IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Tightly-Coupled LiDAR-IMU-Leg Odometry with Online Learned Leg Kinematics Incorporating Foot Tactile Information

Taku Okawara[1,2], Kenji Koide[2], Aoki Takanose[2], Shuji Oishi[2], Masashi Yokozuka[2], Kentaro Uno[1], and Kazuya Yoshida[1]

***Abstract*—In this letter, we present tightly coupled LiDAR-IMU-leg odometry, which is robust to challenging conditions such as featureless environments and deformable terrains. We developed an online learning-based leg kinematics model named the *neural leg kinematics model*, which incorporates tactile information (foot reaction force) to implicitly express the nonlinear dynamics between robot feet and the ground. Online training of this model enhances its adaptability to weight load changes of a robot (e.g., assuming delivery or transportation tasks) and terrain conditions. According to the *neural adaptive leg odometry factor* and online uncertainty estimation of the leg kinematics model-based motion predictions, we jointly solve online training of this kinematics model and odometry estimation on a unified factor graph to retain the consistency of both. The proposed method was verified through real experiments using a quadruped robot in two challenging situations: 1) a sandy beach, representing an extremely featureless area with a deformable terrain, and 2) a campus, including multiple featureless areas and terrain types of asphalt, gravel (deformable terrain), and grass. Experimental results showed that our odometry estimation incorporating the *neural leg kinematics model* outperforms state-of-the-art works. Our project page is available for further details: https://takuokawara.github.io/RAL2025_project_page/**

***Index Terms*—Deep Learning based on factor graph, SLAM, Sensor fusion, Leg odometry with tactile information.**

I. Introduction

LEGGED robots have significant potential for transportation and inspection tasks in challenging environments (e.g., rough terrain) thanks to their superior locomotion capabilities compared to wheeled robots. Accurate and robust odometry estimation is crucial for the reliable navigation required for these tasks. Although state-of-the-art LiDAR-IMU odometry is accurate thanks to the tight coupling of LiDAR and IMU constraints [1], these methods fail in featureless environments (e.g., vast flatlands such as lunar surfaces, tunnels), where LiDAR point clouds degenerate [2]. IMUs can provide information on relative sensor motion by integrating their measurements independently of geometric features to mitigate estimation drift in such situations. However, IMU-based constraints are unstable in the presence of long-term point cloud degeneration because integration errors (especially double integration of linear acceleration) accumulate rapidly due to IMU measurement noise. For legged robots, robot motion can be estimated based on joint motions (leg forward kinematics), namely leg odometry [3], [4]. For translational elements, leg forward kinematics-based constraints are more reliable than IMU-based constraints because leg kinematics-based motion prediction requires a single integration, which accumulates errors more slowly compared to double integration. However, conventional leg odometry algorithms assume that the foot velocity with respect to the ground is zero (no foot slippage); therefore, this algorithm is also unstable under conditions such as deformable terrain and foot slippage.

To cope with challenging situations such as featureless environments and deformable terrain, we developed an odometry estimation algorithm that fuses LiDAR-IMU constraints and leg kinematics constraints in a tightly coupled way. To address such situations where the assumptions of the conventional leg odometry algorithms become invalid, we incorporated foot

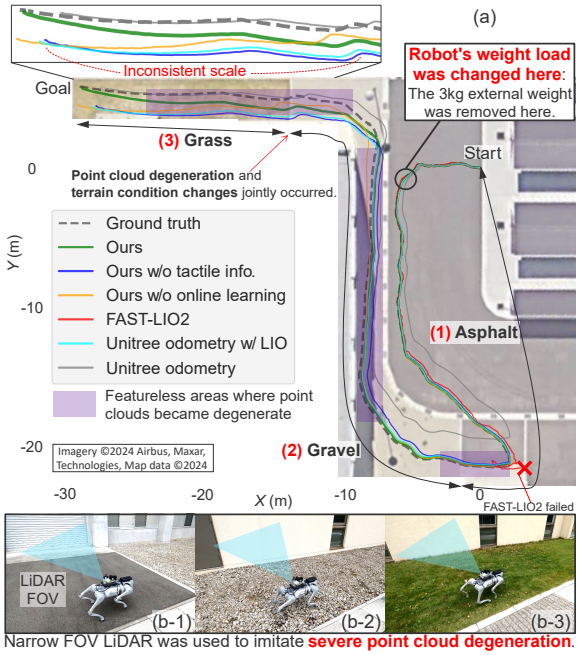

Fig. 1: (a) Odometry estimation results on the campus including terrain condition changes and featureless areas (b-1, b-2, and b-3). To validate the adaptability of our online learning-based kinematics model (*neural leg kinematics model*) for changes in the robot's weight load, the 3 kg external weight was removed in the middle of this experiment.

Manuscript received: February 4, 2025; Revised: April 29, 2025; Accepted: May 29, 2025.

This paper was recommended for publication by Editor Olivier Stasse upon evaluation of the Associate Editor and Reviewers' comments.

*This work was supported in part by and KAKENHI Grant Number 22KJ0292, a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

[1]Authors are with Space Robotics Lab., Dept. Aerospace Eng., Tohoku University, Japan, taku.okawara@aist.go.jp

[2]Authors are with the Dept. of Information Technology and Human Factors, the National Institute of Advanced Industrial Science and Technology, Japan

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

tactile information (reaction force, contact state) into the leg kinematics model to directly express the dynamic interaction between feet and the ground. The reaction forces of feet vary depending on terrain characteristics such as soil parameters, friction coefficients, and foot sinkage [5]. However, explicitly accurate identification of these parameters is difficult because the interaction model is complex and nonlinear. To effectively incorporate tactile information into leg kinematics, we applied a neural network for implicitly expressing the dynamics models that are difficult to model rigorously. We designed the network to be trained online to adapt to changes in the robot's weight load and the terrain types because the foot reaction force varies with both. Moreover, we jointly performed odometry estimation and online training of our network on a unified factor graph to retain both consistency, based on the proposed factor named the *neural adaptive leg odometry factor*.

We extended related work [2], which developed an online learning-based wheel kinematics model without tactile information (e.g., reaction forces or torques applied to the wheels or their axes), with the following contributions:

1) We developed the *neural leg kinematics model*, which incorporates tactile information (foot reaction force) into the leg kinematics. This model is expressed with an online-trained network to consider its adaptability to the robot's weight load and terrain conditions.
2) We proposed the *neural adaptive leg odometry factor* to simultaneously perform odometry estimation and online training of the neural leg kinematics model on a unified factor graph to maintain their consistency.
3) We explicitly estimated the uncertainty (i.e., covariance matrix) of this leg kinematics-based motion constraint online to create a reasonable constraint in challenging conditions (e.g., deformable terrain).

II. Related works

A. Fundamental leg odometry algorithms

Traditional leg odometry algorithms estimate robot velocity based on joint motions (leg forward kinematics) under ideal conditions where the terrain is flat and non-deformed, and none of the feet slip. The robot velocity $\boldsymbol{v}_b$ caused by the j-th leg motion is defined as follows [3], [4]:

$$\boldsymbol{v}_b = -(\boldsymbol{J}_j\ \dot{\boldsymbol{\theta}}_j + \lfloor \boldsymbol{\omega} \rfloor^{\times} \mathrm{fk}(\boldsymbol{\theta}_j)), \tag{1}$$

where $\mathrm{fk}(\boldsymbol{\theta}_j)$ is an operation to output the j-th foot position described in the robot frame through forward kinematics with its leg joint angles $\boldsymbol{\theta}_j$, $\dot{\boldsymbol{\theta}}_j$ is the j-th leg joint angular velocities, $\boldsymbol{J}_j$ is the Jacobian matrix related to the j-th foot position and its joint angles, and $\lfloor \boldsymbol{\omega} \rfloor^{\times}$ is the skew-symmetric matrix for the angular velocity of the robot body. To estimate $\boldsymbol{v}_b$ for robots with an arbitrary number of legs (e.g., quadrupeds or hexapods) making multiple contacts, the average linear velocity of only the feet in contact with the ground is used. Although this leg forward kinematics-based motion prediction is reasonable under the aforementioned ideal conditions, this model is fragile under conditions with terrain deformation and foot slippage. Michael et al. proposed an EKF-based approach with leg forward kinematics and IMU [6] to cope with such challenging situations by constraining each contact foot position with a random walk model with zero-mean Gaussian noise.

B. Proprioceptive and exteroceptive sensor fusion-based odometry estimation for legged robots

Fusing exteroceptive sensors (e.g., camera, LiDAR) and proprioceptive sensors is crucial for enhancing robustness to featureless environments and reducing estimation drift. Loosely coupled [7] and tightly coupled [8] visual-IMU-leg odometry based on factor graph optimization have been proposed for robustness in featureless environments with dynamic objects. Although the above approaches successfully incorporated leg forward kinematics into exteroceptive sensor-based estimation algorithms, they did not account for errors caused by terrain-dependent phenomena (foot slippage and deformations of legs and/or the ground); thus, accurate odometry estimation is difficult in such challenging situations. To cope with such errors, David et al. explicitly estimated the translational velocity bias of Eq. (1) based on a simple linear model in addition to odometry estimation [3].

Errors in leg forward kinematics significantly depend on inaccuracies in kinematic parameters (e.g., link deformation and machining errors). In particular, a foot tip made of rubber can deform upon impact when the foot contacts hard ground; thus, the kinematic parameters of legged robots can change dynamically. Yang et al. proposed performing online calibration of kinematic parameters and odometry estimation simultaneously based on a Kalman filter [4].

Although these approaches demonstrated accurate odometry estimation in featureless areas, their works do not assume more challenging situations, where severely featureless areas and terrain condition changes are jointly included.

C. Leg odometry model with tactile information

Tactile information in legged robots includes elements such as a foot reaction force and a binary contact state (whether the foot is raised or contacting the ground), which can be measured using force sensors, joint torques, or touch sensors. Most approaches [3], [8] use these sensors to detect foot contact for leg kinematics-based motion predictions because only the feet contacting the ground influence the robot's movement through leg kinematics. Fourmy et al. demonstrated that the foot contact force can be used for estimating the displacement of the body of a legged robot [9]. Specifically, the foot contact force is divided by the robot mass to estimate the acceleration caused by the force. Finally, the velocity and displacement are estimated by integrating the acceleration; thus, these values are used as measurements for the robot motion. Kang et al. extended Fourmy's work to jointly perform odometry and external force estimation on a unified factor graph [10].

Although these studies [9], [10] explicitly regard the robot's mass as being constant, the mass can vary depending on robot applications (e.g., transportation tasks or adding devices for augmenting the robot's capability).

Unlike the aforementioned studies, our leg kinematics model can adapt to changes in the robot's weight-load conditions, thanks to online training of our learning model.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

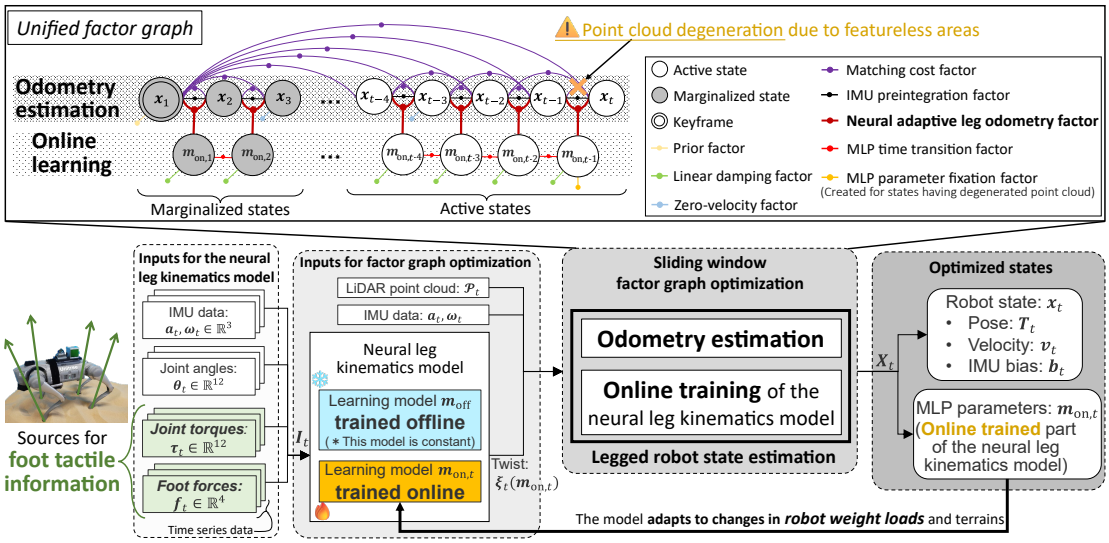

Fig. 2: Overview of the proposed framework. To accomplish reliable legged robot state estimation, we jointly perform LiDAR-IMU-leg odometry and online training of the *neural leg kinematics model*, on a unified factor graph. In contrast to [2], the proposed network infers the robot twist $\boldsymbol{\xi}_t$ by incorporating foot tactile information in addition to proprioceptive sensor data.

D. Learning-based kinematic model for legged robots

In contrast to the aforementioned approaches (Section II-A,II-B,II-C) using model-based leg forward kinematics, learning-based leg kinematics models [11], [12] have been developed to better express terrain-dependently nonlinear dynamics such as foot slippage and deformation of legs and/or the ground. These studies demonstrated robust and accurate odometry estimation using their learning-based kinematic model and exteroceptive sensor constraints. A particularly relevant study is [13], which proposes an invariant EKF-based odometry estimation based on IMU measurements and learning-based leg forward kinematics incorporating tactile information (foot reaction force). Their learning model only outputs the weights (i.e., reliability) of each foot in multiple contacts through input data of foot force sensor values. They only used these weights to calculate a weighted average of each leg forward kinematics-based motion for more accurate odometry estimation instead of the average described in Section II-A. They verified that the odometry estimation with learning-based leg kinematics incorporating tactile information outperformed the conventional algorithms, though their evaluation was limited to indoor flat floors where rigid foot contact is guaranteed, unlike our work. We consider that their weights (reliability) do not work properly on deformable terrains (e.g., gravel or sandy beaches), where rigid foot contact cannot be guaranteed at any time.

These approaches [11]–[13] trained their learning models offline; thus, adaptability to variations in the robot's weight load and terrain conditions was not considered.

III. Methodology Overview

A. System overview

As shown in Fig. 2, we jointly perform odometry estimation and online training of the *neural leg kinematics model* on a unified factor graph to maintain the consistency of all constraints. Online training of this model enhances adaptability to the weight loads of a legged robot and terrain conditions, enabling effective use of foot tactile information (reaction forces) for motion prediction because foot reaction forces vary with both robot weight loads and terrain types. To balance accuracy and computational costs for training the network, we divided the model into two models trained *online* and *offline* [2]. The model trained offline is constant during odometry estimation. The robot kinematic state $\boldsymbol{X}_t$ to be estimated at time t is defined as follows:

$$\boldsymbol{X}_t = [\boldsymbol{x}_t, \boldsymbol{m}_{\text{on},t}], \tag{2}$$

$$\boldsymbol{x}_t = [\boldsymbol{T}_t, \boldsymbol{v}_t, \boldsymbol{b}_t], \tag{3}$$

where $\boldsymbol{x}_t$ is the robot state at time t; $\boldsymbol{T}_t = [\boldsymbol{R}_t|\boldsymbol{t}_t] \in SE(3)$ and $\boldsymbol{v}_t \in \mathbb{R}^3$ are the robot pose and linear velocity in the world frame, respectively; $\boldsymbol{b}_t = [\boldsymbol{b}_t^a, \boldsymbol{b}_t^\omega] \in \mathbb{R}^6$ is the bias of the IMU linear acceleration $\boldsymbol{a}_t$ and the angular velocity $\boldsymbol{\omega}_t$; and $\boldsymbol{m}_{\text{on},t}$ is the weight and bias of the MLP of the neural leg kinematics model trained online. $\boldsymbol{m}_{\text{on},t}$ is dynamically updated along with $\boldsymbol{x}_t$ with the assumption that the robot walks in a feature-rich environment at the beginning of the estimation. Once the online-trained network converges to a state adapted to the current terrain and robot weight load, the neural network-based motion constraint becomes reliable for optimization and enables accurate odometry estimation even in such severe conditions. Note that the proposed method constantly trains our network online along with odometry estimation, based on factor graph optimization to adapt to dynamic changes in weight load and terrain conditions.

B. Overview of neural leg kinematics model

Network input and output: We developed a neural network (*neural leg kinematics model*) that outputs the 6DOF twist (translational and rotational velocities in the tangent space of the robot body) with proprioceptive sensors (joint

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

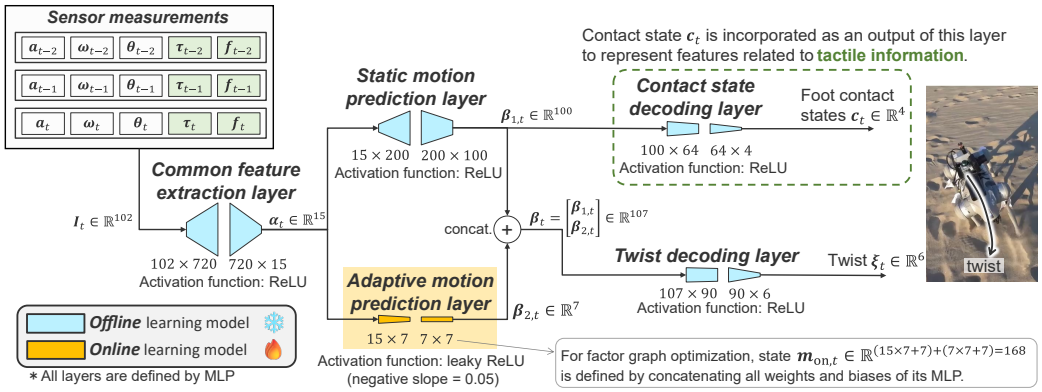

Fig. 3: Neural leg kinematics model structure. This network outputs a legged robot's twist $\boldsymbol{\xi}_t \in \mathbb{R}^6$ and the Boolean contact states for each foot $\boldsymbol{c}_t \in \mathbb{R}^4$ by inputting $\boldsymbol{I}_t$. The dimension of the online learning model's MLP parameter $\boldsymbol{m}_{\mathrm{on},t}$ is 168.

encoder and IMU) and tactile information (foot reaction force). The input data for the network are time series data consisting of IMU data (linear acceleration $\boldsymbol{a}_t \in \mathbb{R}^3$, angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$), joint angles $\boldsymbol{\theta}_t \in \mathbb{R}^{12}$, joint torques $\boldsymbol{\tau}_t \in \mathbb{R}^{12}$, and foot force sensor values $\boldsymbol{f}_t = [{}^{\mathrm{LF}}f_t \;\; {}^{\mathrm{LH}}f_t \;\; {}^{\mathrm{RH}}f_t \;\; {}^{\mathrm{RF}}f_t] \in \mathbb{R}^4$. We define $\boldsymbol{i}_t \in \mathbb{R}^{34}$ by concatenating the above sensor data as follows:

$$\boldsymbol{i}_t = [\boldsymbol{a}_t^\top \;\; \boldsymbol{\omega}_t^\top \;\; \boldsymbol{\theta}_t^\top \;\; \boldsymbol{\tau}_t^\top \;\; \boldsymbol{f}_t^\top]^\top. \tag{4}$$

We concatenate $\boldsymbol{i}_t$ to create the time series data of sensor values with a window size N_{w}. The model can implicitly represent various physical quantities (e.g., joint angular velocity based on the difference of consecutive time of joint angles) by using the time series data as inputs. Finally, the input for the network is defined as $\boldsymbol{I}_t = \rho([\boldsymbol{i}_t, \boldsymbol{i}_{t-1}, ...\boldsymbol{i}_{t-N_{\mathrm{w}}-1}])$, where ρ is a standardization operation. In our implementation, we set N_{w} to 3; thus, the dimension of $\boldsymbol{I}_t$ is 102 (34×3).

We propose incorporating tactile information into leg kinematics models to enable robust motion prediction for terrain-dependent phenomena (e.g., terrain deformation, foot slippage). Foot tactile information (reaction forces) can address these severe conditions because the reaction forces are influenced by foot slippage (friction forces) and terrain deformation [5]. In other words, foot reaction forces contain information on terrain-dependent features. As stated in Section II-C, 3D foot reaction forces can also be used to express a robot's velocity. Because 3D foot reaction force measurements are not directly available on our robot platform (Unitree Go2), joint torques are used as the input data instead. The reason is that foot reaction forces can be expressed based on the motion equation for legged robots using joint torques [10]. Because the accuracy of motion-equation-based reaction force expression depends on a precise model identification for such as unknown disturbances and robot's mass, inertia moment [10], learning-based approaches are preferred to accurately express such complex terms by implicit representation [14]. In addition, a 1D foot force sensor (Unitree Foot Pad) is used to complement the information on the 3D reaction force represented by the joint torque.

Network structure: As shown in Fig. 3, the network structure is divided into an *offline learning model* and *online learning model* based on related work [2], which developed the online trained wheel odometry model without force or torque of wheels. Our online learning model expresses dynamic features regarding terrains and legged robots' weight loads. Whereas, our offline learning model captures features invariant to weight and terrain, such as basic leg forward kinematics and inertial propagation. We designed the above network structure to balance the computational cost and accuracy for realizing reasonable online learning. The size of the online learning model is smaller than that of the offline learning model because features invariant to weight loads and terrains have a greater influence than dynamic features varying with these conditions in leg kinematics models.

First, the *common feature extraction layer* takes $\boldsymbol{I}_t$ as input and extracts common features $\boldsymbol{\alpha}_t$ related to motion prediction. The subsequent *static* and *adaptive motion prediction layers* take these common features $\boldsymbol{\alpha}_t$ as input and output feature vectors $\boldsymbol{\beta}_{1,t}$ and $\boldsymbol{\beta}_{2,t}$ to predict the robot twist in a latent feature space, respectively. $\boldsymbol{\beta}_{1,t}$ also includes features related to foot contact states to account for foot contact effects in our learning model. The *contact state decoding layer* interprets $\boldsymbol{\beta}_{1,t}$ to determine whether each foot is in contact with the ground. The contact state $\boldsymbol{c}_t$ is defined on a scale from 0 to 1, where 0 and 1 indicate non-contact and contact, respectively. Note that $\boldsymbol{c}_t$ is incorporated as an output of this model to represent features related to tactile information; however, it is not directly used for online learning (factor graph optimization). Finally, the *twist decoding layer* decodes the latent features $\boldsymbol{\beta}_t$ into the twist in the robot body $\boldsymbol{\xi}_t$.

IV. Training the Offline Learning Model

Training procedure: The offline learning model is trained to describe features invariant to the robot's weight loads and terrains for accurate motion prediction. The offline batch learning is conducted under the condition that, while the online learning model (adaptive motion prediction layer) varies depending on the robot's weight loads and terrain, the offline learning model is common for these conditions. By including diverse environments and the robot's weight load configurations in datasets for the offline learning process (Fig. 4), we ensure that the offline learning model provides the invariant

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

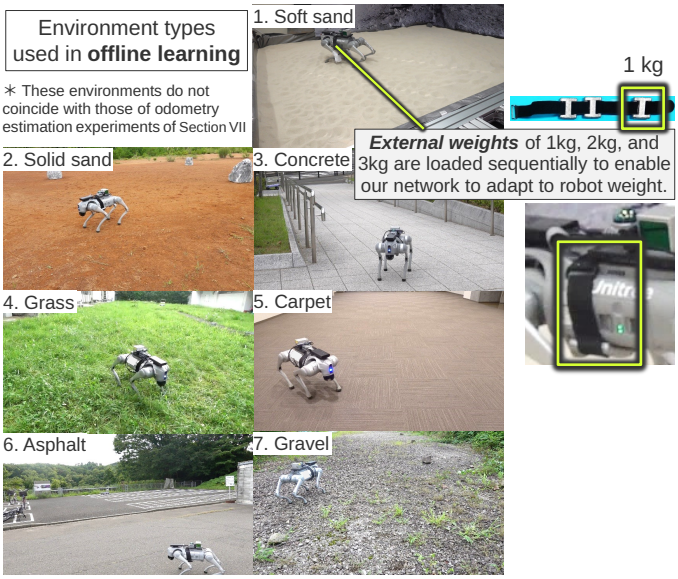

Fig. 4: Datasets of each terrain for offline batch learning.

features that can complement the adaptive responses of the online learning model under varying conditions. This structure aims to improve the performance and adaptability of our model across different terrains and payload scenarios. To meet the aforementioned training conditions, we minimize the overall loss function $\mathcal{L}$ (Eq. (5)), which is defined by the sum of the loss function $\mathcal{L}_n(\boldsymbol{m}_{\text{off}}, \boldsymbol{m}_{\text{on},n})$ for each dataset. Offline batch learning is conducted by minimizing $\mathcal{L}$ such that the MLP parameters for the offline learning model $\boldsymbol{m}_{\text{off}}$ are common for all datasets. In contrast, the MLP parameters for the online learning model $\boldsymbol{m}_{\text{on},n}$ vary depending on the type of dataset.

$$\mathcal{L} = \sum_{n=1}^{N_{\text{s}}} \mathcal{L}_n(\boldsymbol{m}_{\text{off}}, \boldsymbol{m}_{\text{on},n}), \tag{5}$$

$$\mathcal{L}_n(\boldsymbol{m}_{\text{off}}, \boldsymbol{m}_{\text{on},n}) = e_{t,\boldsymbol{\xi}} + w_1 e_{t,\boldsymbol{c}} + w_2 e_{t,\text{r}}, \tag{6}$$

$$e_{t,\boldsymbol{\xi}} = \frac{1}{n_{\text{T}}} \sum_{t=1}^{n_{\text{T}}} (e_{t,\text{trans}} + w_3 e_{t,\text{rot}}). \tag{7}$$

$e_{t,\boldsymbol{\xi}}$, $e_{t,\boldsymbol{c}}$, and $e_{t,\text{r}}$ are the loss for the twist (mean square error), contact states of each foot (binary cross entropy), and L_2 regularization terms for features $\boldsymbol{\beta}_{2,t}$ of the online learning model, respectively. n_{T} is the number of training data in each dataset; $e_{t,\text{trans}}$ and $e_{t,\text{rot}}$ are the square error in the translation and rotation components of the twist $\boldsymbol{\xi}_t$, respectively. w_1 (e.g., 5), w_2 (e.g., 10^{-3}), and w_3 (e.g., 200) are weights adjusting the scales of the loss values.

Offline training details: We used N_{s} sequences including N_{e} environment types and N_{m} robot weight loads for the offline training. N_{e} and N_{m} were set to 7 and 4, respectively, resulting in $N_{\text{s}} = 28$. We changed the weight load by equipping the robot with external weights of 1 kg, 2 kg, and 3 kg. The duration of each sequence was about 10 minutes. As shown in Fig. 4, we included a variety of environments, such as deformable and solid terrains as well as slopes.

We used the Adam optimizer [15] with a 5×10^{-4} learning rate, 2500 epochs, and a batch size of 50 in our implementation. The number of layers and activation functions for each MLP are shown in Fig. 3. We determined the reference of contact states $\boldsymbol{c}_t$ by offline analysis based on thresholding of each foot's force sensor values, similar to related work [3]. We used LiDAR-IMU odometry with an omnidirectional FOV LiDAR (Livox MID-360) to obtain reference data for the twist $\boldsymbol{\xi}_{t,\text{L}}$. We calculate $e_{t,\text{trans}}$ and $e_{t,\text{rot}}$ in Eq. (7) based on the error between $\boldsymbol{\xi}_{t,\text{L}}$ and $\boldsymbol{\xi}_t$ inferred by the network. LiDAR-IMU odometry is highly accurate in feature-rich environments (e.g., our datasets in Fig. 4). The robot also moved locally in each environment to always remain on the local map; thus, accumulation errors did not occur in LiDAR-IMU odometry for the offline training sequences. We manually operated the legged robot (Unitree Go2) through various motions (e.g., straight and curved trajectories on slopes and steps).

TABLE I: Translational and rotational RTEs per 1 m of the proposed network and ablation cases.

Method	t_{RTE} [m]	r_{RTE} [deg]
Proposed network (w/ tactile info., $w_3 = 200$)	**0.06** $\pm$ 0.04	**1.00** $\pm$ 0.69
Network w/o tactile information	0.10 $\pm$ 0.07	1.06 $\pm$ 0.69
Network w/ small rotational weight ($w_3 = 1$)	0.07 $\pm$ 0.04	3.36 $\pm$ 2.87

Network evaluations: To demonstrate the importance of incorporating tactile information and assigning a large rotational weight (e.g., $w_3 = 200$) in the loss function Eq. (7), we compared the proposed network with 1) *network w/o tactile information*, and 2) *network w/ small rotational weight* ($w_3 = 1$). Table I shows the average of relative trajectory errors (RTEs) per 1 m of the proposed network and the other networks using 10% of the offline training datasets (Fig. 4) as validation data. For translational RTEs t_{RTE}, the proposed network (0.06 m) is approximately twice as accurate as the network without tactile information (0.10 m). Regarding rotational RTEs r_{RTE}, the proposed network (1.00 deg) is three times more accurate than the network with a small rotational weight (3.36 deg). The result demonstrated 1) the importance of tactile information for accurate motion prediction in challenging terrains and 2) the effectiveness of the large rotational weight w_3 for loss Eq. (7) in the proposed network.

V. Training Online Learning Model

A. Neural adaptive leg odometry factor

We train the online learning model (adaptive motion prediction layer) on-the-fly with the fixed offline learning model. To perform online training of the network and odometry estimation on a unified factor graph, we propose the *neural adaptive leg odometry factor*, which accounts for both constraints regarding robot motion and online learning. In the online learning phase, the parameter vector $\boldsymbol{m}_{\text{on}}$ concatenating the weights and biases of MLPs for the online learning model is optimized such that online trained leg kinematics-based, LiDAR-based, and IMU-based constraints become consistent.

To derive the error in the neural adaptive leg odometry factor, we first calculate the relative pose by integrating the corresponding twist $\boldsymbol{\xi}_i$ of the robot with a time step Δt_i. Then, we map the relative pose from $se(3)$ to $SE(3)$ space based

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

on exponential map operation (i.e., $\exp(\boldsymbol{\xi}_i \Delta t_i)$). Finally, the error in this factor is defined as follows:

$$e^{\mathrm{Leg}}(\boldsymbol{T}_{i-1}, \boldsymbol{T}_i, \boldsymbol{m}_{\mathrm{on},i}) = (\boldsymbol{r}_i^{\mathrm{Leg}})^\top (\boldsymbol{C}_i^{\mathrm{Leg}})^{-1} \boldsymbol{r}_i^{\mathrm{Leg}}, \tag{8}$$

$$\boldsymbol{r}_i^{\mathrm{Leg}} = \log(\boldsymbol{T}_{i-1}^{-1} \boldsymbol{T}_i \exp(\boldsymbol{\xi}_i \Delta t_i)^{-1}), \tag{9}$$

$$\boldsymbol{\xi}_i = \mathrm{NN}(\boldsymbol{m}_{\mathrm{on},i}), \tag{10}$$

where $(\boldsymbol{C}_i^{\mathrm{Leg}})^{-1}$ is the covariance matrix representing the uncertainty in the network-based motion prediction. NN is an operation to output $\boldsymbol{\xi}_i$ through the neural leg kinematics model, which incorporates the online optimized MLP parameters $\boldsymbol{m}_{\mathrm{on},i}$, fixed offline trained model $\boldsymbol{m}_{\mathrm{off}}$, and the model input (i.e., time series sensor data $\boldsymbol{I}_i$), as seen in Fig. 2. We compute the Jacobian matrix related to $\boldsymbol{r}_i^{\mathrm{Leg}}$ and $\boldsymbol{m}_{\mathrm{on},i}$ by using the LibTorch library to take advantage of automatic differentiation wrapped the results with a GTSAM nonlinear factor class.

B. Online uncertainty estimation for neural adaptive leg odometry factor

We explicitly estimate the uncertainty (i.e., $\boldsymbol{C}_i^{\mathrm{Leg}}$) associated with the *neural adaptive leg odometry factor* online, ensuring that this constraint remains reasonable for challenging terrains (e.g., rough and deformable surfaces). We estimate $\boldsymbol{C}_i^{\mathrm{Leg}}$ by analyzing $\boldsymbol{r}_i^{\mathrm{Leg}}$ (Eq. (9)), which is the residual of the network-based motion prediction. N (e.g., 15) samples of $\boldsymbol{r}_i^{\mathrm{Leg}}$ (i.e., $\boldsymbol{r}_i^{\mathrm{Leg}}$,...,$\boldsymbol{r}_{i-N}^{\mathrm{Leg}}$) are used to properly capture the current terrain condition with the assumption that $\boldsymbol{m}_{\mathrm{on},t}$ does not change drastically across the N samples. The sliding window adjusts a balance between stability and responsiveness for estimating the reasonable uncertainty. According to the definition of variance, for example, the variance of translational X element ${\sigma_{i,\mathrm{Tx}}}^2$ is estimated as follows:

$${\sigma_{i,\mathrm{Tx}}}^2 = \frac{1}{N-1} \sum_{j=1}^{N} r_{i,\mathrm{Tx},j}^2. \tag{11}$$

The other five variances can be estimated similarly to that for the translational X element. When we assume that each element of $\boldsymbol{r}_i^{\mathrm{Leg}}$ is independent, the covariance matrix $\boldsymbol{C}_i^{\mathrm{Leg}}$ can be obtained by setting these variances to the diagonal elements of $\boldsymbol{C}_i^{\mathrm{Leg}}$. In addition, we assume that the estimation of poses is accurate in elements where point clouds do not degenerate; thus, we consider $\boldsymbol{r}_i^{\mathrm{Leg}}$ for the corresponding elements to be reliable for estimating variances.

VI. Implementation Details

The objective function (i.e., factor graph of Fig. 2) for the optimization is the sum of the LiDAR-based (matching cost factor [16]), IMU-based (IMU pre-integration factor [17]), leg kinematics model-based constraints, and others (MLP time transition factor, MLP parameter fixation factor [2], and prior factor). Regarding factors other than the neural adaptive leg odometry factor, overviews and error functions are described in [2]. We implemented the factor graph using GTSAM and incrementally optimized this graph with iSAM2 [18]. We used a fixed lag smoother to ensure real-time optimization within a time window (e.g., 6 s). Only active states in the time window are optimized, while marginalized states are fixed.

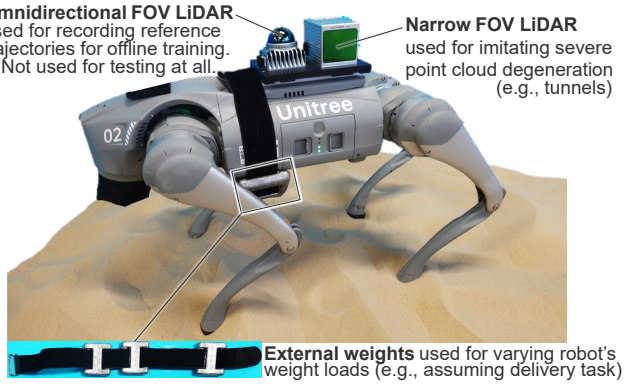

Fig. 5: Unitree Go2 used as a testbed. Note that the omnidirectional LiDAR was used for recording reference trajectories for offline training, and experiments (Fig. 1, Fig. 6) were conducted using only the narrow FOV LiDAR to imitate severe point cloud degeneration (e.g., tunnels and lunar surfaces).

VII. Experimental Results

Experimental conditions: We conducted experiments with a Unitree Go2 (Fig. 5) equipped with a narrow FOV LiDAR (Livox AVIA) and external weights as a testbed to verify the proposed odometry algorithm. We recorded point clouds, IMU data, joint angles, joint torques, and foot forces at frequencies of 10 Hz, 60 Hz, 500 Hz, 500 Hz, and 500 Hz, respectively, to estimate robot states $\boldsymbol{x}_t$ and MLP parameters $\boldsymbol{m}_{\mathrm{on},t}$. We obtained the ground truth trajectories using a total station (Leica TS16). We compared the proposed method (ours) with the following baseline methods:

- *Ours w/o online learning*: We created an ablative method by incorporating the fixed neural network-based motion constraints into the factor graph instead of the *neural adaptive leg odometry factor*, like related works [11], [12]. The network was trained by simple offline batch learning with all datasets (Fig. 4); thus, the *adaptive motion prediction layer* (Fig. 3) was not trained online.
- *Ours w/o tactile information*: A network without tactile information (joint torques $\boldsymbol{\tau}_t \in \mathbb{R}^{12}$ and foot force sensor values $\boldsymbol{f}_t \in \mathbb{R}^4$) was used as a second ablative method to create motion constraints instead of the neural adaptive leg odometry factor, similar to [2], [3].
- *FAST-LIO2*: FAST-LIO2 is state-of-the-art odometry based on tightly-coupled LiDAR-IMU constraints [1].
- *Unitree odometry*: Go2 outputs proprioceptive sensor-based odometry by a Unitree proprietary algorithm.
- *Unitree odometry w/ LIO*: Matching cost factor [16], IMU pre-integration factor [17], and the above Unitree odometry algorithm-based motion constraint are incorporated into a factor graph for reasonable comparison.

Accuracy evaluation: We conducted experiments in two situations: 1) a campus having featureless areas (Fig. 1(b-1,2,3)) and terrain condition changes, and 2) a sandy beach that included severely featureless areas (Fig. 6(b)). Note that these environments were not included in the offline training datasets (Fig. 4). To demonstrate robustness against rapid movements, we manually operated the testbed at its maximum walking

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II: ATEs and RTEs of the odometry algorithms.

Method/Sequence	Campus		Sandy beach	
	ATE [m]	RTE [m]	ATE [m]	RTE [m]
Ours	**0.29** ± 0.12	**0.13** ± 0.04	**0.08** ± 0.05	**0.12** ± 0.04
Ours w/o online learning	0.36 ± 0.17	0.17 ± 0.07	0.90 ± 0.70	0.20 ± 0.10
Ours w/o tactile info.	0.63 ± 0.30	0.15 ± 0.06	0.12 ± 0.06	0.12 ± 0.04
FAST-LIO2	Corrupted	Corrupted	Corrupted	Corrupted
Unitree odometry w/ LIO	0.57 ± 0.32	0.15 ± 0.06	No record	No record
Unitree odometry	0.80 ± 0.46	0.17 ± 0.06	No record	No record

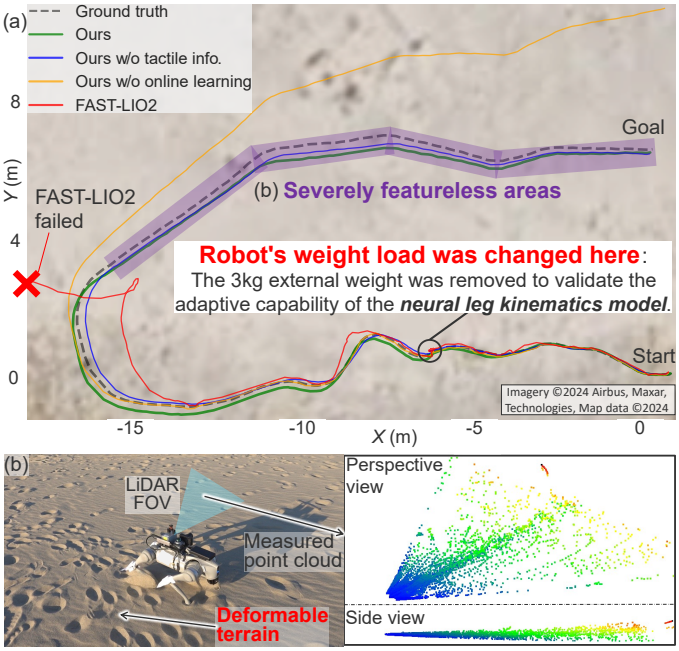

Fig. 6: (a) Odometry estimation results in the sandy beach, including (b) severely featureless areas.

speed (approximately $2\,\mathrm{m/s}$), and the speed of the robot naturally varied due to acceleration, deceleration, and ground surface changes. To initialize the online learning model, we used $\boldsymbol{m}_{\mathrm{on}}$ obtained from the offline learning phase under the condition where the terrain type was grass and no additional weight was applied, to show the adaptability of online learning. We consider that the best initial values of this parameter can be set by combining terrain classification algorithms if the proposed method is deployed in a real operation.

Fig. 1(a) and Fig. 6(a) show the trajectory comparison results. Table II also shows the absolute trajectory errors (ATEs) and relative trajectory errors (RTEs) [19] for all methods. According to these trajectory estimation results and our method's ATEs (campus: $0.29\,\mathrm{m}$; sandy beach: $0.08\,\mathrm{m}$) and RTEs (campus: $0.13\,\mathrm{m}$; sandy beach: $0.12\,\mathrm{m}$), our method outperformed all others because it 1) incorporates tactile information into the leg kinematics model, and 2) adapts to changes in terrain and robot weight load based on online learning. *FAST-LIO2* failed due to the featureless areas (e.g., Fig. 1(b-1) and Fig. 6(b)). In *ours w/o online learning*, the translational scales were particularly incorrect on the grass of the campus (Fig. 1(a)) and the sandy beach (Fig. 6(a)). We consider that this learning model cannot adapt to changes in the robot's weight load. Specifically, the robot's acceleration and velocity, expressed by dividing the reaction force by the robot's mass, are not properly described in this model. While *Unitree odometry* accurately estimated the rotation motions, its translational scales were incorrect, as shown in Fig. 1(a). Similarly, the trajectory of *Unitree odometry w/ LIO* (fusing the above Unitree odometry-based motion constraints and LiDAR-IMU constraints) has also large errors in the translational elements, as seen in Fig. 1(a).

Ours w/o tactile information achieves nearly the same accuracy (ATE: $0.12\,\mathrm{m}$; RTE: $0.12\,\mathrm{m}$) as the proposed method on the sandy beach, likely because the learning model without tactile information was sufficiently trained online to capture terrain-dependent features, similar to [2], [3] that implicitly correct terrain-dependent errors by exteroceptive sensors' constraints without tactile information. Whereas, in the campus sequence, the ATE for *ours w/o tactile information* ($0.63\,\mathrm{m}$) is approximately twice as large as that of the proposed method. Moreover, its translational scales are particularly inconsistent in the grass, as shown in Fig. 1(a). We consider the reason to be that point cloud degeneration and terrain condition changes jointly occurred when the robot transitioned from gravel to grass. In such conditions, while kinematics models without tactile information (e.g., *ours w/o tactile information*, related works [2], [3]) cannot properly describe terrain-dependent features, our model can partly represent them thanks to the inclusion of tactile information.

According to the results, we quantitatively demonstrated that incorporating online learning and tactile information in the neural leg kinematics model is effective for robust odometry estimation against challenging scenarios.

The proposed network performance: Next, we demonstrate the performance of the neural leg kinematics model independently (i.e., LiDAR-based and IMU-based factors are not incorporated). Fig. 7 presents the time histories of the motion errors and their moving averages for our networks trained online and offline, respectively, in the sandy beach sequence. The models trained online and offline correspond to the kinematics model used in *Ours* and *Ours w/o online learning* of Fig. 6, respectively. The online trained model shows a significant error reduction and outperforms the offline trained model thanks to its adaptability, particularly after approximately 40 seconds, when the robot's weight load was removed. While the errors slightly fluctuate due to terrain conditions and walking speed, the overall trend confirms that the proposed model adapts (converges) quickly and consistently improves its accuracy.

Time transition of the neural leg kinematics model via online learning: Finally, we demonstrate that the *neural leg kinematics model* was dynamically trained to adapt to terrain and the robot weight load conditions. We analyzed a time series of the MLP parameters $\boldsymbol{m}_{\mathrm{on}} \in \mathbb{R}^{168}$ (all weights and biases of online learning model) on the campus and the sandy beach sequence. To aid visualization, we embedded the time series of $\boldsymbol{m}_{\mathrm{on}}$ to 2 dimension vectors based on t-SNE [20]. Fig. 8 illustrates the 2D embedded parameter vectors. Note that Fig. 8 shows the embedded vectors for two types of sequences: sequence #1: walking on the campus and sandy beach (i.e., Fig. 1 and Fig. 6), and sequence #2: walking under the same conditions as sequence #1. In both sequences #1 and #2, we can

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

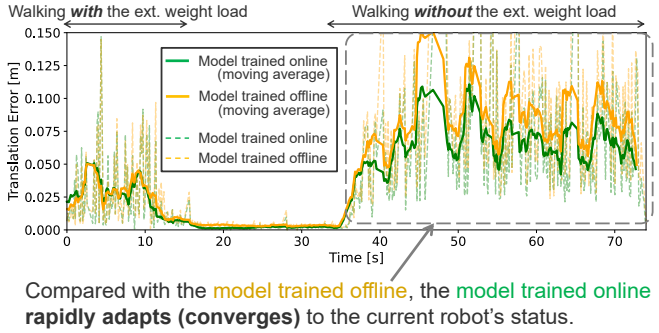

Fig. 7: Time histories of motion errors for our network without LiDAR and IMU factors, during the sandy beach (Fig. 6).

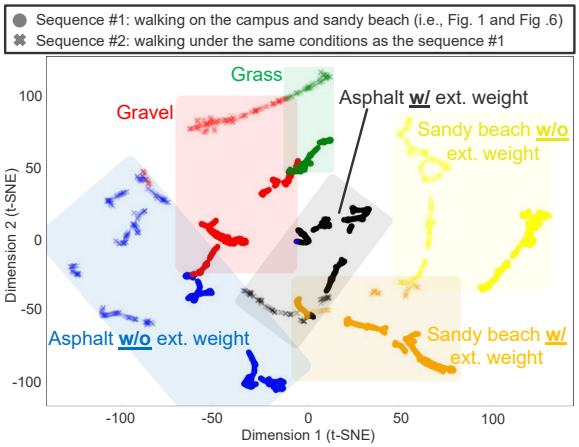

Fig. 8: t-SNE-based visualization of time histories of $\boldsymbol{m}_{\text{on}} \in \mathbb{R}^{168}$ (all weights and biases of online learning model) under the similar two sequences.

see that $\boldsymbol{m}_{\text{on}}$ for each sequence changed similarly to adapt to the current robot situation over time. This result indicates that our network was consistently trained online and converged to similar parameters across different sessions. Therefore, we demonstrated that the neural leg kinematics model successfully adapts to varying robot weight loads and terrain types.

VIII. Conclusion

We presented an odometry estimation algorithm that fuses LiDAR-IMU constraints and online trainable leg kinematic constraints incorporating tactile information (*neural leg kinematics model*) in a tightly coupled way. To effectively incorporate tactile information, the neural leg kinematics model was trained online to dynamically adapt the current robot's weight loads and terrain conditions. We demonstrated that our odometry estimation algorithm outperformed state-of-the-art approaches (Unitree proprietary odometry with LiDAR-IMU constraints, and FAST-LIO2) and ablation study cases (our learning model without tactile information or online learning) even under various conditions, including different terrain types (asphalt, gravel, grass, and sandy beach) and robot weights (3 kg changes), thanks to the *neural adaptive leg odometry factor* and its online uncertainty estimation.

In future work, we will extend our network to infer terrain classification, external forces acting on a robot's body and feet, in addition to the twist and foot contacts. This extension enables us to fully apply this network to legged robot control. We plan to extend our network into a more general model that accommodates various link parameters and release its code.

References

[1] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[2] T. Okawara, K. Koide, S. Oishi, M. Yokozuka, A. Banno, K. Uno, and K. Yoshida, "Tightly-coupled LiDAR-IMU-wheel odometry with an online neural kinematic model learning via factor graph optimization," *Robotics and Autonomous Systems*, vol. 187, no. 104929, 2025.

[3] D. Wisth, M. Camurri, and M. Fallon, "VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2022.

[4] S. Yang, H. Choset, and Z. Manchester, "Online kinematic calibration for legged robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8178–8185, 2022.

[5] A. Vanderkop, N. Kottege, T. Peynot, and P. Corke, "A novel model of interaction dynamics between legged robots and deformable terrain," in *International Conference on Robotics and Automation*, 2022, pp. 6635–6641.

[6] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots: Consistent fusion of leg kinematics and IMU," *Robotics: Science and Systems VIII*, pp. 17–24, 2013.

[7] R. Hartley, M. G. Jadidi, L. Gan, J.-K. Huang, J. W. Grizzle, and R. M. Eustice, "Hybrid contact preintegration for visual-inertial-contact state estimation using factor graphs," in *International Conference on Intelligent Robots and Systems*, 2018, pp. 3783–3790.

[8] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4507–4514, 2019.

[9] M. Fourmy, T. Flayols, P.-A. Léziart, N. Mansard, and J. Solà, "Contact forces preintegration for estimation in legged robotics using factor graphs," in *International Conference on Robotics and Automation*, 2021, pp. 1372–1378.

[10] J. Kang, H. Kim, and K.-S. Kim, "VIEW: Visual-inertial external wrench estimator for legged robot," *IEEE Robotics and Automation Letters*, pp. 8366–8377, 2023.

[11] R. Buchanan, M. Camurri, F. Dellaert, and M. Fallon, "Learning inertial odometry for dynamic legged robot state estimation," in *Conference on robot learning*, 2022, pp. 1575–1584.

[12] J. Wasserman, A. Agarwal, R. Jangir, G. Chowdhary, D. Pathak, and A. Gupta, "Legolas: Deep leg-inertial odometry," in *Conference on Robot Learning*, 2024.

[13] S. Yang, Q. Yang, R. Zhu, Z. Zhang, C. Li, and H. Liu, "State estimation of hydraulic quadruped robots using invariant-EKF and kinematics with neural networks," *Neural Computing and Applications*, vol. 36, no. 5, pp. 2231–2244, 2023.

[14] J. Hu and R. Xiong, "Contact force estimation for robot manipulator using semiparametric model and disturbance kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3365–3375, 2017.

[15] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Globally consistent 3D LiDAR mapping with GPU-accelerated GICP matching cost factors," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8591–8598, 2021.

[17] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[18] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[19] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *International Conference on Intelligent Robots and Systems*, 2018, pp. 7244–7251.

[20] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE." *Journal of machine learning research*, vol. 9, no. 11, pp. 2579–2605, 2008.