesian Gated Recurrent Unit (BGRU) as the process model, a planar flow-based variational autoencoder (PFVAE) as the measurement model, and a Bayesian hyperparameter optimization module inspired by Kalman filtering. The BGRU captures nonlinear temporal dependencies and enhances robustness by modeling parameters as distributions. PFVAE transforms simple latent distributions into more complex posteriors, enabling more accurate modeling of colored noise in sensor data. The Kalman-inspired update module computes a learnable gain to fuse prior and posterior information effectively. Experiments on the KITTI IMU–GPS benchmark demonstrate that VNNPF consistently achieves lower state-estimation errors than several state-of-the-art neural network baselines. These results indicate that VNNPF can provide accurate and robust trajectory estimation for nonlinear dynamic systems with complex sensor noise.

Keywords: state estimation, bayesian recurrent neural networks, variational autoencoders, planar flow, kalman filter, IMU-GPS fusion.

1 Introduction

In various intelligent system applications, accurately acquiring the system state is critical for achieving autonomous decision-making and control. State estimation methods leverage measurement information to infer physical quantities that cannot be directly observed, forming a fundamental cornerstone of modern control and perception systems. In particular, in scenarios such as automatic control [1, 2], wireless communication [3], and robot localization [4], trajectory estimation of moving targets is not only a key technology for real-time perception and control but also a crucial component for ensuring system safety, stability, and autonomy. For example, in autonomous driving systems, sensors must perceive the vehicle state and surrounding environment in real time. Trajectory estimation assists in path planning

Submitted: 16 January 2025
Accepted: 18 December 2025
Published: 25 December 2025

Vol. 2, **No.** 1, 2026.
10.62762/TACS.2025.570823

***Corresponding author:**
Xuebo Jin
jinxuebo@btbu.edu.cn

Citation
Fu, H., & Jin, X. (2025). VNNPF: A Variational Neural Network with Planar Flow for Robust IMU-GPS Fusion and Trajectory Estimation. *ICCK Transactions on Advanced Computing and Systems*, 2(1), 25–41.

© 2025 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

and behavior prediction, thereby improving the system's responsiveness and safety. With the rapid development of sensor technology, storage devices, and wireless communication, massive trajectory data of moving objects—characterized by high dimensionality, multiple features, and strong temporal dependencies—has become increasingly available, enabling data-driven state estimation and trajectory prediction. By analyzing the trajectory patterns of moving targets over a historical time window, it is possible to effectively predict their future states, thus providing a basis for system control and risk warning.

However, in real-world applications, the uncertainty of target motion, complexity of environmental noise, and limitations in sensor stability collectively make it difficult for traditional methods to construct accurate and reliable physical models. During data acquisition in practical systems, inevitable interference and measurement errors often occur due to sensor precision and stability issues. As a result, the collected data typically exhibit strong nonlinearity and high levels of noise, making it difficult for these data to reflect the true noise distribution of real-world environments. Moreover, in complex systems, mechanism-based models tend to have a low degree of matching with actual system dynamics, leading to degraded estimation performance. These challenges significantly complicate the modeling of moving targets and consequently reduce state estimation accuracy.

Traditional estimation methods are generally grounded in probabilistic theory and rely on historical observations combined with physical modeling based on assumed motion patterns to estimate future states. However, these methods suffer from poor generalization capabilities and are often limited to specific types of motion. In practical scenarios, target motion is typically nonlinear and often accompanied by colored noise, which traditional estimation methods cannot handle effectively. As reported in classical analyses of nonlinear estimation, EKF and UKF may experience degraded performance when faced with temporally correlated or non-Gaussian noise characteristics [5]. Such methods are incapable of modeling nonlinear dynamics or characterizing non-white noise, resulting in estimation biases. For maneuvering target modeling, existing Kalman filter-based approaches such as the constant velocity (CV) model [6] and constant acceleration (CA) model [7] treat velocity and acceleration as constants. Yet, in reality, targets frequently exhibit non-uniform velocity and acceleration patterns, making it difficult for these models to accurately describe state transitions. Even more advanced models, such as the Singer model [8] or the current statistical (CS) model [9], which incorporate assumptions of acceleration variation, still struggle to adapt to the complex and agile behaviors of real-world targets. Furthermore, these models typically assume Gaussian noise, whereas real-world noise distributions are often far more complex and deviate significantly from the ideal Gaussian assumption.

Therefore, developing a state estimation model capable of handling nonlinear, non-Gaussian, and dynamically varying behaviors—while reducing dependence on predefined system model structures—constitutes the central research problem of this paper. To address this challenge, we propose an end-to-end Variational Neural Network with Planar Flow (VNNPF), which integrates a Bayesian GRU–based process model, a PF-VAE–based measurement model, and a Kalman-inspired learnable gain into a unified trajectory estimation framework.

This combination is novel in that the BGRU provides a data-driven representation of nonlinear temporal dynamics, the PF-VAE models heavy-tailed and non-Gaussian noise through flow-augmented likelihoods, and the learnable gain enables adaptive fusion between inertial predictions and measurement updates. Together, these components form a coherent probabilistic architecture that simultaneously learns system dynamics, measurement distributions, and data-dependent fusion behavior—an integration not achieved by existing filtering or deep learning approaches.

2 Related Work

2.1 Traditional Estimation Methods

Traditional state estimation methods are primarily based on physical modeling of systems and probabilistic filtering theory, and have been widely applied in practical engineering contexts. The most representative example is the Kalman Filter (KF), which achieves optimal estimation through state-space modeling under the assumption of Gaussian noise, and is well-suited for linear systems operating in white noise environments. However, real-world systems often exhibit nonlinearity and non-Gaussian disturbances, leading to significant limitations in the performance of the KF in many applications [10].

To expand the applicability of filtering techniques, researchers have proposed a range of nonlinear filters, including the Extended Kalman Filter (EKF) [11], the Unscented Kalman Filter (UKF) [12], and the Cubature Kalman Filter (CKF) [13]. These methods utilize techniques such as Taylor series expansion, sigma-point sampling, and high-order cubature transformation to handle nonlinear systems. Nevertheless, they still rely on Gaussian noise assumptions, and their performance often degrades in strongly non-Gaussian noise conditions due to limited accuracy and stability [14–16]. In contrast, the Particle Filter (PF) [17] uses importance sampling to effectively handle both nonlinearity and non-Gaussianity, demonstrating strong generalization capabilities. However, its high computational cost and susceptibility to sample degeneracy constrain its use in high-dimensional or real-time applications.

In terms of motion modeling, traditional filtering methods often rely on simplified assumptions about target dynamics, such as constant velocity (CV), constant acceleration (CA), the Singer model, or the Current Statistical (CS) model. These models assume that the evolution of the target state follows a known prior form, enabling tracking through filtering. However, actual target trajectories are often non-stationary, subject to abrupt changes, and highly uncertain, making it difficult for these models to adapt to complex maneuvering behaviors. Furthermore, the latency in responding to state changes in such models limits the estimation accuracy.

In recent years, researchers have sought to mitigate these limitations through adaptive modeling and multi-model fusion strategies. For example, Lou et al. [18] proposed an adaptive lattice CKF to reduce computational overhead and improve stability in SLAM systems; Dutta et al. [19] developed an improved UKF for denoising ECG signals, outperforming traditional methods; Nagui et al. [20] cascaded the EKF with a linear KF for GPS/INS integration, enhancing localization robustness; and Li et al. [21] introduced the MFC-KF algorithm, incorporating a maximum fuzzy entropy criterion to improve tracking performance under non-Gaussian noise conditions.

Notably, in the field of wireless sensor networks (WSNs), a review by Nguyen et al. [22] highlights that integrating unmanned aerial vehicles (UAVs) into WSNs significantly improves system performance in terms of location estimation, resource scheduling, and channel state identification. This indicates that the future development of traditional estimation methods should more explicitly address non-structural priors, adaptability to dynamic environments, and system-wide collaboration.

2.2 Neural Network-Based Estimation Methods

In recent years, with the widespread application of deep learning in fields such as image recognition, time series modeling, and target tracking, neural networks have increasingly become vital tools for state estimation in complex systems. These methods construct multi-layer nonlinear mapping functions that can extract key features from high-dimensional observational data and learn intricate dynamic relationships among variables. Numerous studies have integrated neural networks with traditional estimation techniques to propose various improvements. For instance, Jouaber et al. [23] modeled the covariance matrix within the Kalman filtering process using neural networks, enabling adaptive filtering. Li et al. [24] introduced a trajectory recognition method that integrates LSTM networks to enhance the performance of current statistical models in long-sequence modeling. Cui et al. [25] combined feedforward neural networks, recurrent neural networks, and attention mechanisms to build an end-to-end state tracking framework that better captures complex temporal features. Zhang et al. [26] proposed an LSTM-UKF hybrid model, leveraging LSTM's temporal memory to assist the Unscented Kalman Filter in predicting nonlinear trajectories. Ge et al. [27] designed an RNN-based filter gain adjustment mechanism to eliminate estimation biases caused by fixed gain settings. Wang et al. [28] incorporated convolutional neural networks into underwater target tracking to determine target observability, and used LSTM to dynamically correct estimation outputs, demonstrating strong stability in practical tests.

The introduction of generative models has further expanded the boundaries of neural network-based estimation approaches. Peng et al. [29] combined bidirectional LSTM with a VAE-GAN model to predict the remaining useful life of aerospace equipment, maintaining strong robustness under complex interference. De Migue et al. [30] developed a recurrent VAE model for UAV trajectory estimation, effectively handling issues such as missing trajectory data. Santhosh et al. [31] designed a VAE-CNN model for traffic anomaly detection, achieving outstanding results in multiple sudden event recognition tasks.

Uckan et al. [32] further extended the application of neural networks in robotic automation by building a customized deep convolutional neural network for indoor scene recognition. Trained on a dataset containing seven typical indoor environments, their model significantly improved robot recognition performance under weak texture and heavy occlusion conditions, achieving an F1-score of 89.79%. These studies collectively demonstrate the effectiveness of deep network architectures in complex perception and modeling tasks, offering a promising path for constructing multimodal state estimation frameworks in the future.

Due to their powerful representational capacity and structural flexibility, neural networks exhibit substantial advantages when dealing with non-Gaussian noise, dynamic changes, and uncertainty modeling. Current research trends are shifting from "neural network-assisted estimation" toward "unified estimation frameworks based on neural networks," which, when combined with mechanisms such as Bayesian inference and normalizing flows, hold great potential for further improving the generalization and practicality of state estimation systems.

2.3 Normalizing Flow-Based Modeling Methods

In recent years, Normalizing Flows (NF) have gained considerable attention in state estimation and trajectory modeling due to their excellent capacity for posterior distribution modeling. By applying a sequence of invertible transformations to a simple initial distribution, normalizing flows are capable of modeling complex target distributions, thereby improving the model's ability to approximate real posterior distributions. The method was first introduced by Rezende et al. [33], providing a theoretical foundation for integrating deep generative models with Bayesian inference.

Within the family of normalizing flow models, Nonlinear Independent Components Estimation (NICE) was proposed by Dinh et al. [34], employing piecewise linear transformations to model complex distributions while maintaining a computationally tractable Jacobian determinant. Subsequently, Real-valued Non-Volume Preserving (RealNVP) flows were developed, enhancing the NICE model with multi-scale structures and learnable affine transformations, thereby increasing both expressiveness and training efficiency for complex posteriors [35]. The Glow model further optimized this architecture by introducing invertible 1×1 convolutions, significantly improving latent representation and generative performance [36]. Normalizing flow methods can be categorized into two types: finite flows and infinite flows. Finite flows utilize a fixed number of invertible transformations to approximate the target distribution and are suitable for most deep generative tasks. In contrast, infinite flows model the evolution of latent variables using differential equations, offering stronger modeling capabilities at the cost of increased computational complexity. Moreover, other variants such as radial flows, hierarchical conditional flows, infinitesimal flows, and inverse autoregressive flows continue to expand the expressiveness of the NF framework.

In the domain of trajectory estimation and state modeling, normalizing flows provide an effective means to model complex observational distributions. For example, the VINFNet model combines state-space reasoning with invertible mapping networks to accurately model the posterior distribution of target trajectories [37]. Delecki et al. [38] enhanced nonlinear system state representation using normalizing flows, enabling deep networks to learn system dynamics more effectively. Papamakarios et al. [39] introduced Masked Autoregressive Flows (MAF) to improve flexibility in generative modeling. In the field of signal processing, Nugraha et al. [40] proposed an NF-VAE model, integrating VAE with normalizing flows to extract latent variables from observations and improve the accuracy of speech spectrogram modeling.

In state estimation tasks, embedding normalizing flows into Variational Autoencoder (VAE) structures has proven to be an effective approach to improve the modeling of observational noise distributions. This enhancement significantly boosts the adaptability of models to complex non-Gaussian environments, making it a prominent trend in recent trajectory estimation model designs.

2.4 Bayesian Neural Network Methods

Bayesian Neural Networks (BNNs), which integrate the strengths of Bayesian inference and deep learning, have demonstrated powerful capabilities in causal modeling and handling complex uncertainties. In recent years, they have been widely applied in tasks such as target tracking and state estimation. By constructing probabilistic relationships among variables, BNNs provide effective modeling of uncertainty in dynamic systems, making them particularly suitable for state estimation scenarios involving incomplete information or complex observational noise. Unlike traditional neural networks, BNNs assume probabilistic distributions over model parameters—such as weights and biases—allowing them to not only learn feature mappings during training but also quantify uncertainty in predictions. Structurally, BNNs maintain the classic architecture of input, hidden, and output layers, but their parameters are updated from prior to posterior distributions through variational inference, enhancing both generalization and robustness. During inference, BNNs sample multiple times from the posterior distribution and average the results, which effectively suppresses overfitting and improves estimation accuracy under small-sample conditions.

BNNs have found broad application in various tasks including target tracking, traffic flow prediction, natural language processing, and risk analysis. For example, Sun et al. [41] proposed modeling traffic data using a Bayesian network based on Gaussian Mixture Models and employed a minimum mean square error criterion for accurate traffic flow estimation. Kong et al. [42] designed a Bayesian network that integrates Gated Recurrent Units (GRUs) with variational inference, achieving notable performance in traffic prediction tasks. Nghiem et al. [43] introduced a hybrid C-LSTM model (CNN-Bayes LSTM), which incorporates Bayesian reasoning into neural networks to alleviate overfitting in low-data regimes and improve uncertainty estimation in predictions.

In the domains of trajectory estimation and path planning, Nayak et al. [44] utilized Bayesian approximation methods to model uncertainty in neural networks, supporting safer trajectory prediction for autonomous driving systems. Li et al. [45] combined Bayesian networks with graph neural networks to construct a composite model for mobile target trajectory estimation, improving predictive robustness in dynamic environments. Posch et al. [46] further introduced variational inference mechanisms into deep networks to approximate high-dimensional posterior distributions, addressing challenges in uncertainty modeling within deep learning. In risk assessment and system modeling, Li et al. [47] applied maritime accident data to Bayesian network structures, demonstrating the adaptability and reasoning capability of Bayesian models in maritime risk prediction.

Collectively, these studies highlight that by incorporating uncertainty modeling, Bayesian neural networks exhibit enhanced flexibility and reliability in complex system state estimation and trajectory prediction tasks. Overall, the integration of Bayesian methods with neural networks has become a key development direction in the design of trajectory learning and estimation models.

2.5 Comparative Summary of Existing Methods

Compared with the aforementioned methods, existing approaches each exhibit limitations when confronted with strongly nonlinear dynamics and complex, non-Gaussian measurement noise. Classical Bayesian filters such as EKF, UKF, and CKF rely on first- or second-order approximations and assume Gaussian uncertainties, which restrict their robustness under real-world disturbances and colored sensor noise. Deep-learning-based estimators relax the structural assumptions of Kalman filtering, yet they often decouple process modeling from measurement modeling, leading to inconsistency in their probabilistic formulation. Recent learned Kalman variants such as NNAKF and KalmaNet partially address this issue through learnable fusion mechanisms, but their process models remain deterministic and their measurement likelihoods typically follow unimodal Gaussian forms, making it difficult to capture heavy-tailed or multimodal noise distributions.

In contrast, the proposed VNNPF provides a unified probabilistic framework that jointly incorporates a Bayesian GRU for modeling nonlinear process dynamics together with transition uncertainty, a planar-flow-augmented VAE for learning expressive non-Gaussian measurement likelihoods, and a learnable fusion gain optimized through Bayesian optimization to adaptively adjust state updates under varying noise conditions. This combination enables the model to handle nonlinear, non-Gaussian, and trajectory-dependent uncertainties in an end-to-end manner, achieving a level of modeling capacity that is not simultaneously addressed by existing classical filters, conventional deep models, or recent learned Kalman filters.

3 Methodology

3.1 Bayesian Filtering and Hyperparameter Optimization

Bayesian filtering is a probabilistic inference method used to estimate the posterior distribution of a system's state based on prior knowledge and observed data. It operates as a recursive algorithm that addresses the problem of state estimation under uncertainty, particularly excelling in scenarios where sensor measurements are corrupted by significant colored noise.

In the context of target motion tracking, the movement process can be regarded as a stochastic process. The core idea of Bayesian filtering is to represent the system state as a probability distribution rather than a deterministic value. This probabilistic representation reflects the uncertainty or confidence associated with the system state. The algorithm utilizes Bayes' theorem to update the state estimates of the stochastic process, continuously integrating the prior distribution with incoming observations to incrementally refine the posterior distribution. Bayes' theorem expresses the relationship for computing the posterior probability given observational data. For a system state variable x and an observation variable z, Bayes' theorem is given by the following formula:

$$p(x|z) = \frac{p(z|x) \cdot p(x)}{p(z)} \tag{1}$$

where $p(x|z)$ is the posterior probability of the state variable x given the observation z, $p(z|x)$ is the likelihood (the probability of observing z given state x), $p(x)$ is the prior probability of the state variable x, and $p(z)$ is the evidence (the marginal probability of observing z).

Bayesian filtering incrementally improves the estimation of the system state in a stochastic process by continuously updating the posterior probability distribution. The general steps of the algorithm are as follows:

1. Initialization: Initialize the prior probability of the system state based on available prior knowledge.
2. Prediction: Using the system's dynamic model, predict the state distribution at the next time step based on the previous state estimate and control input (if applicable).
3. Update: Incorporate new observational data to update the predicted state distribution into a posterior distribution using Bayes' theorem. This step accounts for the reliability of the observations, ensuring that more reliable measurements have a greater influence on the updated estimate.
4. Resampling: To avoid filter degradation (such as sample impoverishment), a resampling step may be applied after the update to maintain diversity in the state distribution.
5. Iteration: Repeat steps 2 through 4 recursively to estimate the system state over time.

The structure of the Bayesian filtering algorithm is illustrated in Figure 1.

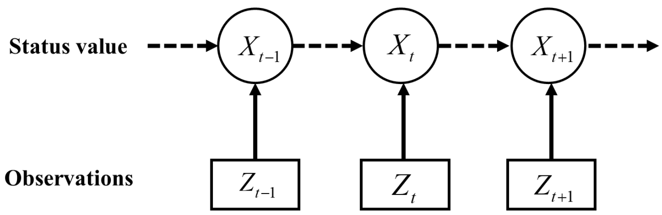

Figure 1. Structure of the bayesian filtering algorithm.

In Figure 1, X_t and Z_t represent the state and observation at time t, respectively. The observation Z_t is incorporated with the state X_t to compute the estimated state at the next time step.

The fundamental principle underlying both the Kalman Filter and Particle Filter is Bayesian filtering. The Kalman Filter is also a recursive estimation algorithm that continuously refines the system state by incorporating new observations. A key concept in Kalman filtering is the introduction of the filter gain, which serves to balance prior knowledge with observational data. By applying the filter gain, the algorithm optimally fuses two uncertain quantities to produce an estimate that is closer to the true value. Inspired by this mechanism, our variational neural network-based estimation framework also adopts the one-step update principle of Bayesian filtering and the filter gain mechanism of the Kalman Filter. The neural network is designed to learn this process and output a state estimate with reduced error.

The Kalman gain is one of the most critical parameters in Kalman filtering. It determines how measurement data and predicted states are combined. The magnitude of the gain adjusts the relative influence of the measurement and the prediction, ultimately resulting in a more accurate state estimate.

The classical Kalman filter gain is computed using the following formula:

$$K(t) = P_{t|t-1}C^T \left[C^T P_{t|t-1}C + R\right]^{-1} \tag{2}$$

However, in this study, both the process function $f(\cdot)$ and the measurement function $h(\cdot)$ are implemented by neural networks, while the covariance terms P

and R are implicitly learned from data. Hence the matrices P and C in the classical gain formula are no longer available, making the analytic computation of $K(t)$ infeasible. To retain the idea of balancing prior predictions and new observations, we treat the fusion gain as a learnable parameter that is optimized via Bayesian hyperparameter optimization. This approach allows the model to adaptively determine how much weight should be given to the measurement update relative to the state prediction, without relying on explicit covariance matrices.

Before training the neural network, it is necessary to set a set of hyperparameters to ensure optimal estimation performance. Traditional approaches based on manual tuning or exhaustive search are time-consuming and often fail to achieve the best results. Therefore, optimizing the hyperparameters is essential. Bayesian optimization leverages prior information to improve search efficiency by iteratively predicting posterior knowledge points based on current observations. Formally, the hyperparameter optimization problem can be defined as follows (Equation 3):

$$K^* = \arg\min_{k \in K} \operatorname{loss}(k) \tag{3}$$

Assume there exists a set of hyperparameters K. For each hyperparameter configuration k_i, the function to be optimized loss() can be evaluated on the objective function loss() to compute the minimum value, thus obtaining the optimal hyperparameter K^*. The value of this optimal hyperparameter corresponds to the concept of the classical filter gain in Equation 2, which is used to fuse measurement data and state estimates.

The objective function to be minimized is defined as the root mean square error (RMSE):

$$f(k_j) = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i(k_j) - y_i)^2}{n}} \tag{4}$$

where k_j denotes the set of all hyperparameters (including the fusion gain), y_i represents the ground truth, and $\hat{y}_i(k_j)$ is the model output under the j-th hyperparameter configuration. In Bayesian optimization terminology, $f(k_j)$ is called the "objective function".

The optimization procedure maintains a dataset $D = \{(k_1, f(k_1)), (k_2, f(k_2)), \cdots, (k_i, f(k_i))\}$ of previously evaluated configurations. Based on D, a Gaussian process (GP) surrogate model is built to approximate the unknown function $f(k)$. The GP provides a posterior predictive distribution $p(f|k, D) \sim \mathcal{N}(\mu(k), \sigma^2(k))$, where $\mu(k)$ and $\sigma(k)$ are the predictive mean and standard deviation at point k.

An acquisition function $u(k)$ is then used to decide where to sample next. We employ the Expected Improvement (EI) criterion:

$$u(k) = \mathbb{E}[\max(f_{\min} - Y, 0)], \quad Y \sim \mathcal{N}(\mu(k), \sigma^2(k)) \tag{5}$$

where $f_{\min} = \min\{f(k_1), f(k_2), \dots\}$ is the best observed value so far. The next hyperparameter configuration is selected by maximizing the acquisition function:

$$k_{i+1} = \arg\max_{k \in \mathcal{K}} u(k) \tag{6}$$

In this study, the posterior model $p(L|D)$ in the Bayesian optimization framework is implemented through a Gaussian process regression surrogate, which provides the predictive mean and variance required for evaluating the acquisition function. The acquisition strategy adopts the Expected Improvement (EI) criterion, so that each new hyperparameter candidate k is selected based on its expected improvement over the current best loss value. By iteratively sampling k according to $u(k)$, the Bayesian optimization process efficiently explores the gain space while avoiding unnecessary evaluations.

We employ Bayesian hyperparameter optimization to optimize the filter gain K. Unlike the Kalman filter gain, the filter gain in our method is optimized using historical information from a preceding time period to obtain a single gain value K. It is not necessary to optimize a new gain at every estimation step. Instead, Bayesian hyperparameter optimization is re-applied only when a change occurs in the motion state of the target. This approach reduces the computational burden on the network without compromising the accuracy of the results.

3.2 Variational Neural Network Model Based on Kalman Filter Structure

Assume a nonlinear system model is given as follows:

$$x(t) = f[x(t-1), w(t-1)] \tag{7}$$

$$z(t) = h[x(t), v(t)] \tag{8}$$

where $x(t)$ is the system state to be estimated, $w(t-1)$ is the process noise, $z(t)$ is the system measurement, and $v(t)$ is the measurement noise.

Equation (7) represents the process model of the system, describing how the estimated state evolves over time. $f(\cdot)$ is the nonlinear process function, and $w(t)$ denotes the process noise, which reflects the uncertainty from $x(t-1)$ to $x(t)$ in the process. Equation (8) corresponds to the measurement model of the system, where $h(\cdot)$ is the nonlinear measurement function, and $v(t)$ represents the measurement noise.

For state estimation, the stochastic process can be regarded as a complex nonlinear system. Such systems exhibit intricate physical evolution characteristics and colored noise distributions, requiring consideration of system diversity and nonlinear effects. In nonlinear systems, the relationship between inputs and outputs does not follow a simple proportional rule but instead involves complex nonlinear dynamics. Moreover, these systems typically involve a large number of parameters, making parameter estimation particularly challenging, especially in high-dimensional scenarios. Additionally, the noise in nonlinear systems often deviates from a Gaussian distribution, leading to more complex statistical characteristics. To address these challenges, we propose a variational neural network architecture designed to model complex nonlinear systems. This network not only captures the nonlinear dynamics but also describes the non-Gaussian noise distributions present in the system. The overall network structure is illustrated in Figure 2:

The proposed model framework is referred to as the Variational Neural Network Model Based on Kalman Filter Structure, as shown in Figure 2. It is inspired by the principles of Bayesian estimation and the structural design of the Kalman filter. In this framework, a Bayesian Gated Recurrent Unit (BGRU) is employed as the state network within the estimator, while a Planar Flow Variational Autoencoder (PFVAE) serves as the measurement network. The filter gain K is determined using Bayesian hyperparameter optimization.

First, the target's positional information at time t (denoted as X_t) is input into the state network based on the Bayesian gated recurrent unit. In the variational Bayesian GRU, the weights and biases are treated as trainable parameter distributions. By sampling these distributions multiple times, different sets of weights and biases are obtained, leading to different state estimates. The mean of these estimates is then computed to produce the state estimation mean at time $t+1$ (denoted as $\hat{X}_{t+1}$), which is output by the network. Next, $\hat{X}_{t+1}$ is passed into the measurement network based on a planar flow variational autoencoder. An LSTM-based encoder extracts the hidden features of the input data. Through a variational inference layer, the encoder's output is mapped to parameters of a Gaussian distribution with zero mean and unit variance. A simple distribution is then passed through the planar flow layer to generate a more complex posterior distribution over the latent variables. Samples drawn from this complex distribution are decoded by the decoder to reconstruct the input, thereby producing an accurate measurement result (denoted as Z_{t+1}).

Within the Bayesian hyperparameter optimization module, the gain parameter K is set. Following the form of the Kalman filter equations, an optimization problem is constructed, and the optimal gain K is determined using Bayesian optimization, yielding the predicted result at the next time step (denoted as X_{t+1}) according to the Kalman prediction step.

The model is trained on the training dataset in a stepwise manner. First, the state network is trained independently to learn the temporal evolution of the latent state from historical trajectories. After the state network has converged, its parameters are fixed, and the measurement network is trained to reconstruct the observations and to capture the statistical characteristics of the measurement noise. Once both subnetworks have been trained, the fusion gain K is determined by Bayesian optimization on a separate validation set, ensuring that the overall estimation framework achieves a stable balance between process prediction and measurement correction. During the testing phase, the trained state network, the trained measurement network, and the optimized gain are combined to perform trajectory estimation on the test data.

3.3 Process and Measurement Network

A Bayesian Gated Recurrent Unit (BGRU) is a variant of the traditional GRU in which the weights and biases are modeled as probability distributions and optimized through variational inference, following the ideas of Bayesian neural networks and variational recurrent models proposed by Blundell et al. [48] and Chung et al. [49].

The BGRU shares the same chain structure and gating mechanisms as the standard GRU. The key difference lies in that, within the BGRU, the weights and biases of the update and reset gates are modeled as trainable parameter distributions rather than fixed values. During training, all weights and biases in the network are initialized as standard normal distributions, with independence assumed between weights and biases. These distributions are then refined through

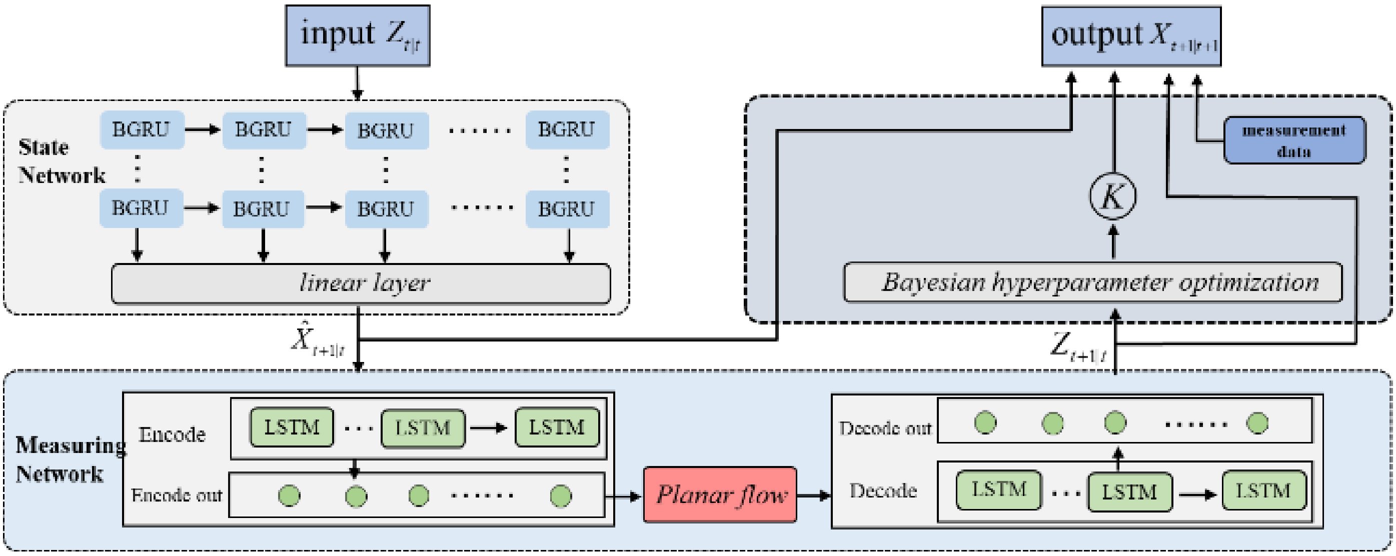

Figure 2. Estimation model based on variational neural network.

backpropagation by continuously adjusting their means and standard deviations, ultimately yielding optimal values. In the prediction phase, multiple samples are drawn from the learned distributions of weights and biases. The mean of the resulting predictions is computed to obtain the final output. The network architecture of the BGRU is illustrated in Figure 3.

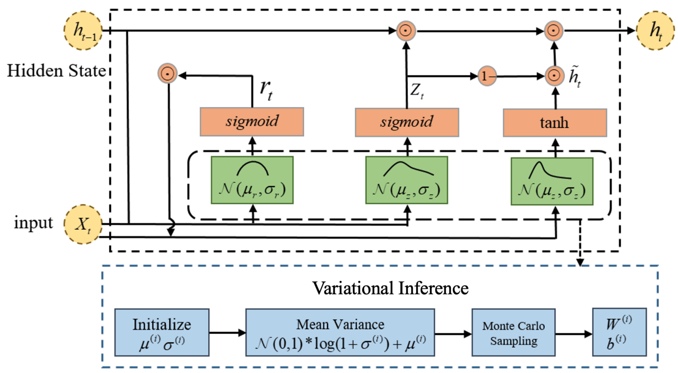

Figure 3. Bayesian gated recurrent unit network architecture.

In traditional models, biases and weights are treated as fixed numerical values. In the BGRU, however, W_z and b_z are no longer specific values but rather samples drawn from a Gaussian distribution with mean μ_z and covariance σ_z. Taking W_r and b_r as examples, let $w_n^{(i)}$ represent the sampled weight at the n-th instance of the i-th layer, and $b_n^{(i)}$ represent the bias. Both are assumed to follow a Gaussian distribution. To ensure the non-negativity of the variational parameters and the differentiability of the standard deviation, a reparameterization trick is applied:

Let $\mathbf{W}^{(i)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and $\mathbf{b}^{(i)} \in \mathbb{R}^{d_{\text{out}}}$ denote the weight matrix and bias vector of the i-th layer. Their variational posterior is assumed to be a factorized Gaussian:

$$\begin{aligned} \mathbf{W}^{(i)} &= \boldsymbol{\mu}_W^{(i)} + \boldsymbol{\sigma}_W^{(i)} \odot \boldsymbol{\epsilon}_W, \quad \boldsymbol{\epsilon}_W \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathbf{b}^{(i)} &= \boldsymbol{\mu}_b^{(i)} + \boldsymbol{\sigma}_b^{(i)} \odot \boldsymbol{\epsilon}_b, \quad \boldsymbol{\epsilon}_b \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \end{aligned} \tag{9}$$

where $\boldsymbol{\mu}_W^{(i)}, \boldsymbol{\sigma}_W^{(i)}$ and $\boldsymbol{\mu}_b^{(i)}, \boldsymbol{\sigma}_b^{(i)}$ are trainable variational parameters, and $\odot$ denotes element-wise multiplication. The same reparameterization is applied to all weight and bias variables in the BGRU gates.

Based on the derivation of variational inference, the evidence lower bound (ELBO) loss function of the network is defined as:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(w|\theta)}[\log p(y|w, x)] - D_{\text{KL}}(q(w|\theta) \parallel p(w)) \tag{10}$$

where $p(w)$ is a user-defined prior distribution (typically a standard Gaussian), and $q(w|\theta)$ is the variational posterior parameterized by $\theta = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$.

The first term in (10) is the expected log-likelihood. For regression with a Gaussian observation model, it reduces to a mean-squared-error (MSE) term up to a constant factor. Therefore, in practice we use the following weighted loss function that balances data fitting and regularization:

$$\mathcal{L}_{\text{BNN}} = \alpha \cdot \text{MSE}(\hat{y}, y) + \beta \cdot D_{\text{KL}}(q(w|\theta) \parallel p(w)) \tag{11}$$

where $\hat{y}$ is the network output, α and β are positive scaling coefficients, and the KL divergence term encourages the variational posterior to stay close to the prior, thereby controlling model complexity.

3.4 Measurement Network

3.4.1 Variational Autoencoder

The Variational Autoencoder (VAE) is an unsupervised neural network designed to handle complex nonlinear data. A VAE learns the latent distribution of data by mapping the input to a latent variable in a lower-dimensional latent space via an encoder, and then reconstructing data samples similar to the input by mapping the latent variable back to the original data space through a decoder.

The basic components and workflow of a VAE are as follows:

1. **Encoder**: The encoder maps the input data $\mathbf{x}$ to parameters of a latent distribution. Typically composed of neural networks, the encoder outputs a mean vector $\boldsymbol{\mu}$ and a log-variance vector $\log \boldsymbol{\sigma}^2$ for the latent variable $\mathbf{z}$.

2. **Reparameterization Trick**: To enable backpropagation through stochastic sampling, we use:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

   where $\odot$ denotes element-wise multiplication.

3. **Planar Flow Transformation** (for PFVAE): To enhance the expressiveness of the variational posterior, we apply planar flow transformations:

$$\mathbf{z}^{(k)} = \mathbf{z}^{(k-1)} + \mathbf{u}_k \cdot \tanh(\mathbf{w}_k^\top \mathbf{z}^{(k-1)} + b_k)$$

   where $\mathbf{u}_k, \mathbf{w}_k \in \mathbb{R}^D$, $b_k \in \mathbb{R}$ are learnable parameters, and $\mathbf{z}^{(0)} = \mathbf{z}$.

4. **Decoder**: The decoder maps the transformed latent variable $\mathbf{z}^{(K)}$ back to the data space to reconstruct $\hat{\mathbf{x}}$.

5. **Loss Function**: The PFVAE loss consists of reconstruction error and KL divergence:

$$\mathcal{L}_{\text{PFVAE}} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))$$

   where $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the prior. In practice, for Gaussian decoder:

$$\mathcal{L}_{\text{PFVAE}} = \frac{1}{2\sigma^2}\|\mathbf{x}-\hat{\mathbf{x}}\|^2 - \frac{1}{2}\sum_{j=1}^{J}(1+\log \sigma_j^2 - \mu_j^2 - \sigma_j^2)$$

3.4.2 Planar Flow

Although Variational Autoencoders (VAEs) have many advantages, they typically adopt a Gaussian prior distribution, which makes it difficult to model complex distribution characteristics commonly found in real-world data. The VAE's ability to learn true distributions is limited, which may result in a significant deviation between the approximate posterior obtained through variational inference and the actual distribution. Normalizing flows offer a flexible and learnable method for constructing probability distributions, and planar flow is a widely used variant within this framework. Combining VAE with planar flow can effectively address the limitation of Gaussian priors in modeling complex posteriors in real data.

In normalizing flows, planar flow provides a transformation with tractable Jacobian determinant. Given an invertible transformation $f : \mathbb{R}^D \to \mathbb{R}^D$, the density after transformation is:

$$q_K(\mathbf{z}_K) = q_0(\mathbf{z}_0) \prod_{k=1}^{K} \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right|^{-1}$$

where $\mathbf{z}_k = f_k(\mathbf{z}_{k-1})$.

For planar flow defined as:

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u}\, h(\mathbf{w}^\top \mathbf{z} + b) \tag{12}$$

where $\mathbf{u}, \mathbf{w} \in \mathbb{R}^D$, $b \in \mathbb{R}$, and $h(\cdot)$ is a smooth activation function (typically $\tanh$). The Jacobian matrix of this transformation is:

$$\frac{\partial f}{\partial \mathbf{z}} = \mathbf{I} + \mathbf{u}\, \psi(\mathbf{z})^\top, \quad \psi(\mathbf{z}) = h'(\mathbf{w}^\top \mathbf{z} + b)\, \mathbf{w}.$$

Applying the matrix determinant lemma, the determinant can be computed efficiently as a scalar:

$$\det \frac{\partial f}{\partial \mathbf{z}} = 1 + \mathbf{u}^\top \psi(\mathbf{z}) \tag{13}$$

which is valid because $\mathbf{u}\psi(\mathbf{z})^\top$ is a rank-one matrix.

For a sequence of K planar-flow transformations, the final latent variable is $\mathbf{z}_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{z}_0)$. Using the change-of-variables formula for normalizing flows, the log-density of $\mathbf{z}_K$ is:

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right| \tag{14}$$

Substituting the determinant expression from (13) yields:

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \ln \left| 1 + \mathbf{u}_k^\top \psi_k(\mathbf{z}_{k-1}) \right| \tag{15}$$

This formulation allows for efficient computation and optimization while providing flexible posterior distributions.

Figure 4 illustrates how a planar flow transforms a simple Gaussian distribution into a more complex one. As shown, the first plot q_0 represents a standard Gaussian distribution. Through multiple transformations by the planar flow, the distribution evolves into a more complex form, as shown in q_2, enabling the model to fit and represent complex distributions that more accurately reflect real-world data.

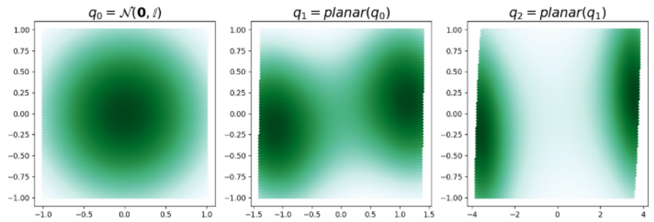

Figure 4. Distribution transformation of planar flow.

3.4.3 *Variational Autoencoder Based on Planar Flow*

A Planar Flow Variational Autoencoder (PFVAE) extends the standard VAE by incorporating planar normalizing flow layers after the encoder so that a simple Gaussian latent distribution can be transformed into a more expressive non-Gaussian posterior, following the normalizing flow framework introduced by Rezende et al. [33].

By integrating planar flow into the variational autoencoder, the PFVAE is constructed. The structure of the PFVAE is shown in Figure 5:

In Fig. 5, $\mathbf{w} = [w_1, w_2, \cdots, w_i]$, $\mathbf{u} = [u_1, u_2, \cdots, u_i]$, and $\mathbf{b} = [b_1, b_2, \cdots, b_i]$ represent the weight, scale, and bias parameters of the planar flow, respectively, while $\mathbf{z} = [z_1, z_2, \cdots, z_i]$ denotes the new latent variable obtained through the planar flow transformation. When planar flow is applied, the evidence lower bound (ELBO) for the initial latent distribution $q_0(\mathbf{z}|\mathbf{x})$ becomes:

$$\begin{aligned}\mathrm{ELBO} &= \mathbb{E}_{q_0(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x},\mathbf{z}) - \log q_0(\mathbf{z}|\mathbf{x})] \\ &= \mathbb{E}_{q_0(\mathbf{z}_0|\mathbf{x})}[\log p(\mathbf{x},\mathbf{z}_k) - \log q_k(\mathbf{z}_k|\mathbf{x})] \\ &= \mathbb{E}_{q_0(\mathbf{z}_0|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z}_k)] - D_{\mathrm{KL}}(q_0(\mathbf{z}_0|\mathbf{x}) \parallel p(\mathbf{z}_0)) \\ &\quad - \mathbb{E}_{q_0(\mathbf{z}_0|\mathbf{x})}\left[\sum_{i=1}^{k} \log \left|\det \frac{\partial f_i}{\partial \mathbf{z}_{i-1}}\right|\right]\end{aligned} \tag{16}$$

For variational inference, we use deep neural networks to construct a mapping from the observation $\mathbf{x}$ to the parameters of the initial density $q_0 = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$, as well as the flow parameters $\boldsymbol{\lambda}$.

The optimization objective can be written as:

$$\begin{aligned}\mathrm{ELBO}(\mathbf{x}) = \mathbb{E}_{q_0(\mathbf{z}_0|\mathbf{x})}&[\log p(\mathbf{x}|\mathbf{z}_k)] \\ &- D_{\mathrm{KL}}(q_0(\mathbf{z}_0|\mathbf{x}) \parallel p(\mathbf{z}_0)) \\ &- \mathbb{E}_{q_0(\mathbf{z}_0|\mathbf{x})}\left[\sum_{i=1}^{k} \log \left|\det \frac{\partial f_i}{\partial \mathbf{z}_{i-1}}\right|\right]\end{aligned} \tag{17}$$

This objective corresponds to a complete evidence lower bound (ELBO): the reconstruction term measures how well the observation can be recovered from the latent variable, the KL divergence regularizes the initial latent density toward the prior, and the log-determinant Jacobian terms account for the transformations introduced by planar flows. Together, these terms form the loss function used to train the PFVAE-based measurement network.

This optimization objective can be optimized using the stochastic gradient descent method. Since q_0 is a standard normal distribution function, we can sample from this distribution. The final latent variable z_i will be sampled from a complex posterior distribution for reconstruction.

In the variational inference of VAE, planar flow is used as a distribution transformer. By applying the given planar flow to the latent posterior distribution sampled from the encoder layer of the VAE, the posterior is made more complex. This enables the VAE prediction model to learn and process data with more complex characteristics, thereby enhancing the model's nonlinearity and dynamic adaptability.

4 IMU-GPS Fusion Localization Experiment

IMU (Inertial Measurement Unit) refers to a device that includes a gyroscope, accelerometer, and magnetometer. In this experiment, IMU data is incorporated in addition to GPS data to achieve fused localization. The network takes multi-feature inputs composed of both IMU and GPS data and outputs displacement estimates.

To illustrate the fusion mechanism, the IMU data provide high-frequency inertial predictions describing short-term vehicle motion, whereas the GPS measurements supply low-frequency but globally referenced position updates. In our framework, the IMU signals are fed into the BGRU-based process

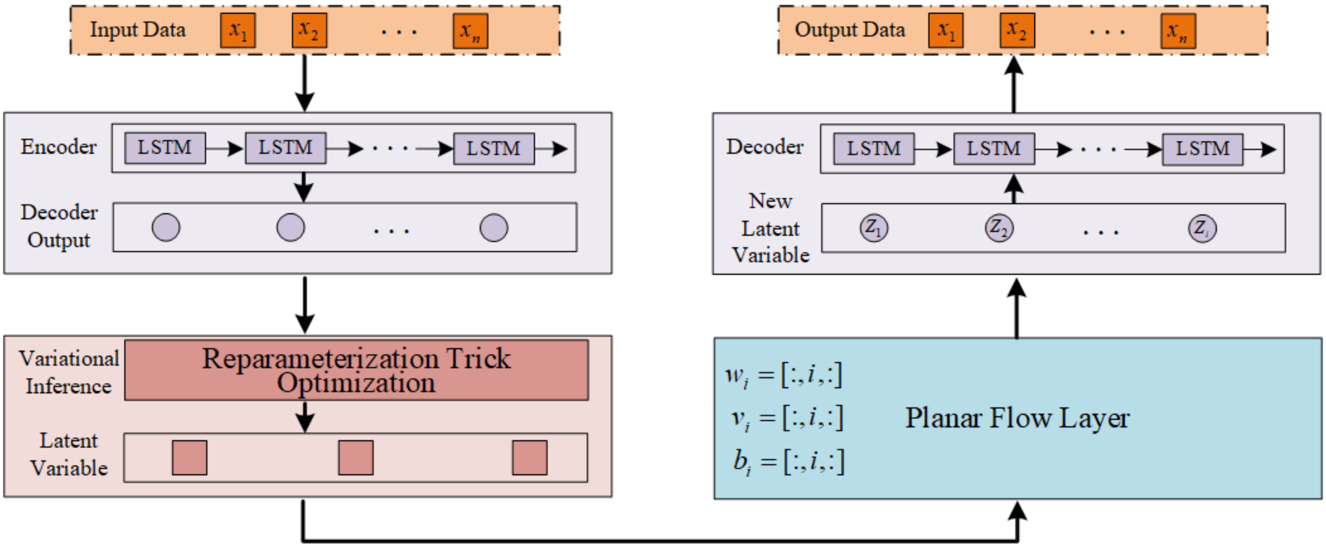

Figure 5. Architecture of the planar flow-based variational autoencoder.

model, while the GPS observations are modeled through the PF-VAE measurement module. During data fusion, two practical challenges arise: (1) the IMU and GPS sensors operate at different sampling rates, leading to asynchronous observations, and (2) GPS measurements occasionally exhibit large deviations caused by multi-path interference or urban-canyon effects. The proposed VNNPF framework mitigates these issues by using the Bayesian GRU to stabilize high-rate IMU predictions and the PF-VAE to learn non-Gaussian GPS noise distributions, allowing robust fusion under real-world sensing conditions.

The dataset used in this experiment is the KITTI benchmark, jointly developed by the Karlsruhe Institute of Technology in Germany and the Toyota Technological Institute at Chicago. It is currently one of the largest and most widely used benchmark datasets for trajectory estimation algorithms in autonomous driving scenarios. The selected dataset's trajectory plot is shown in Figure 6. The experiments were conducted on 9 trajectories.

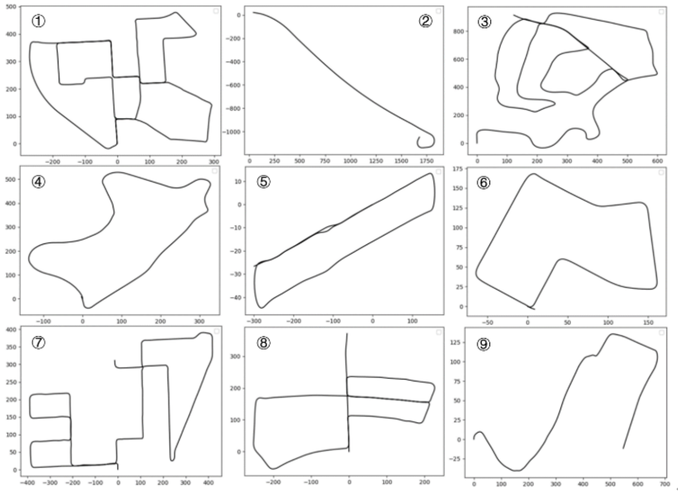

Figure 6. Ground truth trajectories selected from the dataset.

The proposed VNNPF model was compared with several mainstream neural network baselines. Taking the first trajectory as an example, the performance metrics are shown in Table 1. Figures 7, 8, 9, 10 and 11 show a comparison of the RMSE, MAE, MAPE, SMAPE, and R values for trajectories 1–9. Different trajectory numbers are indicated by ①–⑨ in the figure.

Table 1. Comparison of performance metrics between VNNPF and different neural network models (Trajectory No.1).

Model	RMSE	MAE	MAPE	SMAPE	R
GRU	13.112	8.183	14.435	8.693	0.929
LSTM	27.773	17.294	20.361	11.654	0.867
BGRU	35.566	22.411	31.652	12.636	0.814
BLSTM	30.724	19.232	22.174	12.091	0.856
CNN-LSTM	12.311	8.856	11.566	10.293	0.891
Conv-LSTM	16.148	13.714	7.111	8.137	0.872
VAE	6.679	5.411	24.062	7.498	0.955
IAFVAE	6.922	5.542	15.548	7.725	0.959
VNNPF	3.384	2.547	6.077	4.651	0.998

In the experiment, for Trajectory No. 1, the VNNPF model achieved the best performance, with an RMSE of 3.384, MAE of 2.547, MAPE of 6.077, SMAPE of 4.651, and R of 0.998. Among the compared neural network models, VAE achieved the best performance apart from VNNPF. Notably, compared with the BGRU model (the core process component), VNNPF achieved substantially lower errors across all metrics

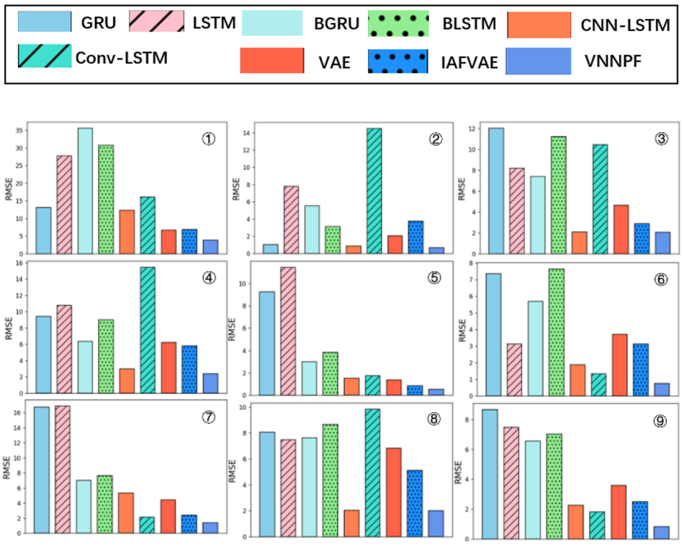

Figure 7. Bar chart of RMSE for different neural network models.

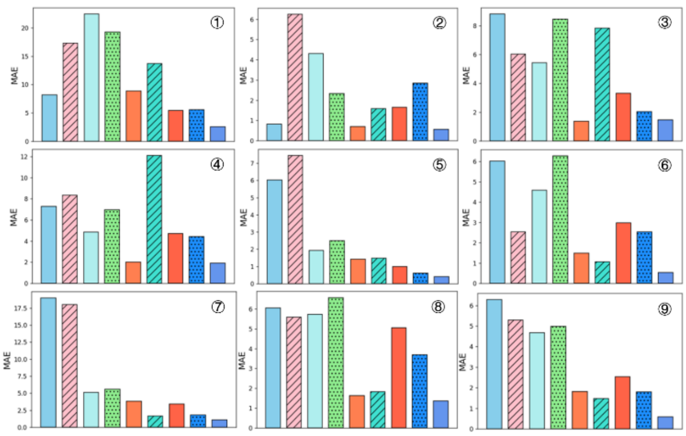

Figure 8. Bar chart of MAE for different neural network models.

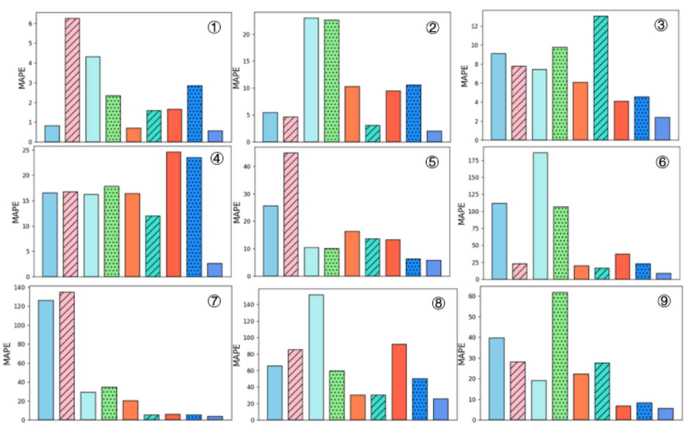

Figure 9. Bar chart of MAPE for different neural network models.

(reductions exceeding 63% for RMSE, MAE, MAPE, and SMAPE) and a significantly higher R value.

To provide a comprehensive visual comparison across all evaluation metrics simultaneously, Figure 12

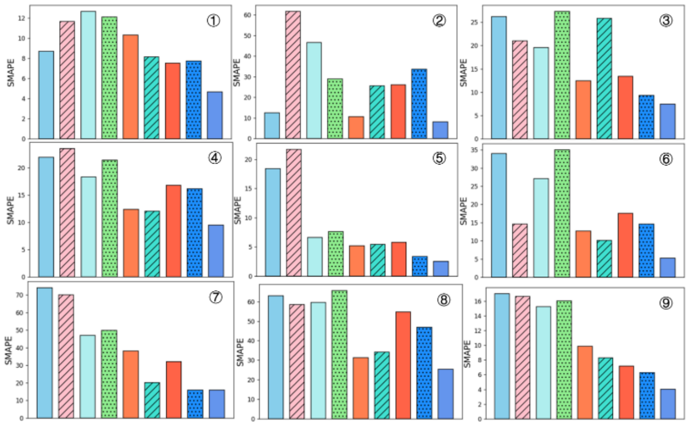

Figure 10. Bar chart of SMAPE for different neural network models.

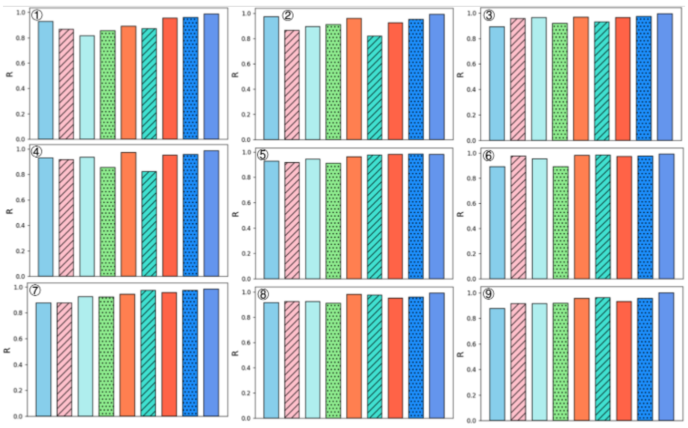

Figure 11. Bar chart of R-squared (R) for different neural network models.

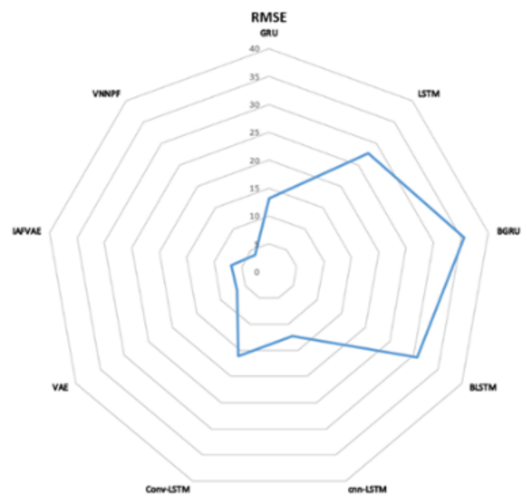

Figure 12. Radar charts of performance metrics for different models for Trajectory No.1.

presents radar charts for Trajectory No.1. The radar charts clearly illustrate the overall superiority of VNNPF, as it exhibits the largest area coverage across all five metrics. In particular, VNNPF dominates in the directions of RMSE, MAE, MAPE, and SMAPE (where smaller values indicate better performance), while also achieving the highest R value. The balanced performance profile of VNNPF is evident from its nearly symmetrical shape in the radar chart, indicating consistent improvements across all evaluation dimensions. In contrast, other models show irregular shapes with clear weaknesses in specific metrics; for instance, BGRU has poor performance in MAPE, while VAE shows relatively weaker performance in R value.

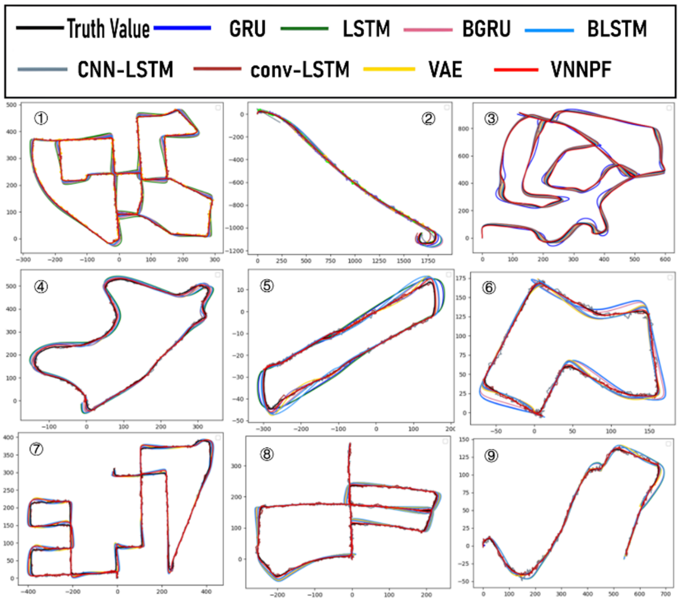

Figure 13. Trajectory results of different models.

From the above experiments, it is evident that the proposed model demonstrates a significant improvement in tracking accuracy compared to other neural network models after incorporating IMU data. The trajectory fitting results shown in Figure 13 clearly illustrate the effectiveness of the proposed model. Based on both the evaluation metrics and the trajectory tracking plots, the method proposed in this paper is applicable to multi-feature trajectory datasets and achieves optimal performance, thereby validating the effectiveness of the approach.

Compared with traditional Kalman filtering methods such as EKF and UKF, the superior performance of the proposed VNNPF model can be attributed to its ability to handle nonlinear dynamics and non-Gaussian sensor noise. Classical filters rely on predefined motion and measurement models and typically assume Gaussian noise distributions. As a result, their performance degrades when the real-world IMU and GPS data contain strong nonlinearities, time-varying noise, or occasional large deviations. In contrast, the BGRU-based process model learns complex temporal dependencies directly from data, reducing the accumulation of inertial drift. Meanwhile, the PF-VAE measurement model captures heavy-tailed and asymmetric GPS noise by learning flexible likelihood distributions through normalizing flows. Furthermore, the Kalman-inspired learnable fusion gain enables adaptive fusion between inertial predictions and GPS observations, overcoming the limitations of fixed analytic gains. These components collectively allow the VNNPF to achieve more accurate and robust state estimation in challenging real-world sensing environments.

In addition to the above experimental evaluation, we also provide a brief discussion of the computational characteristics of the proposed method. The main computational load comes from the Bayesian GRU used for process modeling and the PF-VAE used for measurement modeling. The BGRU requires sequential recurrent updates, while the PF-VAE involves an encoder–decoder pass together with several planar-flow transformations. The Kalman-inspired gain update introduces only lightweight matrix operations and adds negligible overhead. In practice, all components run efficiently on a single GPU, and the overall inference speed on the KITTI dataset satisfies real-time requirements, indicating that the proposed framework is suitable for large-scale or latency-sensitive trajectory estimation tasks.

To ensure consistent comparison within a unified modeling paradigm, the experiments in this study focus on deep learning–based estimation methods. The proposed VNNPF is designed as a fully data-driven variational architecture without relying on explicit analytical motion equations or predefined noise models. Therefore, learning-based methods that similarly operate in an end-to-end data-driven manner provide the most appropriate and directly comparable baselines. The selected models cover recurrent structures, convolutional–recurrent hybrids, variational formulations, and flow-enhanced variants, forming a representative set for evaluating learning capability, noise-modeling flexibility, and fusion performance. Within this experimental scope, the comparison adequately reflects the estimation characteristics and advantages of the proposed framework.

5 Conclusion

In recent years, with the rapid development of the Fourth Industrial Revolution, computational hardware has advanced significantly, and high-performance devices have become increasingly widespread. These advancements have accelerated the growth of big data technologies, artificial intelligence, sensor systems, and deep learning techniques. Today, data has emerged as a critical resource. Vast amounts of data are stored, containing patterns that reflect the evolution of complex systems. Extracting meaningful information and modeling system behaviors through data-driven methods is key to achieving efficient decision-making and risk control. Therefore, research on data-driven estimation methods holds substantial theoretical and practical significance.

Modeling the nonlinear motion of dynamic targets is inherently difficult, and the sensor-collected data is often affected by device limitations and environmental interference, resulting in substantial colored noise and in data that exhibit high complexity and nonlinearity. To address these challenges and enhance the model's ability to represent motion dynamics and manage noisy observations, this paper presents the following contributions:

1. To address the difficulty of modeling nonlinear motion patterns, this paper adopts the Bayesian Gated Recurrent Unit (BGRU) as the state transition model. By representing weights and biases as trainable normal distributions, the BGRU enhances nonlinear modeling capacity and improves robustness and estimation accuracy.

2. To address the presence of colored noise in observational data and the limitations of Gaussian assumptions, this paper introduces the Planar Flow-based Variational Autoencoder (PFVAE) into the measurement network. Planar flow transforms simple Gaussian distributions into more complex posterior distributions, allowing for a more accurate representation of real-world observation characteristics and nonlinear measurement processes.

3. To effectively fuse prior knowledge with posterior information, this paper draws inspiration from the Kalman filter and introduces a filtering gain. The gain is dynamically optimized through Bayesian hyperparameter optimization, which refines the estimation results.

4. By integrating the three components above, this paper proposes a Variational Neural Network Estimation Method based on the Kalman Filter Structure, comprising a state network (BGRU), a measurement network (PFVAE), and a Bayesian optimization module. The model forms a unified end-to-end estimation framework with strong nonlinear modeling ability, dynamic adaptability, and the capacity to handle colored noise.

5. The proposed model is validated through experiments on the KITTI dataset. Results demonstrate that the method outperforms baseline models such as GRU, LSTM, CNN-LSTM, and Conv-LSTM in terms of estimation accuracy and stability, exhibiting strong generalization ability. In scenarios involving IMU integration, the model maintains reliable performance even under GNSS signal loss, indicating improved robustness and stability.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 62173007, Grant 62203020, Grant 62473008, Grant 62433002, and Grant 62476014; in part by the Beijing Nova Program under Grant 20240484710; in part by the Project of Humanities and Social Sciences (Ministry of Education in China, MOC) under Grant 22YJCZH006; in part by the Beijing Scholars Program Grant 099; in part by the Project of ALL China Federation of Supply and Marketing Cooperatives under Grant 202407; in part by the Project of Beijing Municipal University Teacher Team Construction Support Plan under Grant BPHR20220104.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Zhou, D. H., & Frank, P. M. (2002). Fault diagnostics and fault tolerant control. *IEEE Transactions on Aerospace and Electronic Systems*, *34*(2), 420–427. [CrossRef]

[2] Mathiassen, K., Hanssen, L., & Hallingstad, O. (2010, September). A low cost navigation unit for positioning of personnel after loss of GPS position. In *2010 International Conference on Indoor Positioning and Indoor Navigation* (pp. 1–10). IEEE. [CrossRef]

[3] Liu, K. R. (2009). *Cooperative communications and networking*. Cambridge University Press.

[4] Shi, Q. (2023). Trajectory tracking control method of mobile robot based on AVRX operating system. *Journal of Computational Methods in Science and Engineering, 23*(1), 253–265. [CrossRef]

[5] Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE, 92*(3), 401–422. [CrossRef]

[6] Eltoukhy, M., Ahmad, M. O., & Swamy, M. N. S. (2020). An adaptive turn rate estimation for tracking a maneuvering target. *IEEE Access, 8*, 94176–94189. [CrossRef]

[7] Wang, L., & Zhou, G. (2021). Pseudo-spectrum based track-before-detect for weak maneuvering targets in range-Doppler plane. *IEEE Transactions on Vehicular Technology, 70*(4), 3043–3058. [CrossRef]

[8] Jia, S., Zhang, Y., & Wang, G. (2017). Highly maneuvering target tracking using multi-parameter fusion singer model. *Journal of Systems Engineering and Electronics, 28*(5), 841–850. [CrossRef]

[9] Zhenkai, X., Fanying, L., & Lei, Z. (2018). Study on Maneuvering Target On-axis Tracking Algorithm of Modified Current Statistical Model. In *MATEC Web of Conferences* (Vol. 160, p. 02008). EDP Sciences. [CrossRef]

[10] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering, 82*(1), 35–45. [CrossRef]

[11] Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation: Theory algorithms and software*. John Wiley & Sons.

[12] Julier, S. J., & Uhlmann, J. K. (1997, July). New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI* (Vol. 3068, pp. 182–193). SPIE. [CrossRef]

[13] Wu, Y., Hu, D., Wu, M., & Hu, X. (2005, June). Unscented Kalman filtering for additive noise case: Augmented vs. non-augmented. In *Proceedings of the 2005 American Control Conference* (pp. 4051–4055). IEEE. [CrossRef]

[14] Chen, Y., Xie, X., Yu, B., Li, Y., & Lin, K. (2021). Multitarget vehicle tracking and motion state estimation using a novel driving environment perception system of intelligent vehicles. *Journal of Advanced Transportation, 2021*(1), 6251399. [CrossRef]

[15] Kulikov, G. Y., & Kulikova, M. V. (2015). The accurate continuous-discrete extended Kalman filter for radar tracking. *IEEE Transactions on Signal Processing, 64*(4), 948–958. [CrossRef]

[16] Knudsen, T., & Leth, J. (2018). A new continuous discrete unscented Kalman filter. *IEEE Transactions on Automatic Control, 64*(5), 2198–2205. [CrossRef]

[17] Zhang, T., Xu, X., & Wang, Z. (2018). Spacecraft attitude estimation based on matrix Kalman filter and recursive cubature Kalman filter. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 232*(16), 3024–3033. [CrossRef]

[18] Lou, T. S., Wu, H. C., Yue, Z. P., Dong, Y. S., & He, Z. D. (2021, October). Adaptive lattice Kalman Filter-SLAM for robot auto-navigation. In *2021 China Automation Congress* (*CAC*) (pp. 4497–4501). IEEE. [CrossRef]

[19] Dutta, A., & Das, M. (2022, December). ECG signal denoising using adaptive unscented Kalman filter. In *2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation* (*IATMSI*) (pp. 1–6). IEEE. [CrossRef]

[20] Nagui, N., Attallah, O., Zaghloul, M. S., & Morsi, I. (2021). Improved GPS/IMU loosely coupled integration scheme using two Kalman filter-based cascaded stages. *Arabian Journal for Science and Engineering, 46*, 1345–1367. [CrossRef]

[21] Li, L. Q., Sun, Y. C., & Liu, Z. X. (2021). Maximum fuzzy correntropy Kalman filter and its application to bearings-only maneuvering target tracking. *International Journal of Fuzzy Systems, 23*(2), 405–418. [CrossRef]

[22] Nguyen, M. T., Nguyen, C. V., Do, H. T., Hua, H. T., Tran, T. A., Nguyen, A. D., ... & Viola, F. (2021). Uav-assisted data collection in wireless sensor networks: A comprehensive survey. *Electronics, 10*(21), 2603. [CrossRef]

[23] Jouaber, S., Bonnabel, S., Velasco-Forero, S., & Pilte, M. (2021, June). NNAKF: A neural network adapted Kalman filter for target tracking. In *ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*) (pp. 4075–4079). IEEE. [CrossRef]

[24] Li, S., Hu, C., Wang, R., Zhou, C., & Yang, J. (2019, December). A maneuvering tracking method based on LSTM and CS model. In *2019 IEEE International Conference on Signal, Information and Data Processing* (*ICSIDP*) (pp. 1–4). IEEE. [CrossRef]

[25] Yaqi, C., You, H. E., Tiantian, T. A. N. G., & Yu, L. I. U. (2022). A new target tracking filter based on deep learning. *Chinese Journal of Aeronautics, 35*(5), 11–24. [CrossRef]

[26] Zhang, J., Wu, Y., & Jiao, S. (2021, November). Research on trajectory tracking algorithm based on LSTM-UKF. In *2021 7th IEEE International Conference on Network Intelligence and Digital Content* (*IC-NIDC*) (pp. 61–65). IEEE. [CrossRef]

[27] Ge, B., Zhang, H., Jiang, L., Li, Z., & Butt, M. M. (2019). Adaptive unscented Kalman filter for target tracking with unknown time-varying noise covariance. *Sensors, 19*(6), 1371. [CrossRef]

[28] Wang, M., Xu, C., Zhou, C., Gong, Y., & Qiu, B. (2022). Study on underwater target tracking technology based on an LSTM–Kalman filtering method. *Applied Sciences, 12*(10), 5233. [CrossRef]

[29] Peng, Y., Pan, X., Wang, S., Wang, C., Wang, J., & Wu, J.

(2021, May). An aero-engine RUL prediction method based on VAE-GAN. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 953–957). IEEE. [CrossRef]

[30] De Miguel, M. Á., Armingol, J. M., & García, F. (2022). Vehicles trajectory prediction using recurrent VAE network. *IEEE Access, 10*, 32742–32749. [CrossRef]

[31] Santhosh, K. K., Dogra, D. P., Roy, P. P., & Mitra, A. (2021). Vehicular trajectory classification and traffic anomaly detection in videos using a hybrid CNN-VAE architecture. *IEEE Transactions on Intelligent Transportation Systems, 23*(8), 11891–11902. [CrossRef]

[32] Uckan, T., Aslan, C., & Hark, C. (2025). A Comprehensive Hybrid Approach for Indoor Scene Recognition Combining CNNs and Text-Based Features. *Sensors, 25*(17), 5350. [CrossRef]

[33] Rezende, D., & Mohamed, S. (2015, June). Variational inference with normalizing flows. In *International Conference on Machine Learning* (pp. 1530–1538). PMLR.

[34] Dinh, L., Krueger, D., & Bengio, Y. (2014). NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*.

[35] Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*.

[36] Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. *Advances in Neural Information Processing Systems, 31*.

[37] Tang, Q., Liang, J., & Zhu, F. (2023). Multisensors fusion for trajectory tracking based on variational normalizing flow. *IEEE Transactions on Geoscience and Remote Sensing, 61*, 1–13. [CrossRef]

[38] Delecki, H., Kruse, L. A., Schlichting, M. R., & Kochenderfer, M. J. (2023, June). Deep normalizing flows for state estimation. In *2023 26th International Conference on Information Fusion (FUSION)* (pp. 1–6). IEEE. [CrossRef]

[39] Papamakarios, G., Pavlakou, T., & Murray, I. (2017). Masked autoregressive flow for density estimation. *Advances in Neural Information Processing Systems, 30*.

[40] Nugraha, A. A., Sekiguchi, K., & Yoshii, K. (2020). A flow-based deep latent variable model for speech spectrogram modeling and enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28*, 1104–1117. [CrossRef]

[41] Sun, S., Zhang, C., & Yu, G. (2006). A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems, 7*(1), 124–132. [CrossRef]

[42] Kong, J., Fan, X., Jin, X., Lin, S., & Zuo, M. (2023). A variational Bayesian inference-based en-decoder framework for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems, 25*(3), 2966–2975. [CrossRef]

[43] Nghiem, T. L., Le, V. D., Le, T. L., Maréchal, P., Delahaye, D., & Vidosavljevic, A. (2022, October). Applying Bayesian inference in a hybrid CNN-LSTM model for time-series prediction. In *2022 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)* (pp. 1–6). IEEE. [CrossRef]

[44] Nayak, A., Eskandarian, A., & Doerzaph, Z. (2022). Uncertainty estimation of pedestrian future trajectory using Bayesian approximation. *IEEE Open Journal of Intelligent Transportation Systems, 3*, 617–630. [CrossRef]

[45] LI Jiaqi, SHI Yunhe, & ZHANG Xiaofei. (2024). Direct localization of UAV targets in dense urban areas based on sparse Bayesian inference. *Journal of Signal Processing, 40*(5), 815–825. [CrossRef]

[46] Posch, K., Steinbrener, J., & Pilz, J. (2019). Variational inference to measure model uncertainty in deep neural networks. *arXiv preprint arXiv:1902.10189*.

[47] Li, H., Ren, X., & Yang, Z. (2023). Data-driven Bayesian network for risk analysis of global maritime accidents. *Reliability Engineering & System Safety, 230*, 108938. [CrossRef]

[48] Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015, June). Weight uncertainty in neural network. In *International Conference on Machine Learning* (pp. 1613–1622). PMLR.

[49] Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., & Bengio, Y. (2015). A recurrent latent variable model for sequential data. *Advances in Neural Information Processing Systems, 28*.

Heran Fu, graduated from Beijing Technology and Business University in 2022 with a bachelor's degree in Electronic Science and Technology, and is currently a master's candidate in Control Engineering of Beijing Technology and Business University. His main research focus on pattern recognition and information fusion, Robot navigation and control, deep learning, and other related fields. (Email: 1002679396@qq.com)

Xuebo Jin received the B.S. and M.S. degrees in control theory and control engineering from Jilin University, Changchun, China, in 1994 and 1997, and the Ph.D. degree in control theory and control engineering from the University of Zhejiang, Zhejiang, China, in 2004.,She was a Senior Visiting Scholar with the University of Illinois at Chicago, Chicago, IL, USA, in2007. From 2009 to 2012, she was an Assistant Professor with Zhejiang Sci-tech University. Since 2012, she has been a Professor with Beijing Technology and Business University, Beijing, China. Her research includes a variety of areas in information fusion, bigdata analysis, condition estimation, and video tracking. (Email: jinxuebo@btbu.edu.cn)