CoCo-InEKF: State Estimation with Learned Contact Covariances in Dynamic, Contact-Rich Scenarios

Michael Baumgartner*†, David Müller†, Agon Serifi†, Ruben Grandia†,
Espen Knoop†, Markus Gross*†, and Moritz Bächer†
*ETH Zurich, Switzerland, †Disney Research, Switzerland

***Abstract*—Robust state estimation for highly dynamic motion of legged robots remains challenging, especially in dynamic, contact-rich scenarios. Traditional approaches often rely on binary contact states that fail to capture the nuances of partial contact or directional slippage. This paper presents CoCo-InEKF, a differentiable invariant extended Kalman filter that utilizes continuous contact velocity covariances instead of binary contact states. These learned covariances allow the method to dynamically modulate contact confidence, accounting for more nuanced conditions ranging from firm contact to directional slippage or no contact. To predict these covariances for a set of predefined contact candidate points, we employ a lightweight neural network trained end-to-end using a state-error loss. This approach eliminates the need for heuristic ground-truth contact labels. In addition, we propose an automated contact candidate selection procedure and demonstrate that our method is insensitive to their exact placement. Experiments on a bipedal robot demonstrate a superior accuracy-efficiency tradeoff for linear velocity estimation, as well as improved filter consistency compared to baseline methods. This enables the robust execution of challenging motions, including dancing and complex ground interactions — both in simulation and in the real world.**

I. Introduction

Proprioceptive state estimation, i.e., estimating the robot's state without exteroceptive sensors such as cameras or LiDAR, remains a fundamental challenge. These types of estimators provide accurate, high-frequency state information for downstream feedback control during highly dynamic motions or in visually degraded environments. For legged robots, which are typically equipped with an inertial measurement unit (IMU) and actuator encoders that directly measure joint angles, the robot pose and linear velocity estimation requires sensor fusion.

A common strategy is to estimate the contact state of discrete points on the robot, and assume that points in contact remain stationary in the world frame [3, 9, 16]. The performance of these methods is critically dependent on accurate contact estimation and the validity of the stationarity assumption, motivating specialized methods for contact detection [19, 8] and handling of slipping contacts [36, 22, 20]. In particular, Lin et al. [26] proposed augmenting the state-of-the-art invariant extended Kalman filter (InEKF) [16] with learned contact detection. However, this approach requires labeled contact data for training and still treats contact as a binary state, determined independently of the state estimation.

To address these limitations, researchers have explored the use of end-to-end supervised learning for state estimation [21, 39]. Although this approach has enabled the successful deployment of impressive reinforcement learning controllers, these methods underperform when evaluated on state estimation accuracy alone, as we will show in our evaluation.

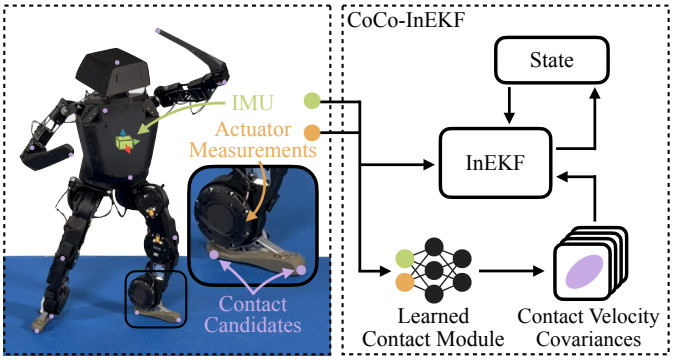

Fig. 1: **CoCo-InEKF.** Given a set of predefined contact candidates and the proprioceptive sensor data from the IMU and actuators of a robot, a learned contact module predicts contact velocity covariances for use in an Invariant EKF.

This paper proposes **Co**ntact**Co**variance**-InEKF**, a novel hybrid approach that combines the strengths of both the end-to-end and InEKF-based methods. We make the standard contact-aided InEKF differentiable by maintaining the state of all contact candidates, regardless of their current contact condition. Motivated by the observation that contact conditions, such as directional slippage, are richer than binary states, we introduce a contact module that predicts contact velocity covariances instead of binary contact states. This allows our model to dynamically modulate contact confidence, which is expressive enough to account for more nuanced states, spanning firm contact, directional slippage, and no contact. However, we lift strict physical enforcement, enabling the model to also build constraints that do not correspond to physical interaction. By applying backpropagation through time (BPTT) [14], we can then train the neural contact module end-to-end using simple state-error losses, avoiding the need for heuristic ground-truth contact labels required by previous methods. This architecture retains the invariance and observation properties of the InEKF and enables robust state estimation for complex, highly dynamic motions such as dancing and multi-contact ground interactions.

Succinctly, our contributions are:

- CoCo-InEKF, a proprioceptive state estimator combining a differentiable InEKF with a neural contact module that predicts contact velocity covariances.

- End-to-end training of the proposed architecture via backpropagation through time, avoiding the need for ground-truth contact information.
- Experimental validation of our method in highly dynamic and contact-rich scenarios, demonstrating improved performance over classical methods, InEKFs with learned contact classification, and end-to-end methods.

II. Related Work

A. Classical State Estimation

The pioneering work by Bloesch et al. [3] introduced an EKF that integrates inertial measurements with leg odometry. This approach treats contact points as stationary *landmarks* in global coordinates, using leg forward kinematics to measure their relative locations with respect to the robot's base. This idea has also been integrated into other filtering methods, such as the unscented Kalman filter [4], factor graphs [15], and a dual β-Kalman filter [41]. The InEKF introduced by Hartley et al. [16] leverages Lie group theory for superior convergence compared to quaternion-based filters. DRIFT [27] provides a modular, modern implementation of this approach.

Independent of the filtering approach, these methods fundamentally rely on an accurately-estimated contact state, and the non-slip assumption. In the absence of direct contact sensors, researchers have developed advanced contact detection methods using force thresholding [12], probabilistic fusion of kinematics and dynamics [19, 8], or integration of the planned contact state [2]. Strategies for slip rejection include estimated foot velocity thresholding [22], outlier detection through a threshold on the Mahalanobis distance of the innovation [4], and estimating slip probability through a hidden Markov model [20]. Recently, Kim et al. [23] adaptively modulated the contact velocity covariance based on an adaptive filtering strategy, an approach similar to ours. While these methods improve robustness, they generally depend on hand-crafted heuristics and user-defined thresholds. In contrast, our approach learns the optimal noise covariance mapping directly from data via a differentiable pipeline, removing the need for expert tuning.

B. Data-Driven State Estimation

Data-driven approaches to state estimation aim to learn an end-to-end mapping from raw sensor inputs to state trajectories without explicitly encoding a structured filtering algorithm. These methods have seen significant adoption in pedestrian pose estimation, where neural networks process raw IMU data to predict movement [10, 17]. In the domain of legged robotics, Ji et al. [21] proposed training an estimation network through supervised learning, concurrently with training a reinforcement learning policy. Recently, Yu et al. [39] demonstrated that a transformer architecture can outperform the multilayer perceptron (MLP) used in earlier research. While these black-box methods show promise, they often lack the guarantees and consistency of traditional filters, especially on out-of-distribution data. We utilize the transformer-based approach as a baseline in this work to demonstrate that purely data-driven models do not yet outperform hybrid approaches.

C. Hybrid State Estimation

Hybrid approaches bridge the gap between physical models and data-driven methods by integrating learning directly into classical filtering frameworks. These strategies are generally categorized by how the learnable components interact with the estimator. One group of methods predicts measurement inputs [40] or input residuals [6, 28] to refine the data before it enters the estimator. Other implementations focus on estimating input noise [5] or relative displacements, which are then fused into the overall state estimate [29, 7, 11]. These techniques have been widely used in inertial odometry, where networks refine raw IMU data to improve accuracy. Alternative strategies involve predicting state residuals applied to the filter output [13, 25] or learning the Kalman gain [30, 18, 33].

The methodology used to train these learnable components represents another critical design choice for hybrid methods. Differentiable formulations enable end-to-end training based solely on state error, eliminating the need for ground-truth labels of intermediate signals [14]. This approach has been successfully applied to various domains, including inertial odometry [28, 5, 40], manipulation [24, 31, 30], and legged robots [18].

In the field of legged robotics, many strategies center on the explicit estimation of contact states. Lin et al. [26] use a network to classify binary contact states based on labeled real-world data. Sun et al. [36] learn a slip state and slip velocity estimator, and heuristically modulate the contact velocity covariance based on the predicted slip state. Youm et al. [38] learn to predict the robot's base linear velocity in addition to contact states. Although these learning-based contact estimators improve performance over their model-based counterparts, they often still rely on thresholds and heuristics to obtain the ground-truth data necessary for supervised learning or to derive a binary contact prediction from a continuous network output. Our method addresses this by learning contact velocity covariances end-to-end via a differentiable formulation.

III. Method

The aim of this work is to enable robust state estimation across a variety of challenging contact-rich scenarios given a robot morphology, including the IMU location, and a user-specified set of contact candidate points on the robot model. As illustrated in Fig. 1, at each time step, a learned module predicts a covariance for each contact candidate point, which encodes the (directional) confidence that the point is stationary in the world frame. These predictions are then fed into a standard InEKF formulation that fuses IMU measurements and leg odometry.

CoCo-InEKF is based on the contact-aided InEKF [16], which estimates the following state, with the world frame denoted by W, the IMU (body) frame by B, and each contact frame by C_i

$$\mathbf{x} := \left({}^{W}\mathbf{R}_B, {}^{W}\mathbf{v}_B, {}^{W}\mathbf{p}_B, {}^{W}\mathbf{p}_{C_1}, \ldots, {}^{W}\mathbf{p}_{C_N}, {}^{B}\mathbf{b}_{\boldsymbol{\omega}}, {}^{B}\mathbf{b}_{\mathbf{a}}\right).$$

The state consists of base orientation ${}^{W}\mathbf{R}_B$, base linear velocity ${}^{W}\mathbf{v}_B$, base position ${}^{W}\mathbf{p}_B$, N contact positions ${}^{W}\mathbf{p}_{C_i}$, and

IMU gyro and acceleration biases ${}^{B}\mathbf{b}_{\boldsymbol{\omega}}$ and ${}^{B}\mathbf{b}_{\mathbf{a}}$. In the standard contact-aided approach, the contact positions are dynamically added to the state when contact is detected, and removed when contact is released. However, in our approach, we permanently maintain the positions of all contact candidates as part of the state, resulting in a differentiable[1] InEKF.

This key difference in modeling is further highlighted by examining the process model for the contact positions,

$$ {}^{W}\dot{\mathbf{p}}_{C_i} = -{}^{W}\mathbf{R}_{B}\,{}^{B}\mathbf{w}_{C_i}, \tag{1} $$

where ${}^{B}\mathbf{w}_{C_i} \sim \mathcal{N}(\mathbf{0}_{3\times1}, {}^{B}\boldsymbol{\Sigma}_{C_i})$ is a Gaussian noise term[2]. This zero-mean velocity model for the contact positions motivates the original interpretation of static landmarks. However, in our method, where a learned module predicts covariances ${}^{B}\boldsymbol{\Sigma}_{C_i}$, the model's confidence is continuously adjusted. The states ${}^{W}\mathbf{p}_{C_i}$ should therefore rather be interpreted as the continuous positions of the contact candidates in global coordinates, independent of their contact condition.

A. Contact-Aided InEKF

Apart from the deviation discussed in the previous section, we follow the standard contact-aided InEKF formulation as summarized in high-level pseudocode in Alg. 1, including the contact covariances predicted by our learned module. Because our formulation changes are limited to keeping all contacts in the state at all times, we retain the desirable filter invariance and observability properties: given at least one contact point, the state is observable up to global translation and yaw rotation [16]. We omit the derivation of the linearization and discretization of the filter's process and measurement models, providing only their continuous-time, non-linear equations in the subsequent sections. For an in-depth discussion, see Hartley et al. [16].

Prediction: The continuous, non-linear process model for the state $\mathbf{x}$ is given by

$$ {}^{W}\dot{\mathbf{R}}_{B} = {}^{W}\mathbf{R}_{B}\left({}^{B}\boldsymbol{\omega} - {}^{B}\mathbf{b}_{\boldsymbol{\omega}} - {}^{B}\mathbf{w}_{\boldsymbol{\omega}}\right)_{\times}, \tag{2} $$

$$ {}^{W}\dot{\mathbf{v}}_{B} = {}^{W}\mathbf{R}_{B}\left({}^{B}\mathbf{a} - {}^{B}\mathbf{b}_{\mathbf{a}} - {}^{B}\mathbf{w}_{\mathbf{a}}\right) + {}^{W}\mathbf{g}, \tag{3} $$

$$ {}^{W}\dot{\mathbf{p}}_{B} = {}^{W}\mathbf{v}_{B}, \tag{4} $$

$$ {}^{W}\dot{\mathbf{p}}_{C_i} = -{}^{W}\mathbf{R}_{B}\,{}^{B}\mathbf{w}_{C_i}, \tag{5} $$

$$ {}^{B}\dot{\mathbf{b}}_{\boldsymbol{\omega}} = {}^{B}\mathbf{w}_{\mathbf{b}_{\boldsymbol{\omega}}}, \tag{6} $$

$$ {}^{B}\dot{\mathbf{b}}_{\mathbf{a}} = {}^{B}\mathbf{w}_{\mathbf{b}_{\mathbf{a}}}, \tag{7} $$

where Eqs. (2)–(4) model the IMU state forward integration based on gyro measurements ${}^{B}\boldsymbol{\omega}$, accelerometer measurements ${}^{B}\mathbf{a}$, and the gravity vector ${}^{W}\mathbf{g}$. Eq. (5) models the zero-mean contact candidate velocity, and Eqs. (6) and (7) model random walks for the bias terms. The additive noise terms ${}^{B}\mathbf{w}_{\boldsymbol{\omega}}$, ${}^{B}\mathbf{w}_{\mathbf{a}}$, ${}^{B}\mathbf{w}_{\mathbf{b}_{\boldsymbol{\omega}}}$, and ${}^{B}\mathbf{w}_{\mathbf{b}_{\mathbf{a}}}$ are zero-mean Gaussians for gyro measurements, linear acceleration measurements, and bias drift.

[1]Without the discrete addition and removal of contact points, the InEKF consists of a constant computation graph, where all operations (matrix multiplication, inversion, and group operations) are differentiable.

[2]Compared to the formulation in [16], we formulate the noise term directly in the IMU frame instead of the contact frame.

ALGORITHM 1: CoCo-InEKF. The $\oplus$ notation signifies an addition on the state manifold.

Initialize: $\mathbf{x}, \mathbf{P}$
for *each time step* t **do**
  Input: IMU and actuator measurements
  `// 1. Prediction`
  ${}^{B}\boldsymbol{\Sigma}_{C_i} \leftarrow$ ContactNet
  $\boldsymbol{\Phi}, \mathbf{Q} \leftarrow$ Linearization & discretization of Eqs. (2)–(7)
  $\mathbf{x}^{-} \leftarrow$ Integrate Eqs. (2)–(7)
  $\mathbf{P}^{-} \leftarrow \boldsymbol{\Phi}\mathbf{P}\boldsymbol{\Phi}^{\top} + \mathbf{Q}$
  `// 2. Correction`
  $\mathbf{z}, \mathbf{H}, \mathbf{N} \leftarrow$ Linearized measurement Eq. (8)
  $\mathbf{K} \leftarrow \mathbf{P}^{-}\mathbf{H}^{\top}(\mathbf{H}\mathbf{P}^{-}\mathbf{H}^{\top} + \mathbf{N})^{-1}$
  $\mathbf{x} \leftarrow \mathbf{K}\mathbf{z} \oplus \mathbf{x}^{-}$
  $\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^{-}$
end

Correction: The leg kinematics are used to form a measurement model. Given joint position measurements $\mathbf{q}$, the forward kinematics $\mathbf{h}_{C_i}(\mathbf{q})$ returns the position of frame C_i relative to the body frame B, expressed in B. The same quantity can be derived from the estimated state, leading to the measurement equation

$$ \mathbf{h}_{C_i}(\mathbf{q}) = {}^{W}\mathbf{R}_{B}^{\top}\left({}^{W}\mathbf{p}_{C_i} - {}^{W}\mathbf{p}_{B}\right) + {}^{B}\mathbf{J}_{C_i}(\mathbf{q})\mathbf{w}_{\mathbf{q}}, \tag{8} $$

where ${}^{B}\mathbf{J}_{C_i}(\mathbf{q})$ is the Jacobian of the forward kinematics, and $\mathbf{w}_{\mathbf{q}}$ is a zero-mean Gaussian noise for the joint positions.

B. Learning Contact Covariances

Our learned contact module predicts per-contact-point velocity covariances in the body frame ${}^{B}\boldsymbol{\Sigma}_{C_i}$. We ensure that the output of the learned contact module is a valid, symmetric, positive-semidefinite matrix by predicting the 6 elements of a lower-triangular 3×3 matrix $\mathbf{L}$. The covariance is then given by ${}^{B}\boldsymbol{\Sigma}_{C_i} = \mathbf{L}\mathbf{L}^{\top}$. The input to this module is a history of length H of proprioceptive sensor inputs,

$$ \mathbf{o} := \left({}^{B}\boldsymbol{\omega}, {}^{B}\mathbf{a}, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}, {}^{B}\mathbf{p}_{B\to C_i}, {}^{B}\mathbf{v}_{B\to C_i}\right), $$

where $\boldsymbol{\tau}$ are actuator torque measurements, ${}^{B}\mathbf{p}_{B\to C_i} = \mathbf{h}_{C_i}(\mathbf{q})$ are relative contact positions, and ${}^{B}\mathbf{v}_{B\to C_i}$ are their relative velocities. To maintain a clean separation between the filter dynamics and learned predictions, and to avoid complex feedback loops, we intentionally exclude internal InEKF state estimates from the contact module's input features. The history of H inputs is denoted by $\mathbf{O}$. As in [26], these inputs are normalized along the time dimension.

Due to computational constraints, we use the more efficient MLP implementation introduced in [26], as the convolutional neural network (CNN)-based approach is unable to run in real time on our onboard system. We evaluate the impact of this architectural change in Sec. V-B1.

Our training setup uses BPTT, as illustrated in Fig. 2. Given an existing control policy, we record the sensor input history and ground-truth states as we roll out the policy in simulation. This is done in parallel across E environments, where we randomize friction, terrain shapes, and disturbance forces. At

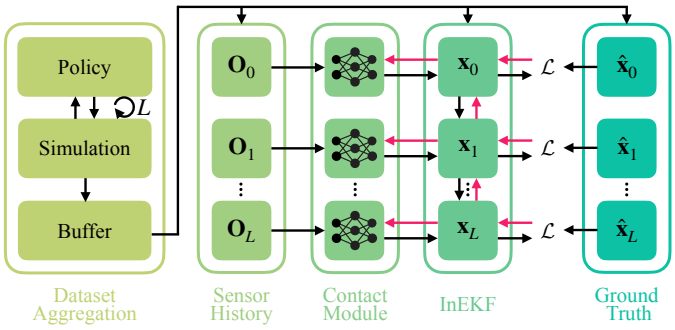

Fig. 2: **Training setup**. Per learning iteration, we collect a training dataset of ground-truth states and sensor measurements by forward-simulating a pretrained policy in E environments. The contact module predicts contact velocity covariances based on the sensor history (IMU + actuator data). These covariances are passed to the differentiable InEKF, rolling out the state estimate for L time steps. We compute a loss on the error between the estimated and ground-truth states, and backpropagate gradients (red arrows) from all time steps through the InEKF to the contact module parameters.

environment initialization, the InEKF is set to the ground-truth state and an initial covariance. Afterward, the InEKF maintains its internal state and covariance across consecutive rollouts up to a maximum episode length T, allowing it to drift away from the ground-truth state. During a rollout, the proprioceptive and ground-truth state data are stored in a buffer of length L and are then used to run CoCo-InEKF and compute losses by comparing the estimated and ground-truth states. Finally, we average the loss across time and environments, backpropagate through our differentiable InEKF to each contact covariance prediction, and update the contact module parameters using the Adam optimizer for a learning iteration. We experimented with re-initializing the InEKF state to the ground-truth state at the start of each rollout in a force-teacher fashion, but observed a degradation in the filter's learning performance.

As the loss function, we use an L2 loss on the body velocity in the body frame,

$$\mathcal{L}\left(\mathbf{x}, \hat{\mathbf{x}}\right) = \left\| {}^{W}\mathbf{R}_B^{\top}\, {}^{W}\mathbf{v}_B - {}^{W}\hat{\mathbf{R}}_B^{\top}\, {}^{W}\hat{\mathbf{v}}_B \right\|_2^2, \tag{9}$$

where the hat symbol $\hat{(\cdot)}$ indicates ground-truth values from the simulator. We optimize for the velocity in the body frame, as this is the frame used by the downstream control policies. Moreover, the body frame velocities are an observable quantity of the InEKF, while the velocity in the world frame is only partially observable due to the unobservable global yaw of the robot. This quantity is also not affected by the InEKF's drifting behavior during an episode. Adding additional loss terms based on the position and orientation states is straightforward, however the InEKF's global drift over the episode must be accounted for in the loss formulations.

C. Automated Contact Candidate Selection

To facilitate the application of CoCo-InEKF to new robots, we present an automated method for generating a set of contact candidates. The method's performance is on par with that of hand-picked contact point sets, as shown in Sec. V-B4.

TABLE I: Baseline methods.

InEKF, GT contacts	InEKF, with ground-truth contact using xy-velocity ($\leq$0.25 m/s) and height ($\leq$0.01 m).
InEKF, heuristic contacts	InEKF, with contact heuristic using estimated xy-velocity ($\leq$0.25 m/s) and height ($\leq$0.01 m).
Hybrid Baseline	Method from [26].
Hybrid Baseline+	As Hybrid Baseline, but with reduced model size and added slip classification.
SET	Unstructured end-to-end transformer-based method [39].

Given a target number of N contact candidates, and a set of robot rigid bodies on which contact points should be placed, we sample $10 \cdot N$ points per rigid body on their respective mesh surfaces, concatenating them. Starting from a random seed and placing the robot in a nominal pose, we then use farthest point sampling to pick N points. Intuitively, this will place points at extremities such as the toes or hands, as well as distributing them across the robot.

IV. Experimental Setup

We test CoCo-InEKF against several baselines, including state-of-the-art state estimation approaches. For a fair comparison, we adapt the baseline methods to accommodate the limited computing power of our robot hardware, evaluating multiple baseline variations for completeness. Models are trained and evaluated on two scenarios consisting of different motion types: dynamic *dancing motions* (foot contact only) and *ground motions* (full-body ground contact). We evaluate design choices through a set of ablation studies, investigate the impact of framework changes on the statistical consistency of the filter, and demonstrate that our method enables dynamic motion on a physical robot.

A. Implementation Details

We evaluate models on Lima, a custom bipedal robot (0.84 m, 16.2 kg, 20 DoF) with an onboard computer (Intel i7, 4-Core, 1.7 GHz) running a 600 Hz control loop. For dancing motions, we place contact points at the heels and toes for a total of $N = 4$. For ground motions, we place $N = 10$ points across the robot.

Models are trained on a single Nvidia RTX 4090 GPU for 100k iterations, or a maximum of 5 days, with $E = 1280$ parallel environments. Unless otherwise noted, we use a history of $H = 20$ and a training buffer length of $L = 128$. All simulations are performed at 600 Hz, matching the robot's low-level loop rate.

For computational timing benchmarks, the models are executed single-threaded on the robot's onboard computer as part of the real-time control loop.

1) Baseline Methods: We evaluate CoCo-InEKF against a set of approaches, as listed in Tab. I. The InEKF method with ground-truth contacts cannot be implemented on a real robot, but presents a best-case scenario for an InEKF with binary contacts.

TABLE II: Configuration of small and large SET models.

	Small	Large
Self-attention blocks	6	6
Heads per block	4	8
Linear token embedding dimensions	128	256
MLP hidden dimensions	256	1024

For the Hybrid Baseline+ method, we adapt the Hybrid Baseline method from Lin et al. [26] to run on our available compute by: (1) removing CNN layers and flattening input data; (2) reducing the MLP hidden layer sizes to 128 and 64 units; (3) predicting the contact state of each contact point independently; and (4) adding a 'slip' prediction class. Similar to Kim et al. [23], we then increase the contact velocity covariance by a factor of $10\times$ when slip is predicted. As we show in Sec. V-B1, these changes yield a significant speedup while maintaining performance.

For the state estimation transformer (SET) method from Yu et al. [39], we adapt the inputs to be consistent with our model and express linear velocities in the body frame so that our loss formulation can be used. We integrate velocity predictions using forward Euler to obtain world position estimates, as these are not directly estimated by the method, and use complementary IMU filtering [37] for a more robust orientation estimate. To evaluate performance trade-offs of this approach, we compare two model sizes as listed in Tab. II.

B. Datasets

For the dancing scenario, we use a VMP policy [34] that tracks arbitrary kinematic reference motions. The observations follow [34] with an asymmetric actor-critic training setup where the privileged observations for the critic are noise-free. For the reference motions, we use a subset of the Reallusion dataset [32], with 81 sequences of 5.6–36.1 s duration, retargeted onto the Lima robot. For training, we concatenate randomly-drawn sequences from this dataset and track them in simulation up to an episode length $T = 100\,\mathrm{s}$. The dancing test dataset comprises 100 such sequences, each 20 s in duration, drawn from the same motion pool but concatenated differently.

For the ground motions, we use a stylized falling policy [35] to track a sequence of goal end poses. For each episode, the robot starts from a standing pose, the goal pose changes every 2 s, and the environment is reset after an episode length $T = 6\,\mathrm{s}$ during training. This produces contact-rich motions with challenging full-body ground contacts. The ground motion test dataset again comprises 100 such sequences, each 20 s in duration, produced by different concatenations of the same goal poses to include more ground interactions.

During training, we apply domain randomization of friction coefficients and disturbance forces. In the dancing scenario, we also vary the terrain. For the test datasets, we apply the same randomization of friction coefficients, but omit the terrain variation and only include periodic disturbances for the ground motions to induce slippage.

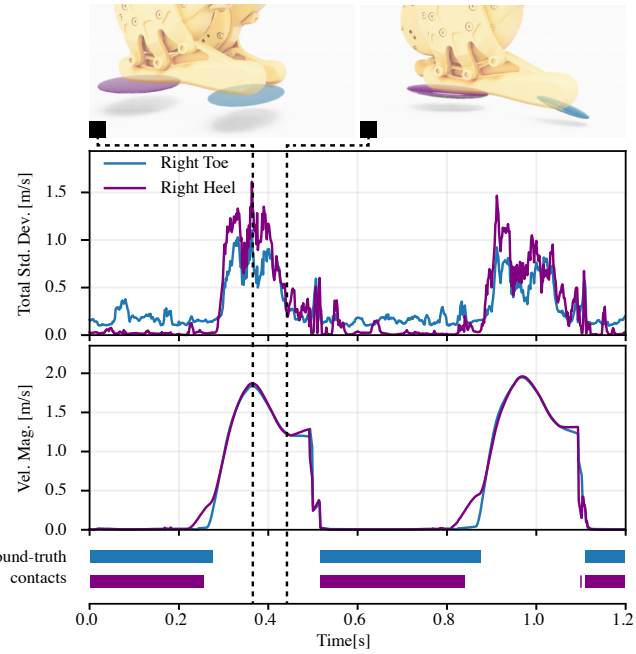

Fig. 3: **Contact Covariance.** Visualization of the contact covariance total standard deviation $\sqrt{\mathrm{tr}({}^{B}\boldsymbol{\Sigma}_{C_i})}$ and contact point velocity magnitude during a forward-walking gait of our robot, along with ground-truth contacts.

C. Metrics

Offline metrics are based on [42]. We report absolute trajectory error (ATE), computed over the trajectory after aligning the initial state, as well as the root mean square error (RMSE), mean absolute error (MAE), median absolute error (MED), and standard deviation (STD). Linear velocity errors are computed in the robot IMU body frame, to prevent cross-coupling to orientation errors. Error units are $\mathrm{m/s}$, m, rad for velocity, position, and rotation, respectively.

For the statistical filter analysis, we evaluate the normalized estimation error squared (NEES) [1] for the combined core filter states comprised of position, orientation, and velocity, as well as for the states in isolation.

V. Results

A. Simulation Experiments

1) Contact Covariances: To provide insights into the velocity covariance representation, we analyze a forward-walking gait cycle, see Fig. 3. We plot the total covariance standard deviation and compare it to the ground-truth binary contact state (computed as in Tab. I), for the right heel and toe contact points. We also visualize the covariance ellipsoids for two instances in time.

Although physical interpretability is not enforced by our framework, we observe an alignment of the ground-truth contacts with features in the covariance. However, the covariance also includes additional nuances that the InEKF can leverage. For example, the total standard deviations of the covariance ellipsoids are observed to agree with the instantaneous velocities

TABLE III: ATE comparison on simulated dancing motions.

	Linear Velocity ATE				Position ATE				Orientation ATE			
Model	RMSE	MAE	MED	STD	RMSE	MAE	MED	STD	RMSE	MAE	MED	STD
InEKF, GT contacts	0.176	0.070	0.037	0.162	0.422	0.119	0.042	0.405	0.033	0.022	0.013	0.025
InEKF, heuristic contacts	2.675	1.429	0.436	2.262	17.423	8.260	0.866	15.340	0.041	0.031	0.023	0.027
Hybrid Baseline	0.123	0.060	0.034	0.107	0.163	0.078	0.048	0.143	0.031	0.019	**0.011**	0.025
Hybrid Baseline+	0.121	0.061	0.036	0.105	**0.111**	**0.065**	**0.040**	**0.090**	**0.027**	**0.019**	0.012	**0.020**
CoCo-InEKF (ours)	**0.046**	**0.028**	**0.018**	**0.037**	0.124	0.079	0.052	0.096	0.031	0.020	0.013	0.024
SET, small	0.279	0.195	0.138	0.199	0.487	0.379	0.313	0.305	0.072	0.059	0.052	0.042
SET, large	0.286	0.203	0.143	0.202	0.395	0.290	0.205	0.269	0.072	0.059	0.052	0.042

TABLE IV: ATE comparison on simulated ground motions.

	Linear Velocity ATE				Position ATE				Orientation ATE			
Model	RMSE	MAE	MED	STD	RMSE	MAE	MED	STD	RMSE	MAE	MED	STD
InEKF, GT contacts	0.418	0.296	0.203	0.295	1.573	0.971	0.512	1.238	0.078	0.048	0.028	0.061
InEKF, heuristic contacts	4.448	3.110	2.117	3.179	32.695	19.738	10.248	26.065	**0.049**	0.039	0.032	**0.029**
Hybrid Baseline	0.363	0.266	0.190	0.247	1.455	0.899	0.414	1.144	0.063	**0.037**	**0.020**	0.051
Hybrid Baseline+	0.428	0.316	0.232	0.289	1.982	1.238	0.613	1.547	0.074	0.041	0.021	0.061
CoCo-InEKF (ours)	0.099	0.069	0.042	**0.071**	0.342	0.271	0.239	0.210	0.069	0.056	0.049	0.040
SET, small	0.107	0.069	0.034	0.082	**0.126**	**0.100**	**0.087**	**0.076**	0.058	0.041	0.031	0.041
SET, large	**0.096**	**0.063**	**0.032**	0.072	0.159	0.132	0.124	0.088	0.058	0.041	0.031	0.041

of the contact points. Moreover, inspecting two snapshots of the forward gait, the directionality of the covariances can be seen to generally be in agreement with the contact candidates' displacement, constraining the tangential directions via low covariance magnitudes.

2) Dancing Motions: The metrics for CoCo-InEKF and the baseline methods, when trained and evaluated on the simulated dancing motions, are summarized in Tab. III. Our proposed CoCo-InEKF model achieves the lowest linear velocity errors, while the InEKF approach with heuristic contact detection exhibits very high errors, indicating estimator divergence. While our method is slightly outperformed in terms of position and orientation ATE, we hypothesize that adding additional losses on these states could improve our method's performance on those metrics. Because of the differentiable training setup, adding such loss terms to Eq. (9) is trivial.

3) Ground Motions: The linear velocity ATE in root frame as well as the position and orientation ATE in world frame are summarized in Tab. IV for the simulated ground motions. Our CoCo-InEKF matches the performance of the much larger SET models for 10 contact points, and shows a significant reduction in all error metrics compared to our other baselines. However, as we show in Sec. V-B3, compared to SET, our novel approach is significantly faster, and able to better scale to additional contact points on the robot body.

Note that 10 contact candidates results in a comparatively sparse contact set for full-body motions, as can be seen from Fig. 5. We hypothesize that as our method reasons about contacts in terms of velocity covariances rather than binary contact flags, the contact points are not required to be exactly in contact. As soon as a contact candidate is stationary along a certain dimension, this information can be used to improve the state estimate.

B. Ablation Studies

To analyze the contribution of our design choices to the overall performance of CoCo-InEKF, we conduct a series of ablation studies focusing on network architecture and the specific hyperparameters used during training.

1) Architecture and Inputs: We first evaluate the performance and inference time of different network architectures. We compare the Hybrid Baseline (CNN architecture), Hybrid Baseline+, SET baselines, and our method. We also evaluate the effect of each of the changes that were made between Hybrid Baseline and Hybrid Baseline+: replacing the CNN with an MLP architecture; reducing MLP hidden layer sizes and predicting each contact independently; adding 'slip' classification to the outputs. The results are summarized in Tab. V. It can be seen that Hybrid Baseline+ achieves better performance than Hybrid Baseline, and a $5\times$ speedup. The SET baselines are computationally most expensive, with the highest RMSE. CoCo-InEKF clearly outperforms the other methods.

We also investigate the effect of history size, comparing $H = 20$ to $H = 150$. Results are shown in Tab. VI. For all methods, $H = 150$ causes performance degradation both in terms of inference time as well as estimation accuracy.

2) Training Configurations: We study the effect of changing the BPTT horizon L, for lengths 64, 128, and 256. Results are summarized in Tab. VII. Significant performance deterioration is seen for the shorter buffer size, presumably because the model cannot predict longer-horizon effects. Performance is also slightly worse for the longer buffer size, which could be explained by vanishing or exploding gradients, or due to higher training cost leading to fewer training iterations.

3) Number of Contact Candidates: Using the ground motions, we study the effect of changing the number of contact candidates. We compare the baseline configuration with 4 contact points (Fig. 4a) to denser configurations of 10

TABLE V: Model architecture ablation w.r.t. linear velocity ATE on dancing motions for our model, baseline models, and intermediate models between Hybrid Baseline and Hybrid Baseline+, showing the effects of individual changes. We also report the number of parameters and the inference time for the neural network (NN), together with the full state estimator (SE). For SET, a single value is reported, as the SE consists solely of the NN.

Model	RMSE	# Params.	NN / SE [ms]
Hybrid Baseline	0.123	2'473'744	1.81 / 2.10
Hybrid Baseline, MLP	0.135	239'824	0.15 / 0.44
Hybrid Baseline+ w/o slip	0.122	239'304	0.16 / 0.45
Hybrid Baseline+	0.121	239'564	0.16 / 0.45
CoCo-InEKF (ours)	**0.046**	240'344	0.14 / 0.42
SET, small	0.279	810'755	2.09
SET, large	0.286	4'770'307	3.02

TABLE VI: History size ablation, $H = 20$ vs. $H = 150$. We also report number of parameters and the inference time for the neural network (NN) and the full state estimator (SE), respectively.

Model	H	RMSE	# Params.	NN / SE [ms]
Hybrid Baseline	20	0.123	2'473'744	1.81 / 2.10
Hybrid Baseline	150	0.150	10'862'352	6.43 / 6.79
Hybrid Baseline+	20	0.121	239'564	0.16 / 0.45
Hybrid Baseline+	150	0.124	1'737'164	0.62 / 0.98
CoCo-InEKF	20	**0.046**	240'344	0.14 / 0.42
CoCo-InEKF	150	0.052	1'737'944	0.61 / 0.97

and 18 hand-picked contact points distributed over the robot (Figs. 5a and 5b). See Tab. VIII for results.

For CoCo-InEKF, a positive correlation between contact point number and estimation accuracy is seen, at the expense of computational cost. The SET baseline shows limited improvement from 10 to 18 contacts.

4) Automated Contact Candidate Selection: To test our automated contact candidate selection approach, we consider two test cases: dancing motions ($N = 8$, feet only), and ground motions ($N = 10$, over the full body). For each test case, we hand-select a baseline and compare it to 9 samples from the automated method. The baselines and automated samples are shown in Fig. 4 for dancing motions and Fig. 5 for ground motions.

TABLE VII: BPTT unroll size ablation. We report linear velocity ATE on synthetic dancing data, as well as the number of training iterations (limited by the 5-day training time).

	RMSE	MAE	MED	STD	# Iters.
$L = 64$	0.066	0.043	0.027	0.051	89'600
$L = 128$ (ours)	**0.046**	**0.028**	**0.018**	**0.037**	64'600
$L = 256$	0.051	0.032	0.021	0.040	18'800

TABLE VIII: Ablation study on the scaling of the number of contact points. We report linear velocity ATE on synthetic ground motion data, number of parameters, and inference time for the neural network (NN) and the full state estimator (SE), respectively. For SET, a single value is reported, as the SE consists solely of the NN.

Model	N	RMSE	# Params.	NN / SE [ms]
CoCo-InEKF	4	0.134	240'344	0.14 / 0.42
CoCo-InEKF	10	0.099	334'844	0.18 / 0.87
CoCo-InEKF	18	**0.069**	460'844	0.26 / 2.01
SET, small	10	0.107	815'363	2.03
SET, large	10	0.096	4'779'523	3.06
SET, small	18	0.104	821'507	2.08
SET, large	18	0.094	4'791'811	3.12

TABLE IX: The automated contact candidate selection compared to the handpicked baseline, for linear velocity ATE. Ranges indicate [worst, best] sample.

	RMSE
Dancing motions, automated	[0.056, 0.052]
Dancing motions, handpicked	0.057
Ground motions, automated	[0.104, 0.092]
Ground motions, handpicked	0.099

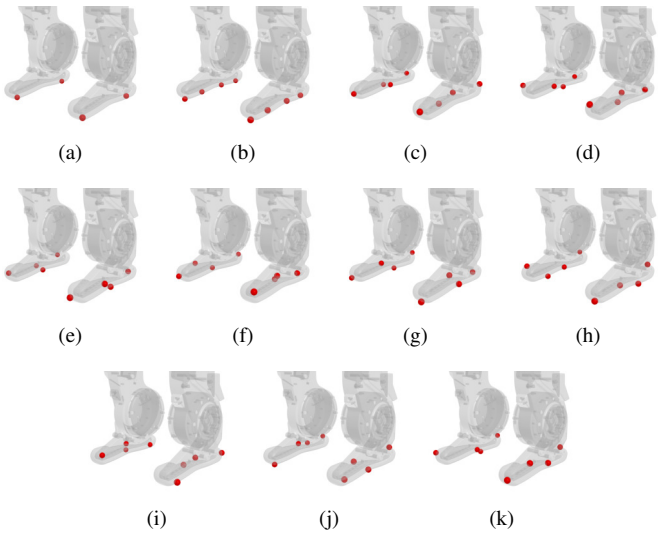

Fig. 4: **Foot contact candidate configurations.** As used in the evaluation and ablation study for dancing motions. Configurations (a) and (b), with 4 and 8 candidates respectively, were handpicked. All others have 8 candidates and were automatically generated using our proposed sampling method.

Results are summarized in Tab. IX. It can be seen that there is little variation across the randomly initialized, automated selections, and that the performance is similar to or better than the handpicked baseline. These results also support the hypothesis that the method is able to use the velocity covariance contact representation to reason about contact points that are not exactly in contact.

C. Statistical Filter Consistency

To assess the consistency of the proposed filter architecture, we employ the normalized estimation error squared (NEES) metric [1]. A state estimator is called consistent if it is unbiased and if the actual estimation error covariance matches the covariance reported by the filter at that time step, i.e.,

$$\mathbb{E}[\tilde{\mathbf{x}}] = \mathbf{0}, \qquad \mathbb{E}\left[\tilde{\mathbf{x}}\,\tilde{\mathbf{x}}^\top\right] = \mathbf{P}, \tag{10}$$

with $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$, where the hat symbol $\hat{(\cdot)}$ indicates the ground-truth state. The NEES normalizes the estimation error by the filter's covariance,

$$\epsilon = \tilde{\mathbf{x}}^\top \mathbf{P}^{-1}\, \tilde{\mathbf{x}}, \tag{11}$$

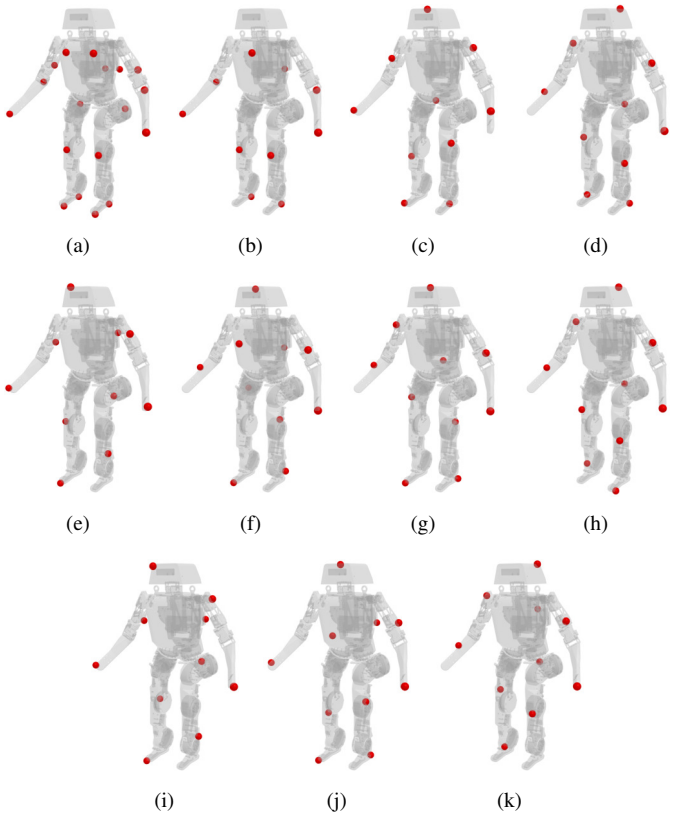

Fig. 5: **Full-body contact candidate configurations.** As used in the evaluation and ablation study for ground motions. Configuration (a) and (b), with 18 and 10 candidates respectively, were handpicked. All others have 10 candidates and were automatically generated using our proposed sampling method.

TABLE X: NEES evaluation on the simulated dancing test data, reported as the percentage of time steps with NEES values within the 95% confidence bounds ($[2.7, 19]$ for the combined core state, $[0.22, 9.35]$ for the individual states).

Model	Core	Vel.	Pos.	Ori.
InEKF, GT contacts	37.7%	66.8%	**60.7%**	65.4%
InEKF, heur. contacts	20.3%	47.4%	23.6%	47.4%
Hybrid Baseline	18.2%	50.2%	11.3%	52.2%
Hybrid Baseline+	18.4%	50.3%	7.9%	51.8%
CoCo-InEKF (ours)	**52.1%**	**71.1%**	59.8%	**68.1%**

and is, under the Gaussian assumption, χ^2-distributed with $n_x = \dim(\tilde{\mathbf{x}})$ degrees of freedom, yielding $\mathbb{E}[\epsilon] = n_x$. Consistency is then evaluated by checking whether ϵ lies within the two-sided $(1-\alpha)$ confidence interval $[r_1, r_2]$ obtained from the inverse χ^2 cumulative distribution function. Values exceeding r_2 indicate an optimistic (overconfident) filter, whereas values below r_1 indicate a conservative (pessimistic) one. For a comprehensive treatment, the reader is referred to [1].

All consistency evaluations are conducted on our diverse dance motion test set, consisting of 100 unique motion trajectories. As shown in Tab. X, CoCo-InEKF matches or exceeds the consistency of the original formulation utilizing ground-truth, privileged information. The baseline methods that estimate

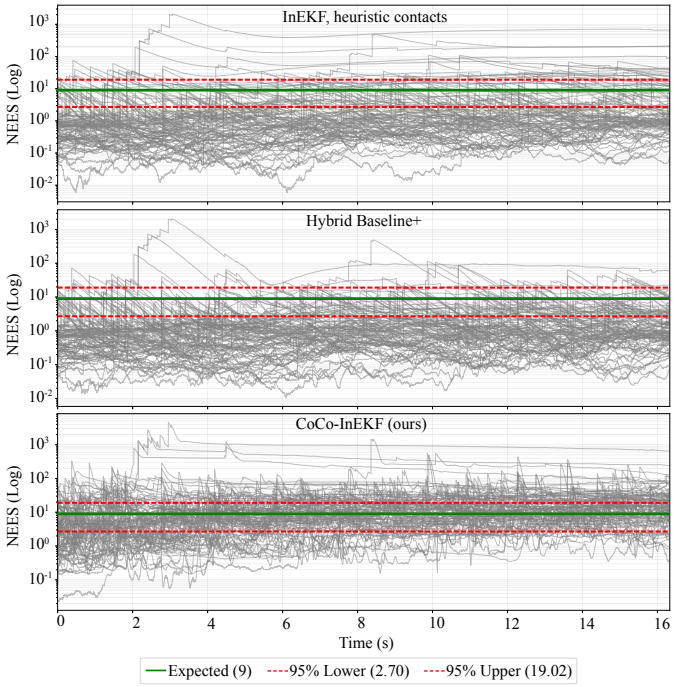

Fig. 6: **Consistency.** Visualization of the normalized estimation error squared (NEES) of the combined core state of baseline InEKF approaches vs. our proposed CoCo-InEKF for 100 dance motion sequences. CoCo-InEKF's NEES values are more consistent with the expected 95% confidence interval of a χ^2 distribution. We omit the Hybrid Baseline, as it performs near identical to the Hybrid Baseline+.

the contact states either heuristically or with a learned binary contact classification show lower overall consistency. They are underconfident across most of the evaluated trajectories, as evident in Fig. 6, whereas CoCo-InEKF exhibits NEES values more in line with the 95% confidence interval extracted from the inverse χ^2 cumulative distribution function. Our formulation thus improves the filter consistency, despite not explicitly optimizing for this metric.

D. Real-World Experiments

We evaluate the state estimators on the physical Lima robot in two distinct scenarios, to gauge the sim-to-real gap and assess the real-time performance of the estimators with a policy.

First, we evaluate them offline on a 10-minute ground-motion dataset comprising 20 messy motion sequences collected with a motion capture (MoCap) system. The MoCap pose data is fused with IMU measurements via a classical InEKF without any contact states to obtain the ground-truth position, orientation, and velocity estimates.

The linear velocity ATE in the root frame, as well as the position and orientation ATE in the world frame, are summarized in Tab. XI. These real-world results agree with our simulation results in Tab. IV and indicate a low sim-to-real gap of our approach. The lower error values compared to the simulated experiments can be explained by less extreme motions due to the physical setup, as well as potentially smaller IMU biases on the real system. Note, however, that this

TABLE XI: ATE comparison on 20 real-world ground motion sequences.

	Linear Velocity ATE				**Position ATE**				**Orientation ATE**			
Model	RMSE	MAE	MED	STD	RMSE	MAE	MED	STD	RMSE	MAE	MED	STD
InEKF, heuristic contacts	1.5419	0.5305	0.0261	1.4478	21.3841	6.0165	**0.0671**	20.5203	**0.0152**	**0.0110**	**0.0081**	0.0105
Hybrid Baseline*	0.1178	0.0738	0.0453	0.0918	0.7212	0.3764	0.2093	0.6152	0.0152	0.0110	0.0085	**0.0105**
Hybrid Baseline+	0.1699	0.1108	0.0669	0.1288	1.4869	0.7305	0.3396	1.2951	0.0188	0.0129	0.0095	0.0137
CoCo-InEKF (ours)	**0.0805**	**0.0398**	**0.0167**	**0.0699**	0.2019	0.1497	0.1181	0.1355	0.0302	0.0201	0.0130	0.0225
SET, small*	0.1002	0.0501	0.0193	0.0867	**0.1665**	**0.1240**	0.0952	**0.1110**	0.0257	0.0165	0.0109	0.0197
SET, large*	0.0974	0.0475	0.0174	0.0850	0.2245	0.1803	0.1498	0.1337	0.0257	0.0165	0.0109	0.0197

TABLE XII: Success rate [%] of various real-world dance motions with the state estimators in-the-loop. The pirouette and moonwalk are not part of the training set.

Model	Dances (training set)	Pirouette (unseen)	Moonwalk (unseen)
MoCap	92	90	**100**
InEKF, heuristic contacts	77	60	**100**
Hybrid Baseline+	85	50	10
CoCo-InEKF (ours)	**95**	**100**	**100**

evaluation is performed offline on recorded real-world data; models marked with * cannot run in real time.

Second, we run the state estimators in the loop with a VMP [34] controller for 13 dynamic dance motions. We utilize the same non-privileged policy inputs as in simulation. The policy inputs contain the body frame linear velocity estimates from our state estimator, alongside the body frame angular velocity directly extracted from the on-board IMU. To demonstrate generalization, we include challenging unseen motions: a pirouette and a moonwalk. We also compare with the previously described MoCap-based state estimator that directly tracks the robot's IMU frame without contacts. Note that some baseline models were too computationally expensive for execution on the robot at the 600 Hz rate, and are therefore not evaluated here.

For each method, we attempt each of the seen motions $5\times$, and the unseen pirouette and moonwalk $10\times$. We record the success rate (i.e., when the robot does not fall). Results are summarized in Tab. XII. It can be seen that our method outperforms all other methods, including the MoCap state estimation — this highlights the challenging nature of the motions. We also refer to the supporting video.

VI. CONCLUSION

CoCo-InEKF is a differentiable Invariant Extended Kalman Filter that utilizes a neural module to predict contact velocity covariances rather than relying on binary contact states. Trained end-to-end via backpropagation through time, the framework avoids the need for ground-truth contact labels and effectively handles complex contact states. Experiments on the Lima bipedal robot demonstrate that the method advances the accuracy-efficiency Pareto front, and can run within a 600 Hz onboard control loop. Furthermore, the system is insensitive to the exact placement of contact candidates, supporting an automated selection process that performs on par with expert-handpicked configurations.

While trained exclusively in simulation, the method supports training on real-world data with ground-truth states obtained from motion capture. We are eager to explore whether incorporating real-world data or greater training diversity can further improve performance. In the future, we plan to apply the framework to diverse robot morphologies and integrate these proprioceptive estimates with exteroceptive sensors, such as LiDAR or vision, for global drift correction. Such advancements will pave the way for agile robots capable of navigating unpredictable, contact-rich environments with unprecedented robustness.

REFERENCES

[1] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, New York, 2001. ISBN 978-0-471-41655-5.

[2] Gerardo Bledt, Patrick M Wensing, Sam Ingersoll, and Sangbae Kim. Contact model fusion for event-based locomotion in unstructured terrains. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4399–4406, 2018.

[3] Michael Bloesch, Marco Hutter, Mark Hoepflinger, Stefan Leutenegger, Christian Gehring, C. David Remy, and Roland Siegwart. State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and IMU. In *Proceedings of Robotics: Science and Systems*, 2012.

[4] Michael Bloesch, Christian Gehring, Péter Fankhauser, Marco Hutter, Mark A Hoepflinger, and Roland Siegwart. State estimation for legged robots on unstable and slippery terrain. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6058–6064, 2013.

[5] Martin Brossard, Axel Barrau, and Silvère Bonnabel. AI-IMU dead-reckoning. *IEEE Transactions on Intelligent Vehicles*, 5(4):585–595, 2020.

[6] Martin Brossard, Silvere Bonnabel, and Axel Barrau. Denoising IMU gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics and Automation Letters*, 5(3):4796–4803, 2020.

[7] Russell Buchanan, Marco Camurri, Frank Dellaert, and Maurice Fallon. Learning Inertial Odometry for Dynamic Legged Robot State Estimation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings*

of the 5th Conference on Robot Learning, volume 164 of *Proceedings of Machine Learning Research*, pages 1575–1584. PMLR, 2022.

[8] Marco Camurri, Maurice Fallon, Stéphane Bazeille, Andreea Radulescu, Victor Barasuol, Darwin G. Caldwell, and Claudio Semini. Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots. *IEEE Robotics and Automation Letters*, 2(2): 1023–1030, 2017.

[9] Marco Camurri, Milad Ramezani, Simona Nobili, and Maurice Fallon. Pronto: A Multi-Sensor State Estimator for Legged Robots in Real-World Scenarios. *Frontiers in Robotics and AI*, 7, 2020. Publisher: Frontiers.

[10] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. IONet: learning to cure the curse of drift in inertial odometry. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[11] Giovanni Cioffi, Leonard Bauersfeld, Elia Kaufmann, and Davide Scaramuzza. Learned inertial odometry for autonomous drone racing. *IEEE Robotics and Automation Letters*, 8(5):2684–2691, 2023.

[12] Geoff Fink and Claudio Semini. Proprioceptive Sensor Fusion for Quadruped Robot State Estimation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10914–10920, 2020.

[13] Yaru Gu, Ze Liu, and Ting Zou. Enhancing Leg Odometry in Legged Robots with Learned Contact Bias: An LSTM Recurrent Neural Network Approach. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6832–6839, 2024.

[14] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop KF: learning discriminative deterministic state estimators. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4383–4391. Curran Associates Inc., 2016.

[15] Ross Hartley, Josh Mangelson, Lu Gan, Maani Ghaffari Jadidi, Jeffrey M Walls, Ryan M Eustice, and Jessy W Grizzle. Legged robot state-estimation through combined forward kinematic and preintegrated contact factors. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4422–4429, 2018.

[16] 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. Publisher: SAGE Publications Ltd STM.

[17] Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 3146–3152, 2020.

[18] Lasse Hohmeyer, Mihaela Popescu, Ivan Bergonzani, Dennis Mronga, and Frank Kirchner. InEKFormer: A Hybrid State Estimator for Humanoid Robots. In *2025 IEEE International Conference on Advanced Robotics (ICAR)*, pages 833–840, 2025.

[19] Jemin Hwangbo, Carmine Dario Bellicoso, Péter Fankhauser, and Marco Hutter. Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3872–3878, 2016.

[20] Fabian Jenelten, Jemin Hwangbo, Fabian Tresoldi, C Dario Bellicoso, and Marco Hutter. Dynamic locomotion on slippery ground. *IEEE Robotics and Automation Letters*, 4(4):4170–4176, 2019.

[21] Gwanghyeon Ji, Juhyeok Mun, Hyeongjun Kim, and Jemin Hwangbo. Concurrent Training of a Control Policy and a State Estimator for Dynamic and Robust Legged Locomotion. *IEEE Robotics and Automation Letters*, 7 (2):4630–4637, 2022.

[22] Joon-Ha Kim, Seungwoo Hong, Gwanghyeon Ji, Seunghun Jeon, Jemin Hwangbo, Jun-Ho Oh, and Hae-Won Park. Legged Robot State Estimation With Dynamic Contact Event Information. *IEEE Robotics and Automation Letters*, 6(4):6733–6740, 2021.

[23] Kyung-Hwan Kim, DongHyun Ahn, Dong-hyun Lee, JuYoung Yoon, and Dong Jin Hyun. Adaptive Invariant Extended Kalman Filter for Legged Robot State Estimation. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3063–3068, 2025.

[24] Michelle A Lee, Brent Yi, Roberto Martín-Martín, Silvio Savarese, and Jeannette Bohg. Multimodal sensor fusion with differentiable filters. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10444–10451, 2020.

[25] Seokju Lee, Hyun-Bin Kim, and Kyung-Soo Kim. Legged Robot State Estimation Using Invariant Neural-Augmented Kalman Filter with a Neural Compensator. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 15445–15452, 2025.

[26] Tzu-Yuan Lin, Ray Zhang, Justin Yu, and Maani Ghaffari. Legged Robot State Estimation using Invariant Kalman Filtering and Learned Contact Events. In *Conference on Robot Learning*, pages 1057–1066. PMLR, 2022.

[27] Tzu-Yuan Lin, Tingjun Li, Wenzhe Tong, and Maani Ghaffari. Proprioceptive Invariant Robot State Estimation, 2024. arXiv:2311.04320 [cs].

[28] Ben Liu, Tzu-Yuan Lin, Wei Zhang, and Maani Ghaffari. Debiasing 6-DOF IMU via Hierarchical Learning of Continuous Bias Dynamics. In *Proceedings of Robotics: Science and Systems*, 2025.

[29] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020.

[30] Xiao Liu, Yifan Zhou, Shuhei Ikemoto, and Heni Ben Amor. α-MDF: An Attention-based Multimodal Differentiable Filter for Robot State Estimation. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 3870–3893. PMLR, 2023.

[31] Nicola A. Piga, Ugo Pattacini, and Lorenzo Natale. A Differentiable Extended Kalman Filter for Object Tracking Under Sliding Regime. *Frontiers in Robotics and AI*, 8, 2021. Publisher: Frontiers.

[32] Reallusion. 3d animation and 2d cartoons made simple., 2023. URL https://www.reallusion.com. studio-mocap-girl-dance, studio-mocap-evolution-of-dance-vol-1, studio-mocap-evolution-of-dance-vol-2, iclone-motion-pack—street-dance-locking.

[33] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adrià López Escoriza, Ruud J. G. van Sloun, and Yonina C. Eldar. KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics. *IEEE Transactions on Signal Processing*, 70:1532–1547, 2022.

[34] Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. VMP: Versatile Motion Priors for Robustly Tracking Motion on Physical Characters. *Computer Graphics Forum*, 43(8):e15175, 2024.

[35] Pascal Strauch, David Müller, Sammy Christen, Agon Serifi, Ruben Grandia, Espen Knoop, and Moritz Bächer. Robot Crash Course: Learning Soft and Stylized Falling, 2025. arXiv:2511.10635 [cs].

[36] Peng Sun, Qi Li, Hao Hu, Junjie Qiang, Weiwei Wu, and Xin Luo. Proprioceptive slip detection and state estimation of multi-legged robots in slippery scenarios. *Frontiers of Mechanical Engineering*, 20(5):36, 2025.

[37] Roberto G. Valenti, Ivan Dryanovski, and Jizhong Xiao. Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs. *Sensors*, 15(8): 19302–19330, 2015. Publisher: Multidisciplinary Digital Publishing Institute.

[38] Donghoon Youm, Hyunsik Oh, Suyoung Choi, Hyeongjun Kim, Seunghun Jeon, and Jemin Hwangbo. Legged Robot State Estimation with Invariant Extended Kalman Filter Using Neural Measurement Network. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 670–676, 2025.

[39] Chen Yu, Yichu Yang, Tianlin Liu, Yangwei You, Mingliang Zhou, and Diyun Xiang. State estimation transformers for agile legged locomotion. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6810–6817, 2024.

[40] Ming Zhang, Mingming Zhang, Yiming Chen, and Mingyang Li. IMU data processing for inertial aided navigation: A recurrent neural network based approach. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3992–3998, 2021.

[41] Tianyi Zhang, Wenhan Cao, Chang Liu, Tao Zhang, Jiangtao Li, and Shengbo Eben Li. Robust State Estimation for Legged Robots With Dual Beta Kalman Filter. *IEEE Robotics and Automation Letters*, 10(8):7955–7962, 2025.

[42] Zichao Zhang and Davide Scaramuzza. A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7244–7251, 2018.