2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Supervisory Measurement-Guided Noise Covariance Estimation

Haoying Li[a], Yifan Peng[a], Xinghan Li[c], and Junfeng Wu[a,b]

***Abstract*— Reliable state estimation hinges on accurate specification of sensor noise covariances, which weigh heterogeneous measurements. In practice, these covariances are difficult to identify due to environmental variability, front-end preprocessing, and other reasons. We address this by formulating noise covariance estimation as a bilevel optimization that, from a Bayesian perspective, factorizes the joint likelihood of so-called odometry and supervisory measurements, thereby balancing information utilization with computational efficiency. The factorization converts the nested Bayesian dependency into a chain structure, enabling efficient parallel computation: at the lower level, an invariant extended Kalman filter with state augmentation estimates trajectories, while a derivative filter computes analytical gradients in parallel for upper-level gradient updates. The upper level refines the covariance to guide the lower-level estimation. Experiments on synthetic and real-world datasets show that our method achieves higher efficiency than existing baselines.**

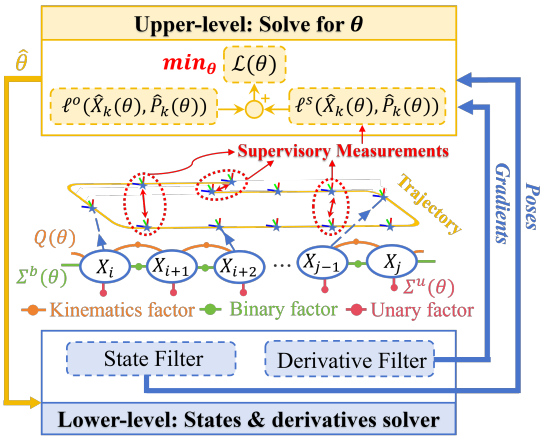


Fig. 1: System framework.

I. INTRODUCTION

Robotic state estimation seeks to recover accurate states such as pose and velocity from noisy multi-sensor data. This is often posed as an optimization problem over sensory data, typically using Maximum Likelihood Estimation (MLE) or Maximum a Posteriori (MAP) inference, where the noise covariance quantifies measurement uncertainty and governs the weighting of measurements, with higher covariances corresponding to less reliable data. Therefore, precise noise covariance specification is vital for weighting measurements appropriately and mitigating unreliable information.

However, obtaining precise knowledge of noise covariance is challenging [1], as sensor characteristics vary with environmental conditions and often require recalibration. Moreover, front-end preprocessing, such as feature extraction, alters raw measurements and obscures their true statistical properties [2]. Noise covariance estimation has consequently received significant research attention. Existing approaches span from MLE and MAP formulations that explicitly optimize noise covariances within probabilistic models [2], [3], to performance-driven methods that tune parameters by minimizing trajectory tracking error [4], [5]. Both gradient-based methods leveraging analytical derivative solutions [6] and derivative-free approaches that avoid explicit gradient computation [7], [8], [9] have been explored.

Despite this progress, two challenges remain: how to fully exploit sensory information and how to estimate noise parameters efficiently. To address them, we partition measurements into odometry, for fast trajectory estimation and gradient evaluation, and supervisory, for complementary noise calibration. Based on an ML formulation, this partition naturally leads to a bilevel optimization problem, as illustrated in Fig. 1. Our key contributions are:

1) **Likelihood factorization in the ML formulation.** We distinguish the roles of odometry and supervisory measurements in forming cross-temporal correlations and separate them in the likelihood derivation. This results in two additive terms—odometry loss and supervisory loss—where supervisory information propagates via a marginal distribution conditioned only on odometry data. The factorization transforms the nested Bayesian network arising from loop closures in SLAM into a chain structure, enabling filter-based methods to incorporate loop closures without excessive complexity, and naturally leads to a bilevel optimization framework.
2) **State Filter and Derivative Filter for bilevel updates.** The chain-structured Bayesian network allows a *State Filter* and a *Derivative Filter* to run in parallel, producing lower-level states and upper-level implicit gradients. Supervisory information influences the solution through covariance–triplet updates rather than nested Bayesian dependencies, reducing complexity. Both filters can be implemented using standard Kalman filtering with state augmentation.
3) **Simulation and Experiment Validation**. We evaluate the proposed method against baseline approaches on synthetic and real-world datasets, demonstrating accuracy and improved efficiency.

[a] School of Data Science, Chinese University of Hong Kong (Shenzhen), Shenzhen, China. [b] School of Artificial Intelligence, Chinese University of Hong Kong (Shenzhen), Shenzhen, China. [c] DeepMirror Inc., Guangzhou, China. Emails: {haoyingli, yifanpeng}@link.cuhk.edu.cn; junfengwu@cuhk.edu.cn; xinghanli0207@gmail.com.

II. Related Works

This section reviews related work on noise covariance estimation, classified by the objectives they optimize.

1) MAP/MLE methods: Noise covariance optimization via MLE/MAP formulations has been extensively studied. While derivative-free methods like expectation maximization jointly optimize parameters and states under likelihood objectives [7], [8], they require full posterior approximation, while point estimation is computationally preferable in practice. Gradient-based methods for parameter estimation in Kalman filtering minimize the negative log-likelihood, also known as the energy function [6], using only the observations directly employed by the filter. The required gradients can be obtained either through forward sensitivity equations [3] or via backpropagation [10]. In [2], states and noise covariances are jointly optimized under a MAP formulation by exploiting the convexity of the joint estimation problem and using an alternating optimization scheme, with most of the computational cost incurred in solving for the states.

2) Performance-driven methods: Several existing works estimate noise parameters based on state-estimation performance. The method in [11] optimizes innovation-based objectives, using measurement residuals in the outer loop and state estimation in the inner loop. The work in [12] proposes an adaptive disturbance estimation for a moving horizon estimator via bilevel optimization with fast derivative computation, while it requires a reference trajectory. The framework in [4] formulates covariance learning as constrained bilevel optimization, which requires ground-truth supervision and uses numerical gradients. Parameters can be optimized to fit the current task using these approaches. However, without probabilistic modeling, the resulting parameters may deviate from the true underlying values.

Extending the aforementioned works, we cast noise parameter estimation as an MLE problem, offering a principled probabilistic basis. We further partition the measurements and design distinct upper- and lower-level objectives, enhancing efficiency while leveraging richer information.

III. Preliminary on Robotic Lie Groups

This section presents the preliminaries on matrix Lie groups used in the derivation of this work, following [13].

Let $\mathbf{G}$ be a matrix Lie group with Lie algebra $\mathfrak{g}$. Note that $\mathfrak{g}$ is isomorphic to $\mathbb{R}^{\dim \mathfrak{g}}$. The exponential map $\exp : \mathbb{R}^{\dim \mathfrak{g}} \to \mathbf{G}, \quad \xi \mapsto \exp_{\mathrm{m}}(\xi^{\wedge})$ relates $\mathfrak{g}$ to its associated group $\mathbf{G}$, where $(\cdot)^{\wedge}$ denotes the skew operator and $\exp_{\mathrm{m}}(\cdot)$ the matrix exponential. In a neighborhood of the identity, $\exp$ is locally invertible, enabling the definition of the logarithm map $\log : \mathbf{G} \to \mathbb{R}^{\dim \mathfrak{g}}, \quad X \mapsto \log(X)^{\vee}$, where $(\cdot)^{\vee}$ is the inverse of $(\cdot)^{\wedge}$. The special orthogonal group $SO(3)$ represents the set of all possible rotations of a rigid body in three-dimensional space. The special Euclidean group $SE(3)$ represents rigid-body transformations, defined as

$$SE(3) \triangleq \left\{ \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \;\middle|\; R \in SO(3), p \in \mathbb{R}^3 \right\}.$$

When velocity is further incorporated into rigid-body motion, the resulting group is $SE_2(3)$, given by

$$SE_2(3) \triangleq \left\{ \begin{bmatrix} R & p & v \\ \mathbf{0} & & I \end{bmatrix} \middle| R \in SO(3), p, v \in \mathbb{R}^3 \right\}, \quad (1)$$

where $\mathbf{0}$ and I denote the zero matrix and the identity matrix, with the dimension omitted when clear from context.

For $x^{\wedge}, y^{\wedge} \in \mathfrak{g}$ and $\|x\|$ small, their compounded exponentials can be approximated to $\exp(x)\exp(y) \approx \exp(\mathrm{dexp}_y^{-1}\, x + y)$, where dexp_x is the left Jacobian of x.

Let $\mathcal{M}$ be the smooth manifold of a Lie group. For any $X_1, X_2 \in \mathcal{M}$ and tangent vector $\xi \in \mathbb{R}^{\dim \mathcal{M}}$, we define the *plus* and *minus* operators, $X_1 \boxplus \xi$ and $X_1 \boxminus X_2$, which map between the manifold and its tangent space. These operators admit left and right versions; the choice is application-dependent and left implicit. For instance, in the left case, $X_1 \boxplus \xi = \exp(\xi) X_1, X_1 \boxminus X_2 = \log(X_1 X_2^{-1})$. On product manifolds, the operators act componentwise [13]. The uncertainty of $X \in \mathcal{M}$ is modeled in the tangent space by a Gaussian distribution [14]. Let $\xi \sim \mathcal{N}(0, \Sigma)$, the induced distribution is

$$\mathcal{N}_L(\bar{X}, \Sigma) \triangleq \eta \exp\!\left(-\tfrac{1}{2}(X \boxminus \bar{X})^{\top} \Sigma^{-1} (X \boxminus \bar{X})\right), \quad (2)$$

where η is the normalization coefficient.

IV. Problem Statement

In this section, we formally state the noise covariance estimation problem.

Consider a mobile robot whose kinematic evolution follows a group-affine discrete-time system as:

$$X_{k+1} = f(X_k, u_k, w_k), \; w_k \sim \mathcal{N}(\mathbf{0}, Q(\theta)), \quad (3)$$

where $X_k \in \mathcal{X}$ denotes the robot's state at time step k on the Lie group $\mathcal{X}$, $u_k \in \mathcal{U}$ is the control input, and w_k is the process noise. Conventionally, introspective sensor measurements (e.g., from IMUs) are used as inputs to the motion model through preintegration, which conflates measurement uncertainty with the process noise w_k [15].

The motion of a robot can be captured using extrospective sensors. In this paper, we categorize their measurements into two types [16]. Unary measurements, including those from GPS receivers, correspond to a single motion state and provide information in a fixed reference frame, whereas binary measurements, including LiDAR or RGB-D point clouds from iterative closest point (ICP), relate two consecutive motion states. At this stage, we do not constrain the sensor types to specific implementations to maintain theoretical completeness. The probabilistic models for these two measurement classes are

$$y_k^b = h(X_{k-1}, X_k) \boxplus \nu_k^b, \qquad \nu^b \sim \mathcal{N}(\mathbf{0}, \Sigma^b(\theta)), \quad (4)$$

$$y_k^u = g(X_k) \boxplus \nu_k^u, \qquad \nu^u \sim \mathcal{N}(\mathbf{0}, \Sigma^u(\theta)), \quad (5)$$

where $y_k^u \in \mathcal{Y}^u, y_k^b \in \mathcal{Y}^b$ with $\mathcal{Y}^u$ and $\mathcal{Y}^b$ denoting the respective output spaces. Let $\boldsymbol{y}_{1:k}^o = \{y_i^u, y_i^b\}_{i=1:k}$ denote the set of unary and consecutive binary observations up

to time step k, corresponding to motion priors or motion observations with covariance parameters to be estimated.

As discussed in Section I, accurate knowledge of process and measurement noise covariances is crucial for odometry but remains difficult to determine in practice. Consequently, we treat these covariances as unknown quantities and introduce a parameterization via a variable θ. This parameterization may take different forms, from simple vectorization of matrix entries (with symmetry, positive definiteness, and other constraints enforced) to more expressive representations based on neural networks. While some prior works [4], [5] leverage ground-truth robot poses as supervisory signals to learn the noise parameters, in this paper, we propose a general approach. We introduce a new class of measurements—termed *supervisory measurements*—that directly inform the robot's relative poses over extended trajectories with the known measurement covariance. These measurements provide supervision for estimating θ. Formally, the supervisory measurement model is

$$y_{ij}^s = s(X_i, X_j) \boxplus \nu_{ij}, \quad \nu_{ij} \sim \mathcal{N}(\mathbf{0}, \psi), \tag{6}$$

where ψ denotes the covariance. Typical examples include loop closures and ground-truth poses, the latter can be interpreted as y_{0j}^s, i.e., relative poses with respect to the initial pose or another fixed frame.

Remark 1. *Ground-truth poses, as supervisory measurements obtained from motion capture systems or other high-precision sources, are typically modeled with negligible or zero uncertainty. Under this assumption, one can set $\psi = \mathbf{0}$ in* (6)*. This corresponds to a Dirac delta probability, which arises as the limit of a sequence of Gaussian distributions centered at the origin with variances tending to zero.* □

Define the state projection $\pi(y_{ij}^s) = \{X_i, X_j\}$, which takes a supervisory measurement y_{ij}^s and returns the set of states involved in y_{ij}^s. Similarly, the index map is defined as $\pi_{\mathrm{id}}(y_{ij}^s) = \{i, j\}$. Letting $\mathcal{I}_k^s = \{i \in 1, \dots, k \mid \{i, j\} = \pi_{\mathrm{id}}(y_{ij}^s), \exists j \in 1, \dots, k\}$, then $\boldsymbol{y}_{1:k}^s \triangleq \{y_{ij}\}_{\{i,j\} \in \mathcal{I}_k^s}$.

Given odometry measurements $\boldsymbol{y}_{1:N}^o$ and supervisory measurements $\boldsymbol{y}_{1:N}^s$ collected up to time N, the noise covariance estimation problem can be formulated as a MLE problem

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, p(\boldsymbol{y}_{1:N}^o, \boldsymbol{y}_{1:N}^s \mid \theta), \tag{7}$$

where $\hat{\theta}$ is termed the MLE estimate to θ and Θ represents the admissible parameter space.

To ensure clarity in subsequent discussions, some notational conventions are clarified below. The set of states up to time N is denoted in bold italic as $\boldsymbol{X}_{1:N} = \{X_1, \dots, X_N\}$. Within this set, $\boldsymbol{X}_{1:N}^s$ represents the subset of states involved in supervisory measurements, defined as $\boldsymbol{X}_{1:N}^s = \{X_i \mid X_i \in \pi(z_{ij}), \exists z_{ij} \in \boldsymbol{y}_{1:N}^s\}$. For notational simplicity, we may omit the time subscript $1:N$ from the above symbols whenever the time horizon is clear from the context. The remaining states, excluding those in $\boldsymbol{X}^s$, are denoted as $\boldsymbol{X}^o = \boldsymbol{X} \setminus \boldsymbol{X}^s$.

V. Methodology

Our subsequent derivations are based on the following assumption regarding the measurement and process noises in the system.

Assumption 1. *The noises w_k's, ν_k^b's, ν_k^u's, and ν_{ij}'s are mutually independent random variables.* □

The assumption is standard in the SLAM literature [2], [17], enabling tractable theoretical analysis while remaining practically justifiable for most real-world applications.

A. Problem Formulation

By introducing the motion states as latent variables and invoking Assumption 1, the likelihood (7) is factorized as

$$\begin{aligned}
&p(\boldsymbol{y}^o, \boldsymbol{y}^s \mid \theta) \\
&= \int\int p(\boldsymbol{y}^o, \boldsymbol{y}^s, \boldsymbol{X}^o, \boldsymbol{X}^s \mid \theta)\, \mathrm{d}\boldsymbol{X}^o \mathrm{d}\boldsymbol{X}^s \\
&= \int\int p(\boldsymbol{y}^s \mid \boldsymbol{X}^o, \boldsymbol{X}^s, \boldsymbol{y}^o, \theta)\, p(\boldsymbol{X}^o, \boldsymbol{X}^s, \boldsymbol{y}^o \mid \theta)\, \mathrm{d}\boldsymbol{X}^o \mathrm{d}\boldsymbol{X}^s \\
&= \int\int p(\boldsymbol{y}^s \mid \boldsymbol{X}^s)\, p(\boldsymbol{X}^o, \boldsymbol{X}^s \mid \boldsymbol{y}^o, \theta)\, p(\boldsymbol{y}^o \mid \theta)\, \mathrm{d}\boldsymbol{X}^o \mathrm{d}\boldsymbol{X}^s,
\end{aligned}$$

where the integrations are taken over some product spaces of $\mathcal{X}$ by default. Further marginalizing out $\boldsymbol{X}^o$ yields

$$\begin{aligned}
&p(\boldsymbol{y}^o, \boldsymbol{y}^s \mid \theta) \\
=&p(\boldsymbol{y}^o \mid \theta) \int p(\boldsymbol{y}^s \mid \boldsymbol{X}^s) \left(\int p(\boldsymbol{X}^o, \boldsymbol{X}^s \mid \boldsymbol{y}^o, \theta)\, \mathrm{d}\boldsymbol{X}^o \right) \mathrm{d}\boldsymbol{X}^s \\
=&p(\boldsymbol{y}^o \mid \theta) \int p(\boldsymbol{y}^s \mid \boldsymbol{X}^s)\, p(\boldsymbol{X}^s \mid \boldsymbol{y}^o, \theta)\, \mathrm{d}\boldsymbol{X}^s.
\end{aligned}$$

The negative log-likelihood is decomposed into two terms

$$L(\theta) \triangleq -\log p(\boldsymbol{y}^o, \boldsymbol{y}^s \mid \theta) = \ell^o(\theta) + \ell^s(\theta),$$

where $\ell^o(\theta) \triangleq -\log p(\boldsymbol{y}^o \mid \theta)$ and

$$\ell^s(\theta) \triangleq -\log \int p(\boldsymbol{y}^s \mid \boldsymbol{X}^s)\, p(\boldsymbol{X}^s \mid \boldsymbol{y}^o, \theta)\ \mathrm{d}\boldsymbol{X}^s. \tag{8}$$

We refer to $\ell^o(\theta)$ as the *odometry loss* and $\ell^s(\theta)$ as the *supervisory loss*, with $\ell^o(\theta)$ admits the following factorization

$$\ell^o(\theta) = -\sum_{k=1}^{N} \log p\left(y_k^o \mid \boldsymbol{y}_{1:k-1}^o, \theta\right), \tag{9}$$

where it further holds that

$$p\left(y_k^o \mid \boldsymbol{y}_{1:k-1}^o, \theta\right) = \int p\left(y_k^o \mid X_k, \theta\right) p\left(X_k \mid \boldsymbol{y}_{1:k-1}^o, \theta\right) \mathrm{d}X_k.$$

where $p(y_k^o \mid X_k, \theta)$ represents the measurement models (4) (5), and $p\left(X_k \mid \boldsymbol{y}_{1:k-1}^o, \theta\right)$ is the Gaussian predictive distribution. With the above factorization in place, the distribution $p\left(X_k \mid \boldsymbol{y}_{1:k-1}^o, \theta\right)$ and $p(\boldsymbol{X}^s \mid \boldsymbol{y}^o, \theta)$ remain to be derived. We tone down the hope of acquiring exact computation for them, for exact analytical expressions for these distributions are generally intractable and rarely available in closed form. Therefore, in what follows, we compute them by resorting to the local linearization of the measurement model within the tangent space of X_k.

B. State Filter

This part presents the design of an invariant extended Kalman filter (InEKF) that recursively computes $p\left(X_k \mid \boldsymbol{y}^o_{1:k-1}, \theta\right)$ with states augmentation. As will be shown in the subsequent derivation, the augmented state in the InEKF enables the calculation of $p(\boldsymbol{X}^s \mid \boldsymbol{y}^o, \theta)$.

To keep track of the correlation between states in $\boldsymbol{X}^s_k$, first define the following composite state

$$Y_k = [X_{k-1}^\top, X_k^\top, X_{i_{k,1}}^\top, \ldots, X_{i_{k,l}}^\top]^\top.$$

In the above definition, $X_{i_{k,1}}, \ldots, X_{i_{k,l}}$ collectively constitute $\boldsymbol{X}^s_k$, with a predetermined index set $\mathcal{I}^s_k$ (e.g., the indices of keyframe candidates selected for loop closures).

The notations used in the state filter are clarified here. Let $\bar{Y}_k, \bar{P}_k$ and $\hat{Y}_k, \hat{P}_k$ denote the prior and the posterior estimates with the estimation error covariance of Y_k. The estimates and associated errors are given by

$$\begin{aligned}\bar{Y}_k =&[\hat{X}_{k-1|k-1}^\top, \bar{X}_k^\top, \bar{X}_{i_{k,1}}^\top, \ldots, \bar{X}_{i_{k,l}}^\top]^\top,\ \bar{\zeta}_k \triangleq Y_k \boxminus \bar{Y}_k,\\ \hat{Y}_k =&[\hat{X}_{k-1|k}^\top,\ \hat{X}_{k|k}^\top, \hat{X}_{i_{k,1}}^\top, \ldots, \hat{X}_{i_{k,l}}^\top]^\top,\ \hat{\zeta}_k \triangleq Y_k \boxminus \hat{Y}_k,\end{aligned}$$

where the state estimate at time $k-1$ after incorporating y^b_{k-1} is denoted by $\hat{X}_{k-1|k-1}$, while $\hat{X}_{k-1|k}$ refers to the estimate obtained after incorporating y^b_k.

Upon receiving the input u_{k-1}, while other states remain unchanged, $\hat{X}_{k-1|k-1}$ is propagated through the nominal part of (3) as follows:

$$\bar{X}_k = f(\hat{X}_{k-1|k-1}, u_{k-1}, \mathbf{0}) \tag{10}$$

The resulting a priori estimation error and its covariance can be written in partitioned form as

$$\bar{\zeta}_k = [(\bar{\zeta}^o_k)^\top \quad (\bar{\zeta}^s_k)^\top]^\top, \quad \bar{P}_k = \begin{bmatrix} \bar{P}^o_k & \bar{P}^{os}_k \\ \bar{P}^{so}_k & \bar{P}^s_k \end{bmatrix},$$

where $(\cdot)^o$ corresponds to the two-frame odometry states at time step k, i.e., X_{k-1}, X_k, and $(\cdot)^s$ relates to the $\boldsymbol{X}^s_k$. The estimation error covariance matrix evolves as

$$\bar{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^\top + Q_{0,k}, \tag{11}$$

where $F_{k-1} = \left[\begin{smallmatrix} \Phi_{k-1} & \mathbf{0} \\ \mathbf{0} & I \end{smallmatrix}\right]$, $Q_{0,k} = \left[\begin{smallmatrix} Q_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{smallmatrix}\right]$. The computation of Φ_{k-1} and Q_k follow [18], which employs the first-order approximation of the Baker–Campbell–Hausdorff (BCH) formula and owing to the log-linear property of group-affine dynamics, Φ_{k-1} is independent of X_{k-1}.

The update step serves to incorporate odometry measurements. For generality, unary and binary measurements together form a stacked residual

$$r_k = [(r^u_k)^\top \quad (r^b_k)^\top]^\top, \tag{12}$$

where $r^u_k \triangleq y^u_k \boxminus g(\bar{X}_k)$ and $r^b_k \triangleq y^b_k \boxminus h(\hat{X}_{k-1|k-1}, \bar{X}_k)$. In practice, only the available measurements are included in this step. The residual Jacobian and the corresponding measurement noise covariance take the form

$$H_k = \begin{bmatrix} H^u_k \\ H^b_k \end{bmatrix}, \Sigma_k = \begin{bmatrix} \Sigma^u & 0 \\ 0 & \Sigma^b \end{bmatrix}, \tag{13}$$

where $H^b_k \triangleq \left[\left.\frac{\partial r^b_k(\hat{X}_{k-1|k-1}\boxplus\xi)}{\partial\xi}\right|_{\xi=\mathbf{0}} \quad \left.\frac{\partial r^b_k(\bar{X}_k\boxplus\xi)}{\partial\xi}\right|_{\xi=\mathbf{0}}\right]$ and $H^u_k \triangleq \left.\frac{\partial r^u_k(\bar{X}_k\boxplus\xi)}{\partial\xi}\right|_{\xi=\mathbf{0}}$. Let $H_{0,k} = [H_k \ \ \mathbf{0}]$, it follows that

$$S_k = H_{0,k}\bar{P}_k H_{0,k}^\top + \Sigma_k = H_k\bar{P}^o_k H_k^\top + \Sigma_k, \tag{14}$$

$$K_k = \bar{P}_k H_{0,k}^\top S_k^{-1}, \quad \check{P}_k = \bar{P}_k - K_k H_{0,k}\bar{P}_k \tag{15}$$

$$\hat{Y}_k = \bar{Y}_k \boxplus (K_k r_k). \tag{16}$$

Once the update completes, $\hat{X}_k$ will append to the end of $\hat{Y}_k$ if $X_k \in \boldsymbol{X}^s_k$, resulting in the covariance update

$$\hat{P}_k = J_k\check{P}_k J_k^\top, \tag{17}$$

where $J_k = [I \ J^\top]^\top$ with $J = [I \ \mathbf{0}]$ when the appending takes place, and $J_k = I$, otherwise.

Remark 2 (Dimension Control)**.** *In the absence of unary measurements, cross-temporal correlations vanish and state augmentation can be omitted; otherwise, the number of augmented states can be deliberately limited to control computational complexity. In practice, calibration trajectories are typically short, and downsampling can be applied if needed. Specifically, with M states for supervisory measurements, yields at most $M(M-1)/2$ pairwise distinct observations—generally sufficient to support reliable calibration.* □

C. Derivatives of ℓ^o and ℓ^s with respect to θ

By substituting the Gaussian predictive distribution from Subsection V-B into (9) and discarding additive constants, the odometry loss takes the closed-form expression [6]

$$\ell^o(\theta) \cong \sum_{k=1}^{N} \frac{1}{2}\log|S_k(\theta)| + \frac{1}{2}r_k(\theta)^\top (S_k(\theta))^{-1} r_k(\theta), \tag{18}$$

where r_k is from (12) and S_k is from (14) and the symbol $\cong$ denotes equivalence up to the omitted additive constants.

For ease of presentation, we define $l^o_k(\theta) \triangleq \frac{1}{2}\log|S_k(\theta)| + \frac{1}{2}r_k(\theta)^\top S_k(\theta)^{-1} r_k(\theta)$. As shown in (18), evaluating the gradient of $\ell^o(\theta)$ with respect to θ requires the derivatives of S_k and r_k with respect to θ, which in turn necessitates a sensitivity analysis of the state filter. Following [6], we apply a term-wise differential to the state filter (11)–(17), which is referred to as the *derivative filter*.

Theorem 1 (Derivative of ℓ^o)**.** *The derivative of $\ell^o(\theta)$ with respect to each component θ_j of θ is the sum of its derivatives from each time step:*

$$\frac{\partial\ell^o(\theta)}{\partial\theta_j} = \sum_{k=1}^{N} \frac{\partial l^o_k(\theta)}{\partial\theta_j}. \tag{22}$$

Each constituent term $\frac{\partial l^o_k(\theta)}{\partial\theta_j}$ has the form

$$\frac{1}{2}\operatorname{tr}(S_k^{-1}\frac{\partial S_k}{\partial\theta_j}) + (\frac{\partial r_k}{\partial\theta_j})^\top S_k^{-1} r_k - \frac{1}{2}r_k^\top S_k^{-1}(\frac{\partial S_k}{\partial\theta_j})S_k^{-1} r_k, \tag{23}$$

with $S_k(\theta)$ and $r_k(\theta)$ abbreviated as S_k and r_k. The computation of $\partial S_k/\partial\theta_j$ and $\partial r_k/\partial\theta_j$ follows (19) *and* (21).□

The following describes the computation of ℓ^s and the derivation of its gradient. Specifically, the MAP estimate

from $p(\boldsymbol{X}^s \mid \boldsymbol{y}^o, \theta)$ is $[\hat{X}_{i_{N,1}}^\top, \ldots, \hat{X}_{i_{N,l}}^\top]^\top$ with $\hat{P}_N^s$ extracted from the output of the state filter. For notational simplicity, we define $\hat{\boldsymbol{X}}^s \triangleq [(\hat{X}_1^s)^\top, \ldots, (\hat{X}_l^s)^\top]^\top$.

For each $y_{ik}^s \in \boldsymbol{y}^s$, the residual is defined as $v_{ik} = y_{ik}^s \boxminus s(\hat{X}_i^s, \hat{X}_k^s)$, and its Jacobian w.r.t. the error state of $\hat{Y}^s$ has the following form

$$H_{ik}^s = \begin{bmatrix} \mathbf{0} & \frac{\partial v_{ik}(\hat{X}_i^s \boxplus \xi)}{\partial \xi}\big|_{\xi=\mathbf{0}} & \mathbf{0} & \frac{\partial v_{ik}(\hat{X}_k^s \boxplus \xi)}{\partial \xi}\big|_{\xi=\mathbf{0}} & \mathbf{0} \end{bmatrix} \tag{24}$$

Stacking all residuals gives the aggregate vector and Jacobian

$$v \triangleq \mathrm{col}\{v_{ik}\}, \qquad H^s \triangleq \mathrm{col}\{H_{ik}^s\}, \tag{25}$$

where $\mathrm{col}(\cdot)$ stacks vectors or block rows vertically.

Given the above stacked measurements associated with the supervisory states, the uncertainty of $\boldsymbol{X}^s$ is modeled in the tangent space as in (2), given by $p(\boldsymbol{y}^s \mid \boldsymbol{X}^s) = \mathcal{N}(\boldsymbol{y}^s \mid H^s \zeta_N^s, \Psi)$ and $p(\boldsymbol{X}^s \mid \boldsymbol{y}^o, \theta) = \mathcal{N}(\zeta_N^s \mid \hat{\zeta}_N^s, \hat{P}_N^s)$. Substituting these distributions into (8) yields the expression for $\ell^s(\theta)$. By applying the chain rule, Theorem 2 establishes the derivative of ℓ^s with respect to θ.

Theorem 2 (Derivative of ℓ^s)**.** *The supervisory loss ℓ^s is computed as*

$$\ell^s(\theta) \cong \frac{1}{2}\log|C(\theta)| + \frac{1}{2} v(\theta)^\top C(\theta)^{-1} v(\theta), \tag{26}$$

where v is given by (25) *and $C = H^s \hat{P}_N^s(\theta)(H^s)^\top + \Psi$ with Ψ denoting the block-diagonal matrix formed from ψ. The derivative of ℓ^s with respect to θ_j is*

$$\begin{aligned} \frac{\partial \ell^s(\theta)}{\partial \theta_j} = \frac{1}{2}\,\mathrm{tr}\left(C^{-1}\frac{\partial C}{\partial \theta_j}\right) + \left(\frac{\partial v}{\partial \theta_j}\right)^\top C^{-1} v \\ - \frac{1}{2} v^\top C^{-1}\left(\frac{\partial C}{\partial \theta_j}\right) C^{-1} v. \end{aligned} \tag{27}$$

where $\partial C/\partial\theta_j$ and $\partial v/\partial\theta_j$ are computed as

$$\begin{aligned} \frac{\partial C}{\partial \theta_j} &= \frac{\partial H^s}{\partial \theta_j}\hat{P}_N^s (H^s)^\top + H^s \frac{\partial \hat{P}_N^s}{\partial \theta_j}(H^s)^\top + H^s \hat{P}_N^s \left(\frac{\partial H^s}{\partial \theta_j}\right)^\top, \\ \frac{\partial v}{\partial \theta_j} &= -H^s \frac{\partial \hat{\zeta}_N^s}{\partial \theta_j}, \quad \frac{\partial H^s}{\partial \theta_j} = \frac{\partial H^s}{\partial \hat{\zeta}_N^s}\frac{\partial \hat{\zeta}_N^s}{\partial \theta_j}. \end{aligned}$$

□

Combining $\ell^o(\theta)$ (18) and $\ell^s(\theta)$ (26), the full negative log-likelihood up to an additive constant is expressed as

$$\begin{aligned} L(\theta) &\triangleq \mathcal{L}(\bar{\boldsymbol{X}}(\theta), \bar{\boldsymbol{P}}(\theta), \hat{\boldsymbol{X}}^s(\theta), \hat{P}_N^s(\theta)) \\ &\cong \ell^o(\bar{\boldsymbol{X}}(\theta), \bar{\boldsymbol{P}}(\theta)) + \ell^s(\hat{\boldsymbol{X}}^s(\theta), \hat{P}_N^s(\theta)). \end{aligned}$$

Consequently, the measurement noise covariance estimation problem can be formulated as the following bilevel optimization problem [1]:

Problem 1.

$$\textit{(Upper level)} \quad \hat{\theta} = \arg\min_{\theta \in \Theta} \mathcal{L}(\hat{\boldsymbol{X}}(\theta), \hat{\boldsymbol{P}}(\theta)),$$

$$\textit{(Lower level)} \quad \textit{s.t.}\{\hat{\boldsymbol{X}}(\theta), \hat{\boldsymbol{P}}(\theta)\} = \operatorname*{argmax}_{X_k,\, 1\le k\le N} p(X_k \mid \boldsymbol{y}_{1:k}^o, \theta).$$

To address Problem 1, the lower-level is solved using the state filter introduced in Section V-B. For the upper-level problem, a gradient descent method is employed, where the gradient is given by

$$\nabla_\theta \mathcal{L} = \nabla_\theta \ell^o + \nabla_\theta \ell^s, \tag{28}$$

where $\nabla_\theta \ell^o$ is obtained from Theorem 1, and $\nabla_\theta \ell^s$ is derived according to Theorem 2. In summary, the computation diagram is shown in Fig. 2.

Algorithm 1 Supervisory Measurement-Guided Noise Covariance Estimation

Require: Initial state X_0, observations $\boldsymbol{y}^o$, $\boldsymbol{y}^s$, initial hyperparameters $\theta^{(0)}$, threshold ϵ, max_iter
Ensure: Noise covariance parameters θ

```
Initialize iteration counter i ← 0
while ‖∇_θ L(θ)‖ > ε and i < max_iter do
    Lower-Level :
    Run state filter (11)-(17) to obtain X̂(θ^(i)), P̂(θ^(i)), and concurrently obtain ∇_θ ℓ^o(θ^(i)) via (22)
    Upper-Level :
    Compute ∇_θ ℓ^s(θ^(i)) via (27)
    Evaluate ∇_θ L(θ^(i)) using (28)
    Update θ̃^(i+1) ← θ^(i) − η^(i) ∇_θ L(θ̂)
    θ^(i+1) ← Projection_Θ(θ̃^(i+1))
    i ← i + 1
end while
```

[1] To simplify notation, we allow an ambiguity by using $\hat{\boldsymbol{X}}(\theta)$ and $\hat{\boldsymbol{P}}(\theta)$ to denote both $\bar{\boldsymbol{X}}(\theta), \hat{\boldsymbol{X}}^s(\theta)$ and $\bar{\boldsymbol{P}}(\theta), \hat{\boldsymbol{P}}_N^s(\theta)$.

$$\begin{aligned} &\frac{\partial \bar{P}_k}{\partial \theta_j} = F_k \frac{\partial \hat{P}_{k-1}}{\partial \theta_j} F_k^\top + \frac{\partial Q_{0,k}}{\partial \theta_j}, \quad \frac{\partial H_k}{\partial \theta_j} = \frac{\partial H_k}{\partial \bar{\zeta}_k}\frac{\partial \bar{\zeta}_k}{\partial \theta_j}, \\ &\frac{\partial S_k}{\partial \theta_j} = H_k \frac{\partial \bar{P}_k}{\partial \theta_j} H_k^\top + \left(\frac{\partial H_k}{\partial \theta_j}\right)\bar{P}_k H_k^\top + H_k \bar{P}_k \left(\frac{\partial H_k}{\partial \theta_j}\right)^\top + \frac{\partial \Sigma_k}{\partial \theta_j}, \qquad (19) \\ &\frac{\partial W_k}{\partial \theta_j} = -W_k \frac{\partial S_k}{\partial \theta_j} W_k, \quad \frac{\partial K_k}{\partial \theta_j} = \frac{\partial \bar{P}_k}{\partial \theta_j} H_k^\top W_k + \bar{P}_k \left(\frac{\partial H_k}{\partial \theta_j}\right)^\top W_k + \bar{P}_k H_k^\top \frac{\partial W_k}{\partial \theta_j}, \\ &\frac{\partial \bar{P}_k}{\partial \theta_j} = \frac{\partial \bar{P}_k}{\partial \theta_j} - \frac{\partial K_k}{\partial \theta_j} H_k \bar{P}_k - K_k \left(\frac{\partial H_k}{\partial \theta_j}\right)\bar{P}_k - K_k H_k \frac{\partial \bar{P}_k}{\partial \theta_j}, \qquad (20) \\ &\frac{\partial \hat{P}_k}{\partial \theta_j} = J_k \frac{\partial \check{P}_k}{\partial \theta_j} J_k^\top. \end{aligned}$$

$$\begin{aligned} &\frac{\partial \bar{\zeta}_k}{\partial \theta_j} = F_k \frac{\partial \hat{\zeta}_{k-1}}{\partial \theta_j}, \\ &\frac{\partial r_k}{\partial \theta_j} = -H_k \frac{\partial \bar{\zeta}_k}{\partial \theta_j}, \qquad (21) \\ &\frac{\partial \hat{\zeta}_k}{\partial \theta_j} = \mathcal{J}_k \frac{\partial \bar{\zeta}_k}{\partial \theta_j} + \frac{\partial K_k}{\partial \theta_j} r_k + K_k \frac{\partial r_k}{\partial \theta_j}, \\ &\frac{\partial \hat{\zeta}_k}{\partial \theta_j} = J_k \frac{\partial \hat{\zeta}_k}{\partial \theta_j}, \end{aligned}$$

where $\frac{\partial \hat{P}_0}{\partial \theta_j} = 0$ and $\frac{\partial \hat{\zeta}_0}{\partial \theta_j} = 0$ at initialization, and $\mathcal{J}_k \triangleq \mathrm{dexp}^{-1}_{(K_k r_k)}$.

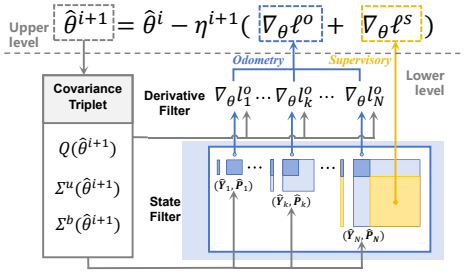


Fig. 2: Logical flow for solving the bilevel problem.

D. Methodology Summary and More Discussions

The overall execution of the noise–covariance estimation procedure is summarized in Algorithm 1. At the upper level, the problem is posed as a constrained optimization, where the operation Projection_Θ projects the estimated parameters onto the feasible set Θ. To improve numerical stability, feasibility constraints can incorporate bounds on the eigenvalues or the condition number of the covariance matrix [4]. In this work, the step size $\eta^{(i)}$ is obtained using the Armijo line search rule. More generally, once the objective function and its gradient are available, a wide range of gradient-based optimization algorithms can be employed.

Multiple parameterizations are possible for noise covariance matrices. A widely adopted approach is Cholesky decomposition [10], which inherently guarantees positive definiteness, while the diagonal form [4] is a special case. Our framework is agnostic to the specific choice of parameterization and can accommodate more expressive representations, including differentiable neural network layers [12].

The proposed method is not limited to the InEKF. For a general EKF, the required derivatives can be obtained via sensitivity equations. We focus on the InEKF and affine group systems in this work, as they are prevalent in practice and possess favorable linearization properties [19] that simplify the derivative–filter design.

Finally, we briefly relate our approach to prior work. Noise covariance estimation is often posed as alternating optimization over states and covariances, using either a shared or separate objective at two levels. We start with the covariance-level objective $L(\theta)$, which is the log-likelihood of the parameters of interest conditioned on all measurements, as derived from a Bayesian framework. Several prior works [4], [5], [12] tune hyperparameters, including noise covariances, by minimizing state-estimation losses against ground-truth poses, $L(\hat{\boldsymbol{X}}(\theta))$. Although effective in reducing trajectory error, the resulting covariances may not reflect the true noise statistics well. To remove reliance on ground truth, [20] introduced the approximate posterior error, which in the EKF reduces to the trace of the estimation error covariance, i.e., $L(\hat{\boldsymbol{P}}(\theta))$. Innovation-based methods learn covariances from measurement residuals, leading to $L(\hat{\boldsymbol{X}}(\theta))$. The traditional energy minimization approach [3], [6] corresponds to the loss $\ell^o(\theta)$, whereas our formulation incorporates the prevalent supervisory measurements in SLAM as an additional source of information to reinforce estimation performance. More recently, [2] proposed joint MAP estimation of states and noise covariances with closed-form covariance updates, but this framework is incompatible with filter-based state estimation and entails substantial cost due to repeated full-state optimization refinement.

VI. Simulation and Experiment

A. Baselines

We compare our method against three baselines. In general, all baseline approaches adopt a bilevel structure for parameter tuning. The lower-level problem is the same for all methods: optimizing the states conditioned on a given noise covariance. The primary distinction lies in the upper-level loss function for the parameters, detailed as follows.

1) *Approximate posterior error (APE)* [9].

$$\mathcal{L}^{\text{APE}}(\theta) = 1/N \sum_{k=1}^{N} \text{tr}(\hat{P}_k). \tag{29}$$

2) *Mean squared error of state estimate (MSE)* [4].

$$\mathcal{L}^{\text{MSE}}(\theta) = 1/N \sum_{k=1}^{N} \|\hat{T}_k \boxminus T_k^{\text{gt}}\|_2^2, \tag{30}$$

where $T_k^{\text{gt}} \in SE(3)$ denote the ground truth poses, and $\hat{T}_k \in SE(3)$ is the estimation pose from the filter.

3) *Average innovation (Innov.)* [11].

$$\mathcal{L}^{\text{innov}}(\theta) = 1/N \sum_{k=1}^{N} \|h(\xi_k) \boxminus y_k\|_2^2. \tag{31}$$

The implementations of gradient descent optimization in [4], [11] follow the original papers, which rely on numerical gradients. The approach of [9] uses Bayesian optimization for general hyperparameters, whereas we focus on a differentiable noise parameter and apply gradient descent.

B. Simulation

We first evaluate the algorithm in simulation. The state $X_k \in SE_2(3)$ as in (1) is estimated using a left-invariant extended Kalman filter (LInEKF). The system kinematics is given by the IMU, where in LInEKF the process noise covariance is $\text{blkdiag}(Q_g, \mathbf{0}, Q_a)$ [18], with gyroscope noise $n_g \sim \mathcal{N}(\mathbf{0}, Q_g)$ and accelerometer noise $n_a \sim \mathcal{N}(\mathbf{0}, Q_a)$. The unary measurements given by GPS and binary measurements given by visual odometry (VO) are specified as $y^u(X_k) = p_k + \nu^u,\ y^b(X_k, X_{k+1}) = \exp(\nu^b)(X_k)^{-1}X_{k+1}$, with Gaussian noises $\nu^u \sim \mathcal{N}(0, \Sigma^u),\ \nu^b \sim \mathcal{N}(0, \Sigma^b)$. The supervisory measurements are the relative poses provided by loop closures. The trajectory is shown in Fig. 3 (a). The first 20 seconds are used for noise covariance estimation, termed the calibration stage, with several supervisory measurements generated. The remaining 70 seconds are used for testing, with both stages sharing the same noise parameters.

The proposed method is evaluated against baseline approaches, with the MSE on the test set adopted as the performance metric. Unary and binary measurement noises are modeled as diagonal matrices of size three and six. Process noise is parameterized as $Q_a = \exp(\theta_{Q_a})I$ and $Q_g = \exp(\theta_{Q_g})I$, with $\theta_j \in [-6, 2]$ ensuring positivity. To assess supervisory effects, we compare 780 loop closures from 40 keyframes (*loop+*) and 190 from 20 downsampled

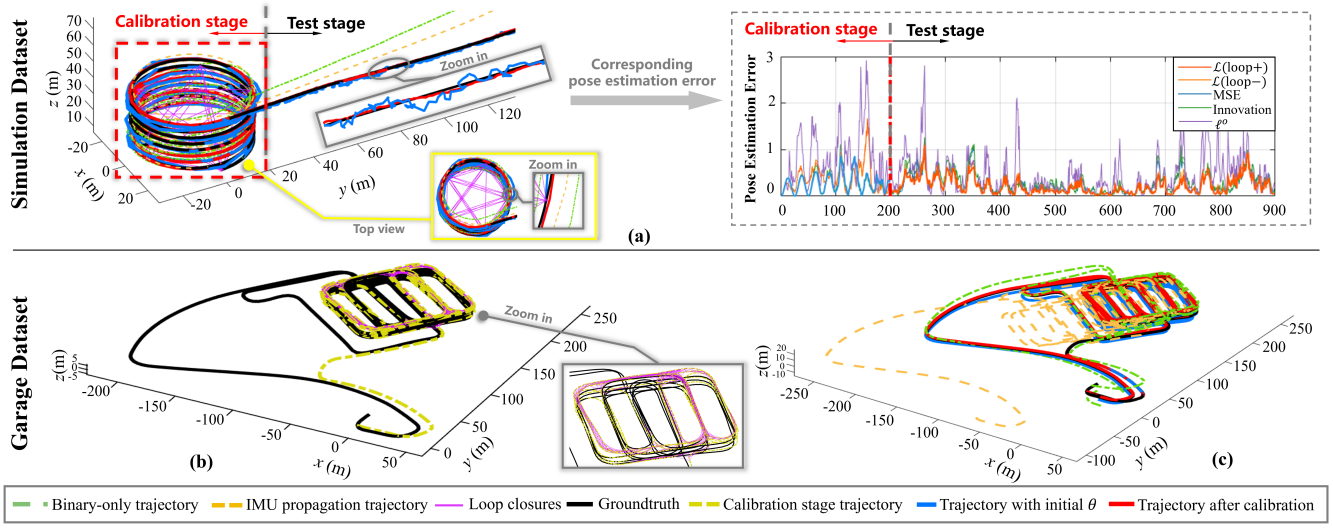


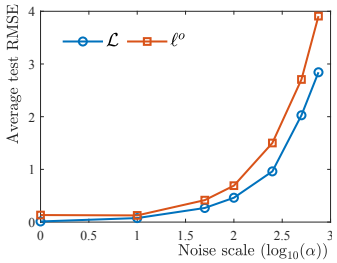
Fig. 3: (a) Loop-to-Open dataset (Synthetic). Left: trajectory with calibration results; Right: pose estimation error over time. The lowest-MSE methods are highlighted during two stages. (b) Visualization of the Garage dataset. (c) Trajectory comparison using only IMU or binary measurements, and trajectories before and after parameter tuning.

TABLE I: Comparison of different optimization methods across datasets.

Dataset	Sensor	Real Noise	Init. Value	ℓ^o only	APE	Innov.	MSE	Ours (loop-)	Ours (loop+)
Loop-to-Open	GPS	4.00×10^{0}	1.00×10^{0}	3.76×10^{0}	8.21×10^{-4}	8.17×10^{1}	5.92×10^{1}	4.27×10^{0}	4.52×10^{0}
		9.00×10^{0}	1.00×10^{0}	8.04×10^{0}	8.08×10^{-4}	1.00×10^{2}	1.00×10^{2}	8.99×10^{0}	9.78×10^{0}
		1.00×10^{0}	1.00×10^{0}	8.90×10^{-1}	1.08×10^{-5}	1.76×10^{1}	6.59×10^{0}	8.80×10^{-1}	8.70×10^{-1}
	IMU	1.00×10^{-6}	1.00×10^{-1}	5.48×10^{-4}	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}	1.00×10^{-6}
		1.00×10^{-4}	1.00×10^{-1}	1.06×10^{-1}	1.12×10^{-2}	2.23×10^{-3}	1.34×10^{-3}	6.67×10^{-3}	5.36×10^{-3}
	VO	1.00×10^{-4}	1.00×10^{1}	3.91×10^{-4}	9.98×10^{0}	9.98×10^{0}	9.99×10^{0}	3.30×10^{-2}	3.30×10^{-4}
		2.50×10^{-3}	1.00×10^{1}	3.08×10^{-3}	9.99×10^{0}	9.99×10^{0}	9.99×10^{0}	3.22×10^{-2}	3.24×10^{-3}
		1.00×10^{-4}	1.00×10^{1}	3.74×10^{-4}	9.98×10^{0}	9.97×10^{0}	9.98×10^{0}	3.46×10^{-2}	3.45×10^{-4}
		1.00×10^{-6}	1.00×10^{1}	9.62×10^{-3}	5.79×10^{0}	6.87×10^{0}	9.98×10^{0}	9.31×10^{-3}	8.45×10^{-3}
		1.00×10^{-6}	1.00×10^{1}	3.84×10^{-3}	5.42×10^{0}	4.99×10^{0}	5.08×10^{0}	2.39×10^{-3}	1.67×10^{-3}
		2.50×10^{-5}	1.00×10^{1}	5.76×10^{-2}	5.19×10^{0}	7.63×10^{0}	8.56×10^{0}	5.10×10^{-2}	4.45×10^{-2}
	Test MSE	–	3.90×10^{0}	4.40×10^{-1}	3.59×10^{0}	3.30×10^{-1}	3.20×10^{-1}	2.40×10^{-1}	$\mathbf{2.10 \times 10^{-1}}$
Garage Dataset	IMU	1.00×10^{-6}	1.00×10^{-2}	1.00×10^{-6}	1.03×10^{-2}	1.09×10^{-2}	9.98×10^{-4}	1.00×10^{-6}	1.00×10^{-6}
		1.00×10^{-4}	1.00×10^{-2}	1.51×10^{-5}	1.04×10^{-2}	9.98×10^{-4}	1.00×10^{-3}	2.78×10^{-5}	1.14×10^{-3}
	Test MSE	–	5.81×10^{0}	2.25×10^{0}	3.12×10^{0}	1.54×10^{0}	1.49×10^{0}	9.30×10^{-1}	$\mathbf{8.10 \times 10^{-1}}$

keyframes (*loop-*), with the standard deviation of the loop noise set to 10^{-3}. True parameters and results are reported in TABLE I. Innovation and MSE assess only the states and ignore covariances, though relative reliability (e.g., smaller noise for GPS z-axis) can still be observed. For binary measurements, all algorithms are less sensitive to the exact noise levels. Our method, together with the ℓ^o formulation, offers a likelihood-based approach that yields covariances closer to the ground truth. The comparison of ℓ^o with and without supervisory loops shows that supervisory measurements effectively suppress odometry drift. As illustrated in Fig. 3 (a), while using $\mathcal{L}^{\text{MSE}}$ yields the lowest MSE in the calibration stage but deviates from true parameters, our method attains the lowest MSE in the long open-loop test trajectory.

A Monte Carlo experiment is conducted to evaluate the impact of supervisory measurements by comparing ℓ^o and



α	ℓ^o	$\mathcal{L}$
1	0.20	**0.18**
10	**0.22**	**0.22**
50	**0.22**	0.23
100	0.28	**0.26**
500	0.83	**0.77**
750	1.42	**1.39**

Fig. 4: Test RMSE and Wasserstein error across α.

$\mathcal{L}$ as loss functions. Measurement noises are parameterized by a base level β and a scaling factor α, with $\theta_j = (\alpha\beta_j)^2$, where β_j is sampled from $[1, 2] \times 10^{-4}$ for Q, $[2, 10] \times 10^{-3}$ for Σ^u, and $[1, 6] \times 10^{-4}$ for Σ^b. All algorithms are initialized identically and run for 20 iterations under the same optimizer settings. The average 2-Wasserstein error [2] is

also reported, measuring the discrepancy between estimated and true covariances, with smaller values indicating more accurate estimation. As shown in Fig. 4, $\mathcal{L}$ yields consistently lower MSE than ℓ^o across noise levels and, under large noise, achieves smaller Wasserstein error, indicating more accurate covariance estimation. This demonstrates the additional benefits of incorporating supervisory measurements.

C. Dataset Experiment

The garage dataset [21] provides real-world binary measurements and loop closures for evaluation. It features calibration segments with abundant loops and validation sections with open-loop trajectories. Lacking unary measurements, we use only IMU and binary data. Following convention, we treat the outlier-free estimation using all measurements as groundtruth. Noisy IMU measurements are then generated from this trajectory using known noise parameters, enabling convenient evaluation of the algorithm. As outlier rejection is not the focus of this work, we replace the false positive loop closures with measurements assigned a noise standard deviation of 0.05. In the *loop-* case, 136 loop pairs are used, while in the *loop+* case, 234 pairs are included.

The calibration stage and ground-truth trajectory are illustrated in Fig. 3 (b), where large drift is observed in IMU-only or binary-only estimates. With the initial parameters, the trajectory exhibits large drift; after calibration, accurate estimates are obtained even without absolute information from sensors such as GPS, provided that IMU and binary measurements are appropriately weighted. As shown in TABLE I, the baseline methods $\mathcal{L}^{\text{MSE}}$ and $\mathcal{L}^{\text{innov}}$ improve state estimation but may misestimate covariances, whereas our method achieves the most effective tuning. Furthermore, incorporating more loop closures yields stronger supervision.

VII. Conclusion and Future Work

This work presents a Bayesian framework for noise covariance estimation, formulated as a bilevel optimization via probability factorization. State and derivative filters run concurrently to provide estimates and analytical gradients, enabling the use of supervisory measurements for richer information with high efficiency. The method is validated on simulated and real-world datasets, and remains flexible and extensible. While the supervisory noise covariance is assumed to be known, this is not restrictive since its gradient is explicit, allowing straightforward optimization.

Several aspects of the framework warrant further exploration. We plan to apply the algorithm to real-world SLAM systems such as [17]. Besides, we will explore deployment-oriented parameterizations, for instance, differentiable neural embeddings that adapt measurement noises to context. Moreover, the framework's flexibility also enables tuning of broader hyperparameters, which remains to be investigated.

ACKNOWLEDGMENT

The authors would like to thank Prof. Shuang Li for her valuable insights and constructive suggestions on the ideas and theoretical aspects of this work. This work was supported in part by NSFC under Grant 62273288; in part by Shenzhen Science and Technology Program JCYJ20240813113609013.

References

[1] Kamak Ebadi, Lukas Bernreiter, Harel Biggie, Gavin Catt, Yun Chang, Arghya Chatterjee, Christopher E Denniston, Simon-Pierre Deschênes, Kyle Harlow, Shehryar Khattak, et al. Present and future of slam in extreme environments: The darpa subt challenge. *IEEE Transactions on Robotics*, 40:936–959, 2023.

[2] Kasra Khosoussi and Iman Shames. Joint state and noise covariance estimation. *arXiv preprint arXiv:2502.04584*, 2025.

[3] Julia V Tsyganova and Maria V Kulikova. Svd-based kalman filter derivative computation. *IEEE Transactions on Automatic Control*, 62(9):4869–4875, 2017.

[4] Mohamad Qadri, Zachary Manchester, and Michael Kaess. Learning covariances for estimation with constrained bilevel optimization. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15951–15957. IEEE, 2024.

[5] Ben Liu, Tzu-Yuan Lin, Wei Zhang, and Maani Ghaffari. Debiasing 6-dof imu via hierarchical learning of continuous bias dynamics. *arXiv preprint arXiv:2504.09495*, 2025.

[6] Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

[7] David J. Yoon, Haowei Zhang, Mona Gridseth, Hugues Thomas, and Timothy D. Barfoot. Unsupervised learning of lidar features for use ina probabilistic trajectory estimator. *IEEE Robotics and Automation Letters*, 6(2):2130–2138, 2021.

[8] Keenan Burnett, David J Yoon, Angela P Schoellig, and Timothy D Barfoot. Radar odometry combining probabilistic estimation and unsupervised feature learning. *arXiv preprint arXiv:2105.14152*, 2021.

[9] Humphrey Hu and George Kantor. Efficient automatic perception system parameter tuning on site without expert supervision. In *Conference on Robot Learning*, pages 57–66. PMLR, 2017.

[10] Colin Parellier, Axel Barrau, and Silvere Bonnabel. Speeding-up backpropagation of gradients through the kalman filter via closed-form expressions. *IEEE Transactions on Automatic Control*, 68(12):8171–8177, 2023.

[11] Shushuai Li, Christophe De Wagter, and Guido CHE de Croon. Unsupervised tuning of filter parameters without ground-truth applied to aerial robots. *IEEE Robotics and Automation Letters*, 4(4):4102–4107, 2019.

[12] Bingheng Wang, Zhengtian Ma, Shupeng Lai, and Lin Zhao. Neural moving horizon estimation for robust flight control. *IEEE Transactions on Robotics*, 2023.

[13] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

[14] Timothy D Barfoot and Paul T Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[15] Keenan Burnett, Angela P Schoellig, and Timothy D Barfoot. Imu as an input vs. a measurement of the state in inertial-aided state estimation. *arXiv preprint arXiv:2403.05968*, 2024.

[16] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

[17] Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, et al. Fast-livo2: Fast, direct lidar-inertial-visual odometry. *IEEE Transactions on Robotics*, 2024.

[18] Ross Hartley, Maani Ghaffari, Ryan M Eustice, and Jessy W Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation. *The International Journal of Robotics Research*, 39(4):402–430, 2020.

[19] Axel Barrau and Silvere Bonnabel. The invariant extended kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812, 2016.

[20] Humphrey Hu and George Kantor. Introspective evaluation of perception performance for parameter tuning without ground truth. In *Robotics: Science and Systems*, 2017.

[21] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3d slam: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4597–4604. IEEE, 2015.