Extended Abstract Track

EqNIO: Subequivariant Neural Inertial Odometry

Editors: Bahareh Tolooshams, Derek Lim, Congyue Deng, Nina Miolane

Abstract

Neural networks that regress the displacement and associated covariance of an inertial measurement unit (IMU) purely from its accelerometer and gyroscope measurements have become key enablers to low-drift inertial odometry, but still ignore the physical roto-reflective symmetries inherent in IMU data, thus hindering generalization. In this work, we show that IMU data, displacements and covariances transform equivariantly, when rotated around and reflected across planes parallel to gravity. We design a neural network that equivariantly estimates a gravity-aligned frame from IMU data, leveraging tailored linear and non-linear layers, and uses it to canonicalize the data. We train an off-the-shelf inertial odometry network on this data and map its outputs back into the original frame, thus obtaining equivariant covariances and displacements. To highlight its generality, we apply the framework to both filter-based and end-to-end approaches and show better performance on the TLIO, Aria, RIDI and OxIOD datasets than existing methods.

Keywords: equivariance, inertial odometry, subequivariance

1. Introduction

Inertial Measurement Units (IMUs) measure body accelerations and angular velocities and are widely used to track inertial frames in robot navigation, AR/VR, etc. IMU-based Inertial Odometry (IO) promises robust tracking which does not suffer from motion blur and saturation effects as would a Visual Inertial Odometry (VIO) system. Purely IO can be broadly classified into kinematic, and learning-based approaches. Kinematic-based approaches Leishman et al. (2014); Titterton et al. (2004); Bortz (1971); Solin et al. (2018); Groves (2015); Hartley et al. (2020); Brajdic and Harle (2013); Jimenez et al. (2009); Ho et al. (2016); Foxlin (2005); Rajagopal (2008); Beaufils et al. (2019) leverage analytical solutions, loop closures or handcrafted pseudo measurements for an Extended Kalman Filter (EKF), but suffer from drift due to noise. By contrast, learning-based approaches leverage Deep Neural Networks (NN) to denoise IMU measurements Brossard et al. (2020b); Buchanan et al. (2023); Brossard et al. (2020a); Steinbrener et al. (2022), regress velocity/displacement Yan et al. (2018); Asraf et al. (2022); Herath et al. (2020); Chen et al. (2018a); Sun et al. (2021) or provide motion priors (i.e. displacement and covariances) Liu et al. (2020)(TLIO), Chen et al. (2021a); Russell and Reale (2021) for EKF/factor-graph based filtering. NN-based statistical displacement priors (TLIO, Herath et al. (2020)(RONIN)) have been instrumental in reducing this drift and performing competitively against VIO methods. Yet, these NNs fail to generalize beyond the particular motion directions and mounting orientations they were trained on. In this work, we aim to develop generalizable networks by leveraging the equivariance of IMU data, displacement and covariance when rotated around and reflected across planes parallel to the gravity axis (i.e. $O(2)$ roto-reflections, an $O(3)$ subequivariance). Our proposed network enforces this symmetry by design, and produces roto-reflection consistent trajectories, in contrast to TLIO, RONIN, Russell and Reale (2021), and Cao et al. (2022)(RIO) (Figure 1) which rely on rotation augmentations or auxiliary losses. Extensive research has been conducted on how group equivariant networks Cohen and Welling (2016); Cesa et al. (2021); Xu et al. (2022) process a variety of inputs, including point clouds Thomas et al. (2018); Chen et al. (2021b); Deng et al. (2021); Villar et al. (2021), 2D Worrall et al. (2017); Weiler and Cesa (2019), 3D Weiler et al. (2018); Esteves et al. (2019), spherical

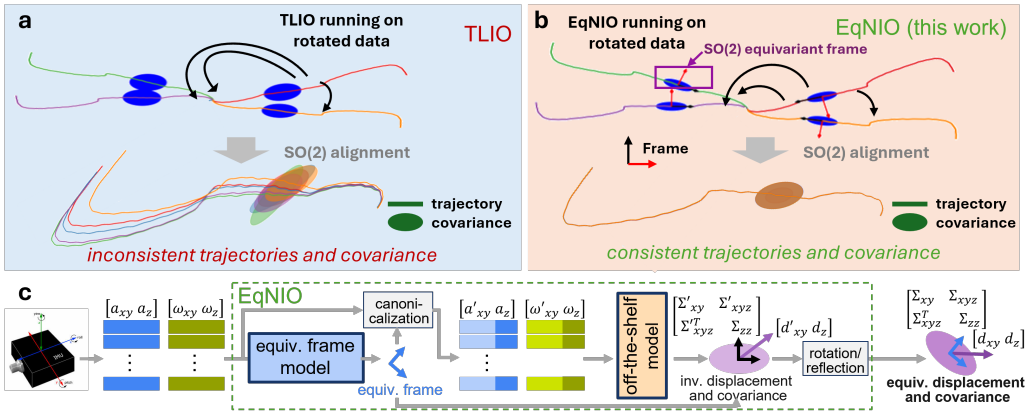

Figure 1: Trajectories and covariances from non-equivariant TLIO (a) and EqNIO (b) for identical trajectories with different IMU frames. Unlike TLIO's, our de-rotated trajectories and ellipsoids are perfectly aligned. c) EqNIO canonicalizes gravity-aligned IMU data via a frame derived from an equivariant network, and predicts invariant displacement (d') and covariance (Σ') using an off-the-shelf model. Mapping back the outputs into the original frame, yields equivariant displacement (d) and covariance (Σ).

images Cohen et al. (2018); Esteves et al. (2018, 2020, 2023), graphs Satorras et al. (2021), and general manifolds Cohen et al. (2019b,a); Weiler et al. (2021); Xu et al. (2024); Finzi et al. (2021). Related works Han et al. (2022); Chen et al. (2023) tackle subequivariance using equivariant graph networks. However, since IMU data forms a temporal sequence of non-trivially transforming vector measurements affected by a global gravity direction, novel equivariant linear and non-linear layers need to be designed. In particular, gravity reduces the full $O(3)$ equivariance to $O(2)$ equivariance, and angular rates need to be non-trivially preprocessed before they can be equivariantly processed. Using these layers we produce an equivariant frame (F), with which we canonicalize the IMU data, before running an off-the-shelf network that produces invariant displacements and covariances. Mapping them into the original frame, produces equivariant outputs.

2. Method

Given a sequence IMU accelerometer and gyroscope measurements $\{(a_i, \omega_i)\}_{i=1}^{n}$ $(a_i, \omega_i \in \mathbb{R}^3)$, expressed in the local IMU inertial frame we regress 3 degrees-of-freedom (DoF) linear velocities (RONIN) or 3D displacement measurements $d \in \mathbb{R}^3$ and covariances $\Sigma \in \mathbb{R}^{3\times3}$ (TLIO). In the latter case, we treat the network outputs as measurements and fuse them in an Extended Kalman Filter (EKF) estimating the IMU state, *i.e.* orientation, position, velocity, IMU biases, and uncertainties. Preliminaries on inertial odometry and details on the EKF are given in Appendix A.1.2 and A.4.

We map the IMU data to a *gravity-aligned frame* by rotating it to a frame with a gravity-aligned z-axis using the orientation estimated from the current EKF state. This frame, however, is ill-defined, because it is specified up to an arbitrary yaw rotation and reflection. This implies the IMU data only behaves equivariantly when roto-reflected around gravity. These roto-reflections $R \in O_{\boldsymbol{g}}(3) := \{R \in O(3) | R\boldsymbol{g} = \boldsymbol{g}\}$ form a subgroup of $O(3)$ isomorphic to $O(2)$, and thus we treat the problem with $O(2)$ equivariance techniques.

Similarly, $R \in SO_{\boldsymbol{g}}(3)$ form a subgroup of $SO(3)$ isomorphic to $SO(2)$ which we call rotation subequivariance.

Our framework (see Fig. 1 bottom), (i) bijects IMU data into a space that transforms equivariantly under the specific group representation of $O(2)$, (ii) estimates a canonical frame using an equivariant neural network to canonicalize the IMU data, (iii) predicts linear velocity or displacement and covariance with an off-the-shelf neural network Φ, and (iv) finally remaps the outputs into the original frame, yielding equivariant outputs.

Bijection: Let $R_{2\times2}$ be the roto-reflection around gravity (z-axis) which acts on acceleration as $a_i' = (R_{2\times2}\oplus1)a_i$, where $\oplus$ constructs a block diagonal matrix of $R_{2\times2}$ and 1. Unfortunately, for angular rates, this transformation is different $\omega_i' = \det(R_{2\times2})(R_{2\times2} \oplus 1)\omega_i$, i.e. it changes sign upon reflection, therefore we decompose ω_i into v_1, v_2 such that $\omega = v_1 \times v_2$ Selecting $v_{1/2} = \sqrt{\|\omega\|}w_{1/2}/\|w_{1/2}\|$ with $w_1 = (-\omega_y, \omega_x, 0)$ and $w_2 = \omega \times w_1$, yields the desired result. The decomposed parts transform as $v_{1/2}' = (R_{2\times2} \oplus 1)v_{1/2}$ under rotation $R_{2\times2}$, identical to the acceleration, and their cross product has the desirable property $\omega' = v_1' \times v_2' = \det(R_{2\times2})(R_{2\times2} \oplus 1)(v_1 \times v_2) = \det(R_{2\times2})(R_{2\times2} \oplus 1)\omega$.

Equivariant Frame (Eq F.): We use a two-branch architecture inspired by Villar et al. (2021) (broken down in Appendix A.3) to process n IMU measurements as $n \times 2 \times C_0^v$ ($C_0^v = 3$) vector features derived from $v_{1,xy}, v_{2,xy}, a_{xy}$ each transforming with $R_{2\times2}$, and $n \times C_0^s$, ($C_0^s = 9$) scalar features, comprised of the invariant norms of these vectors, their dot products, and the z-components of each vector. While scalars are processed with conventional MLPs and 1D convolutions, vectors are processed with specifically designed equivariant linear layers. We then mix scalars and vector features with equivariant non-linearities. Our network outputs a set of two vector features (see Section A.3 in the Appendix) which are converted into an orthonormal $SO(2)$ or $O(2)$ frame F via Gram-Schmidt orthogonalization.

Linear Layer: Using Eq. 2 in Finzi et al. (2021), we find the basis of weights that commute with rotation of vector features $v^{\text{in}} \in \mathbb{R}^2$, i.e. $v'^{\text{out}} = WR_{2\times2}v^{\text{in}} = R_{2\times2}Wv^{\text{in}} = R_{2\times2}v^{\text{out}}$

$$SO(2): \quad v^{\text{out}} = v^{\text{in}}W_1 + R_{90}v^{\text{in}}W_2 \qquad O(2): \quad v^{\text{out}} = v^{\text{in}}W_1 \tag{1}$$

with $v^{\text{out}} \in \mathbb{R}^{2\times C_{\text{out}}}$, $v^{\text{in}} \in \mathbb{R}^{2\times C_{\text{in}}}$, $W_1, W_2 \in \mathbb{R}^{C_{\text{in}}\times C_{\text{out}}}$ and R_{90} a 90°. Summing linear projections over a temporal receptive field yields 1D convolutions.

Non-Linear Layer: We apply a pointwise nonlinearity inspired by the gated nonlinearity Weiler et al. (2018). Specifically, for n vector and scalar features $v^{\text{in}} \in \mathbb{R}^{n\times2\times C}, s^{\text{in}} \in \mathbb{R}^{n\times C}$, we process channel-wise concatenated norm features $\|v^{\text{in}}\| \in \mathbb{R}^{n\times C}$ and scalar features s^{in} with an MLP with output of size $n \times 2C$. We then split this output into new norm features $\gamma \in \mathbb{R}^{n\times C}$ and activations $\beta \in \mathbb{R}^{n\times C}$ which we modulate with a non-linearity $s^{\text{out}} = \sigma(\beta)$ and use to rescale the original vector features $v^{\text{out}} = \gamma v^{\text{in}}$. See Figure 3 for more details.

Remapping of Outputs: Invariant displacement d' and covariance Σ' from our base model are made equivariant via $d = (F \oplus 1)d'$, and $\Sigma = (F \oplus 1)\Sigma'(F \oplus 1)^\intercal$. Since F is equivariant to $R_{2\times2}$ roto-reflection and d' and Σ' are invariant, d and Σ transform equivariantly.

3. Experiments and Ablation

Implementation and Training Details: As TLIO we use loss function $\mathcal{L}_{MSE}(\hat{d}, d) = \|d - \hat{d}\|_2$ in the first stage, and $\mathcal{L}_{MLE}(\hat{d}, d) = (d - \hat{d})^T\Sigma^{-1}(d - \hat{d})$ in the second stage when $\mathcal{L}_{MSE}$ converges. Unlike TLIO we parameterize covariance as a diagonal matrix in the canonical frame, and we empirically observed that d_{xy} and d_z are independent (see

Dataset	TLIO Dataset						Aria Dataset					
Model	MSE*	ATE	ATE*	RTE	RTE*	AYE	MSE*	ATE	ATE*	RTE	RTE*	AYE
TLIO	3.242	1.812	3.722	0.500	0.551	2.376	5.322	1.285	2.102	0.464	0.521	2.073
TLIO-N	3.333	1.722	3.079	0.521	0.542	2.366	15.248	1.969	4.560	0.834	0.977	2.309
TLIO-NQ	3.008	1.429	2.443	0.495	0.496	2.411	2.437	1.213	2.071	0.458	0.508	2.096
TLIO-PCA	3.473	1.506	2.709	0.523	0.535	2.459	6.558	1.717	4.635	0.771	0.976	2.232
Eq CNN	3.194	1.580	3.385	0.564	0.610	2.394	8.946	3.223	6.916	1.091	1.251	2.299
TLIO + Eq F. SO(2)	3.194	1.480	2.401	0.490	0.501	2.428	2.457	1.178	1.864	0.449	0.484	2.084
TLIO + Eq F. O(2)	2.982	1.433	2.406	0.458	0.478	2.389	2.304	1.118	1.849	0.416	0.465	2.059

Dataset	RONIN-U		RONIN-S		RIDI-T		RIDI-C		OxIOD	
Model	ATE*	RTE*	ATE*	RTE*	ATE*	RTE*	ATE*	RTE*	ATE*	RTE*
RONIN-100%	5.14	4.37	3.54	2.67	1.63	1.91	1.67	1.62	3.46	4.39
RIO B-ResNet	5.57	4.38	-	-	1.19	1.75	-	-	3.52	4.42
RIO J-ResNet	5.02	4.23	-	-	1.13	1.65	-	-	3.59	4.43
RIO B-ResNet-TTT	5.05	4.14	-	-	1.04	1.53	-	-	2.92	3.67
RIO J-ResNet-TTT	5.07	4.17	-	-	1.03	1.51	-	-	2.96	3.74
RONIN + Eq F. SO(2)	5.18	4.35	3.67	2.72	0.86	1.59	0.63	1.39	1.22	2.39
RONIN + Eq F. O(2)	4.42	3.95	3.32	2.66	0.82	1.52	0.70	1.41	1.28	2.10

Table 1: Diplacement error on various datasets (red, orange and yellow denote the first, second and third lowest result). $*$ denote results without EKF. MSE are multiplied by 100.

Appendix A.1.3 for more covariance details and Appendix A.9 for the empirical proof.)

Baseline Models: We apply our framework to **RONIN**, an end-to-end deep learning approach RONIN and compare **RIO**, and a filter-based approach with a learned prior TLIO. While RONIN regresses only the 2D velocities and integrates them to produce a trajectory, TLIO estimates orientation, position, velocity, IMU biases, and covariances, which are used as measurement updates of an extended Kalman filter (EKF). With TLIO-N, TLIO-NQ, TLIO-PCA, Eq CNN we denote TLIO trained without yaw augmentations, non-equivariant frames, PCA to predict handcrafted frames and fully equivariant CNN respectively.

Metrics: The NN performance is evaluated with Mean Squared Error (MSE) (m^2), Absolute Translation Error (ATE) (m), and Relative Translation Error (RTE) (m), on trajectories reconstructed via cumulative summation or EKF filtering (see Appendices A.2, A.3, A.6, A.9, A.5 for full experiments).

Results: As seen in Table 1, on the Aria Dataset, Eq F. $O(2)$ outperforms other methods by 56%, 12%, and 10% with the baseline TLIO model on MSE*, ATE*, and RTE* respectively. It outperforms even RIO J-ResNet-TTT by 56% and 43% on ATE* and RTE* on the RIDI-T and OxIOD Datasets, despite being trained on only 50% of the data. These results highlight the strong generalization capabilities of our method.

Ablation: Table 1 shows that yaw augmentations improve generalization, non-equivariant MLP and handcrafted frames (PCA) predict deleterious equivariant, non-smooth frames, and a fully equivariant CNN using the basic layers described in Section 2 is overly restrictive.

4. Conclusion

Our framework robustly regresses an equivariant frame, capturing the inherent symmetry of IMU data, and enforcing $O(3)$ subequivariance, i.e. $O(2)$ equivariance, in both predicted velocity/displacement and covariance. Coupled with off-the-shelf filter-based or end-to-end models, it improves the state-of-the-art in neural IO.

References

Omri Asraf, Firas Shama, and Itzik Klein. Pdrnet: A deep-learning pedestrian dead reckoning framework. *IEEE Sensors Journal*, 22(6):4932–4939, 2022. doi: 10.1109/JSEN.2021.3066840.

Bertrand Beaufils, Frédéric Chazal, Marc Grelet, and Bertrand Michel. Robust stride detector from ankle-mounted inertial sensors for pedestrian navigation and activity recognition with machine learning approaches. *Sensors*, 19(20):4491, 2019.

John E. Bortz. A new mathematical formulation for strapdown inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-7(1):61–66, 1971. doi: 10.1109/TAES.1971.310252.

Agata Brajdic and Robert Harle. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 225–234, 2013.

Martin Brossard, Axel Barrau, and Silvère Bonnabel. Ai-imu dead-reckoning. *IEEE Transactions on Intelligent Vehicles*, 5(4):585–595, 2020a. doi: 10.1109/TIV.2020.2980758.

Martin Brossard, Silvère Bonnabel, and Axel Barrau. Denoising imu gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robotics and Automation Letters*, 5(3):4796–4803, 2020b. doi: 10.1109/LRA.2020.3003256.

Russell Buchanan, Varun Agrawal, Marco Camurri, Frank Dellaert, and Maurice Fallon. Deep imu bias inference for robust visual-inertial odometry with factor graphs. *IEEE Robotics and Automation Letters*, 8(1):41–48, 2023. doi: 10.1109/LRA.2022.3222956.

Xiya Cao, Caifa Zhou, Dandan Zeng, and Yongliang Wang. Rio: Rotation-equivariance supervised learning of robust inertial odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6614–6623, 2022.

Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build e (n)-equivariant steerable cnns. In *International conference on learning representations*, 2021.

Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018a.

Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, and Niki Trigoni. Oxiod: The dataset for deep inertial odometry. *CoRR*, abs/1809.07491, 2018b. URL http://arxiv.org/abs/1809.07491.

Danpeng Chen, Nan Wang, Runsen Xu, Weijian Xie, Hujun Bao, and Guofeng Zhang. Rnin-vio: Robust neural inertial navigation aided visual-inertial odometry in challenging scenes. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 275–283. IEEE, 2021a.

Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3d point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14514–14523, 2021b.

Runfa Chen, Jiaqi Han, Fuchun Sun, and Wenbing Huang. Subequivariant graph reinforcement learning in 3d environments. In *International Conference on Machine Learning*, pages 4545–4565. PMLR, 2023.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pages 1321–1330. PMLR, 2019a.

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *Advances in neural information processing systems*, 32, 2019b.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.

Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018.

Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1568–1577, 2019.

Carlos Esteves, Ameesh Makadia, and Kostas Daniilidis. Spin-weighted spherical cnns. *Advances in Neural Information Processing Systems*, 33:8614–8625, 2020.

Carlos Esteves, Jean-Jacques Slotine, and Ameesh Makadia. Scaling spherical cnns. *arXiv preprint arXiv:2306.05420*, 2023.

Marc Finzi, Max Welling, and Andrew Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In *International conference on machine learning*, pages 3318–3328. PMLR, 2021.

Eric Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer graphics and applications*, 25(6):38–46, 2005.

Paul D Groves. Principles of gnss, inertial, and multisensor integrated navigation systems, [book review]. *IEEE Aerospace and Electronic Systems Magazine*, 30(2):26–27, 2015.

Jiaqi Han, Wenbing Huang, Hengbo Ma, Jiachen Li, Josh Tenenbaum, and Chuang Gan. Learning physical dynamics with subequivariant graph neural networks. *Advances in Neural Information Processing Systems*, 35:26256–26268, 2022.

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.

Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3146–3152. IEEE, 2020.

Ngoc-Huynh Ho, Phuc Huu Truong, and Gu-Min Jeong. Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone. *Sensors*, 16(9):1423, 2016.

Antonio R Jimenez, Fernando Seco, Carlos Prieto, and Jorge Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 37–42. IEEE, 2009.

Robert C. Leishman, John C. Macdonald, Randal W. Beard, and Timothy W. McLain. Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Systems Magazine*, 34(1):28–41, 2014. doi: 10.1109/MCS.2013.2287362.

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.

Zhaoyang Lv, Nickolas Charron, Pierre Moulon, Alexander Gamino, Cheng Peng, Chris Sweeney, Edward Miller, Huixuan Tang, Jeff Meissner, Jing Dong, et al. Aria everyday activities dataset. *arXiv preprint arXiv:2402.13349*, 2024.

Sujatha Rajagopal. Personal dead reckoning system with shoe mounted inertial sensors. *Master's Degree Project, Stockholm, Sweden*, 2008.

Rebecca L Russell and Christopher Reale. Multivariate uncertainty in deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12):7937–7943, 2021.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

Arno Solin, Santiago Cortes, Esa Rahtu, and Juho Kannala. Inertial odometry on handheld smartphones. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–5. IEEE, 2018.

Jan Steinbrener, Christian Brommer, Thomas Jantos, Alessandro Fornasier, and Stephan Weiss. Improved state propagation through ai-based pre-processing and down-sampling of high-speed inertial data. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6084–6090, 2022. doi: 10.1109/ICRA46639.2022.9811989.

Scott Sun, Dennis Melamed, and Kris Kitani. Idol: Inertial deep orientation-estimation and localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6128–6137, 2021.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

D. Titterton, J.L. Weston, Institution of Electrical Engineers, American Institute of Aeronautics, and Astronautics. *Strapdown Inertial Navigation Technology*. IEE Radar Series. Institution of Engineering and Technology, 2004. ISBN 9780863413582. URL https://books.google.com/books?id=WwrCrn54n5cC.

S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. doi: 10.1109/34.88573.

Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. *Advances in Neural Information Processing Systems*, 34:28848–28863, 2021.

Dian Wang, Jung Yeon Park, Neel Sortur, Lawson L. S. Wong, Robin Walters, and Robert Platt. The surprising effectiveness of equivariant models in domains with latent symmetry, 2023. URL https://arxiv.org/abs/2211.09231.

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018.

Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. *arXiv preprint arXiv:2106.06020*, 2021.

Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5028–5037, 2017.

Yinshuang Xu, Jiahui Lei, Edgar Dobriban, and Kostas Daniilidis. Unified fourier-based kernel and nonlinearity design for equivariant networks on homogeneous spaces. In *International Conference on Machine Learning*, pages 24596–24614. PMLR, 2022.

Yinshuang Xu, Jiahui Lei, and Kostas Daniilidis. $se(3)$ equivariant convolution and transformer in ray space. *Advances in Neural Information Processing Systems*, 36, 2024.

Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Appendix A. Appendix

A.1. Preliminary

A.1.1. Equivariance

In this section, we introduce more preliminaries of group and representation theory which form the mathematical tools for equivariance.

Group The group G is a set equipped with an associative binary operation $\cdot$ which maps two arbitrary two elements in G to an element in G. It includes an identity element, and every element in the set has an inverse element.

In this paper, we focus on the group $SO(2)$ and $O(2)$. $SO(2)$ is the set of all 2D planar rotations, represented by 2x2 orthogonal matrices with determinant 1. This group operation is matrix multiplication, and each rotation matrix has an inverse, which is its transpose. The identity element is the matrix representing no rotation.

$O(2)$ consists of all distance-preserving transformations in Euclidean 2D space, including both rotations and reflections. Elements of $O(2)$ are 2x2 orthogonal matrices, with the group operation being matrix multiplication. Each transformation matrix has an inverse, and the identity element is the matrix representing no transformation.

Group Representation and Irreducible Representation Group representation is a homomorphism from the group G to the general linear map of a vector space V of a file K, denoted $GL(V)$.

An irreducible representation (irrep) of a group G is a representation in which the only invariant subspaces under the action of G are the trivial subspace $\{\mathbf{0}\}$ and the entire space V. In other words, an irreducible representation cannot be broken down into smaller, nontrivial representations,i.e., it cannot be the direct sum of several nontrivial representations.

For $SO(2)$, we can use $\theta \in (0, 2\pi]$ to represent $SO(2)$, for any θ, the irreducible representation of the frequency $n \in \mathbb{N}$ is:

$$\rho_n(\theta) = \begin{pmatrix} \cos n\theta & -\sin n\theta \\ \sin n\theta & \cos n\theta \end{pmatrix}.$$

For $O(2)$, we can use $r \in \{-1, 1\}$ to denote reflection and $\theta \in (0, 2\pi]$ to denote rotation. The trivial representation $\rho_0(r, \theta) = 1$. For the nontrivial representation of frequency $n \in \mathbb{N}^+$

$$\rho_n(r, \theta) = \begin{pmatrix} \cos(n\theta) & -\sin(n\theta) \\ \sin(n\theta) & \cos(n\theta) \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & r \end{pmatrix}$$

There is another one-dimensional irreps for $O(2)$, $\rho(r, \theta) = r$ which corresponds to the trivial representation of rotation.

The introduction to group representations has been covered extensively in previous work on equivariance Cohen and Welling (2016); Weiler et al. (2018); Xu et al. (2024). Specifically, for $SO(2)$ and $O(2)$, Weiler and Cesa (2019) provide a detailed introduction.

Invariance and Equivariance Given a network $\Phi : \mathcal{X} \to \mathcal{Y}$, if for any $x \in \mathcal{X}$,

$$\Phi(\rho^{\mathcal{X}} x) = \Phi(x),$$

implies the group representation $\rho^{\mathcal{Y}}$ of the output space is trivial, *i.e.* identity, and the input does not transform (*i.e.* the input is invariant) under the action of the group. In our paper, the coordinates/ projections of $3D$ vector to the gravity axis z-axis are invariant, therefore we call them invariant scalars.

A network $\Phi : \mathcal{X} \to \mathcal{Y}$ is equivariant if it satisfies the constraint

$$\Phi(\rho^{\mathcal{X}} x) = \rho^{\mathcal{Y}} \Phi(x).$$

In this paper, when the output is displacement the $z-$ component is invariant while $xy-$ components are acted under the representation of ρ_1 defined in the above section. Hence, for displacement, $\rho^{\mathcal{Y}} = \rho_1 \oplus 1$ and for covariance $3D$ covariance, $\rho^{\mathcal{Y}} = (\rho_1 \oplus 1) \otimes (\rho_1 \oplus 1)$

Subequivariance As mentioned in prior works Chen et al. (2023); Han et al. (2022), the existence of gravity breaks the symmetry in the vertical direction, reducing O(3) to its subgroup O(2). We formally characterize this phenomenon of equivariance relaxation as subequivariance. We have mathematically defined the subequivariance in Section 2 of the paper. In simpler terms, the gravity axis is decoupled and treated as an invariant scalar while the other two axes are handled as a separate 2D vector. Upon rotation, the invariant scalar remains constant while the other two axes are transformed under rotation. So we are limited now to SO(2) rotations and roto-reflections. In the general case of equivariance, the 3D vector would be considered three-dimensional and an SO(3) rotation would act on it. The transformation would be along all three axes.

A.1.2. Inertial Odometry

In this section, we introduce more preliminaries on the terms used in inertial odometry.

Inertial Measurement Unit Inertial Measurement Unit (IMU) is an electronic device that measures and reports linear acceleration, angular velocity, orientation, and other gravitational forces. An IMU typically consists of a 3-axis accelerometer, a 3-axis gyroscope, and depending on the heading requirement a 3-axis magnetometer.

An accelerometer measures instantaneous linear acceleration (a_i). It can be thought of as a mass on a spring, however in micro-electro-mechanical systems (MEMS) it is beams that flex instead of spring.

A gyroscope measures instantaneous angular velocity (ω_i). It measures the angular velocity of its frame, not any external forces. Traditionally, this can be measured by the fictitious forces that act on a moving object brought about by the Coriolis effect, when the frame of reference is rotating. In MEMs, however, we use high-frequency oscillations of a mass to capture angular velocity readings by the capacitance sense cones that pick up the torque that gets generated.

World Frame A world frame, also known as a cartesian coordinate frame, is a fixed frame with a known location and does not change over time.

Gravity-aligned World Frame When the world frame has one of its axes perfectly aligned with the gravity vector, it is said to be a gravity-aligned world frame. In this paper, we denote this frame with w.

Local-gravity-aligned Frame A local-gravity-aligned frame has one of its axes aligned with the gravity vector at all times but it is not fixed to a known location.

Body Frame A body frame comprises the origin and orientation of the object described by the navigation solution. In this paper, the body frame is the IMU's frame. This is denoted as i for the IMU data.

Gravity-compensation Gravity compensation refers to the removal of the gravity vector from the accelerometer reading.

Gravity-alignment Gravity-alignment of IMU data refers to expressing the data in the gravity-aligned frame. This is done by aligning the z-axis of the IMU inertial frame with the gravity vector pointing downwards and is usually achieved by fixing the roll and pitch (rotations around the x and y axes) or by applying a transformation estimated by the relative orientation between the gravity vector and a fixed z-axis pointing downwards. This is usually achieved with a simple rotation.

A.1.3. Uncertainty Quantification in Inertial Odometry

In this section, we provide more context on uncertainty quantification in odometry and detail the different parameterizations used for regressing the covariance matrix in the paper.

Homoscedastic Uncertainty Homoscedatic uncertainty refers to uncertainty that does not vary for different samples, i.e., it is constant.

Heteroscedastic Uncertainty Heteroscedastic uncertainty is uncertainty that is dependent on the sample, i.e., it varies from sample to sample.

Epistemic Uncertainty Epistemic uncertainty is uncertainty in model parameters. This can be reduced by training the model for longer and/or increasing the training dataset to include more diverse samples.

Aleatoric Uncertainty Aleatoric uncertainty is the inherent noise of the samples. This cannot be reduced by tuning the network or increasing the diversity of the data.

Why do we need to estimate uncertainty in inertial odometry? In inertial odometry when we use a probabilistic filter-based approach like a Kalman Filter, the filter estimates the probability distribution over the pose recursively. While integrating the neural network prediction, the filter fuses the prediction with other sensor measurements, like raw IMU data in TLIO Liu et al. (2020), by weighing it based on the accuracy or reliability of the measurements. For neural networks, this reliability is obtained by estimating the uncertainty. If we use a fixed uncertainty (homoscedastic) it is seen to cause catastrophic failures of perception systems. The uncertainty estimated in TLIO captures the extend to which input measurements encode the motion model prior.

What is the uncertainty we are estimating in inertial odometry? We are regressing aleatoric uncertainty using the neural network and training the model till the epistemic uncertainty is very small as compared to aleatoric uncertainty.

How is the uncertainty estimated in this paper? We regress aleatoric uncertainty as a covariance matrix jointly while regressing 3D displacement following the architecture of TLIO Liu et al. (2020). Since there is no ground truth for the covariance, we use the negative log-likelihood loss of the prediction using the regressed Gaussian distribution. As this loss captures the Mahalanobis distance, the network gets jointly trained to tune the covariance prediction. We do not estimate epistemic uncertainty separately in this paper, but as mentioned in Russell and Reale (2021) we train the network until the epistemic uncertainty is small as compared to aleatoric uncertainty.

Diagonal covariance matrix TLIO Liu et al. (2020) regresses only the three diagonal elements of the covariance matrix as $\log \sigma_{xx}$, $\log \sigma_{yy}$ and $\log \sigma_{zz}$ and the off-diagonal elements are zero. This formulation assumes the axes are decoupled and constrains the uncertainty ellipsoid to be along the local gravity-aligned frame.

Full covariance matrix using Pearson correlation Russell and Reale (2021) define a parameterization to regress the full covariance matrix. They regress six values of which three are the diagonal elements $\log \sigma_{xx}$, $\log \sigma_{yy}$ and $\log \sigma_{zz}$ and the remaining three are Pearson correlation coefficients ρ_{xy}, ρ_{yz}, and ρ_{xz}. The diagonal elements are obtained by exponential activation while the off-diagonal elements are computed as follows

$$\Sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$$

where ρ_{ij} passes through tanh activation.

Diagonal covariance matrix in canonical frame In our approach we regress the three diagonal elements as $\log \sigma_{xx}$, $\log \sigma_{yy}$ and $\log \sigma_{zz}$ in the invariant canonical frame. Since the z-axis is decoupled from the xy-axis, only σ_{xx} and σ_{yy} are back-projected using the equivariant frame to obtain a full 2D covariance matrix from the diagonal entries. The resulting matrix is as follows

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & 0 \\ \sigma_{xy} & \sigma_{yy} & 0 \\ 0 & 0 & \sigma_{zz} \end{bmatrix}$$

A.2. Dataset Details

In this section, we provide a detailed description of the 4 datasets used in this work - TLIO and Aria for TLIO architecture, and RONIN, RIDI and OxIOD for RONIN architecture.

TLIO Dataset- The TLIO Dataset Liu et al. (2020) is a headset dataset that consists of IMU raw data at 1kHz and ground truth obtained from MSCKF at 200 Hz for 400 sequences totaling 60 hours. The ground truth consists of position, orientation, velocity, IMU biases and noises in $\mathbb{R}^3$. The dataset was collected using a custom rig where an IMU (Bosch BMI055) is mounted on a headset rigidly attached to the cameras. This dataset captures a variety of activities including walking, organizing the kitchen, going up and down stairs, on multiple different physical devices and more than 5 people for a wide range of individual motion patterns, and IMU systematic errors. We use their data splits for training (80%), validation (10%), and testing(10%).

Aria Everyday Dataset- Aria Everyday Dataset Lv et al. (2024) is an open-sourced egocentric dataset that is collected using Project Aria Glasses. This dataset consists of 143 recordings accumulating to 7.3 hrs capturing diversity in wearers and everyday activities like reading, morning exercise, and relaxing. There are two IMUs on the left and right side of the headset of frequencies 800 and 1kHz respectively. They have two sources of ground truth-open and closed loop trajectory at 1kHz. Open loop trajectory is strictly causal while closed loop jointly processes multiple recordings to place them in a common coordinate system. The ground truth contains position and orientation in $\mathbb{R}^3$. We use it as a test dataset. The raw right IMU data is used to compare closed-loop trajectory with EKF results. The data was downsampled to 200Hz and preprocessed using the closed-loop trajectory to test the Neural Network trained on TLIO.

RONIN Dataset- RONIN Dataset Herath et al. (2020) consists of pedestrian data with IMU frequency and ground truth at 200Hz. RONIN data features diverse sensor placements, like the device placed in a bag, held in hand, and placed deep inside the pocket, and multiple Android devices from three vendors Asus Zenfone AR, Samsung Galaxy S9 and Google Pixel 2 XL. Hence, this dataset has different IMUs depending on the vendor. We use RONIN data splits to train and test their model with and without our framework.

RIDI Dataset- RIDI Dataset Yan et al. (2018) is another pedestrian dataset with IMU frequency and ground truth at 200 Hz. This dataset features specific human motion patterns like walking forward/backward, walking sidewards, and acceleration/deceleration. They also record data with four different sensor placements. We report test results of RONIN models on both RIDI test and cross-subject datasets. RIDI results are presented after post-processing the predicted trajectory with the Umeyama algorithm Umeyama (1991) for fair comparison against other methods.

OxIOD Dataset- OxIOD Dataset Chen et al. (2018b) stands for Oxford Inertial Odometry Dataset consists of various device placements/attachments, motion modes, devices, and users capturing everyday usage of mobile devices. The dataset contains 158 sequences totaling 42.5 km and 14.72 hours captured in a motion capture system. We use their unseen multi-attachments test dataset for evaluating our framework applied to RONIN architecture.

A.3. Equivariant Network Implementation Details

In this section, we describe in detail the equivariant network implementation and how it is combined with TLIO and RONIN. The input to the framework is IMU samples from the accelerometer and gyroscope for a window of 1s with IMU frequency 200Hz resulting in $n = 200$ samples. All IMU samples within a window are gravity-aligned with the first sample at the beginning of the window, previously referred to as the clone state. During network training, the samples are aligned using the ground-truth orientation of the clone state. The network design, as seen in Figure 2 and Figure 3, differs in architecture for $SO(2)$ and $O(2)$ and hence described separately below.

$SO(2)$- We decouple the z-axis from the other two axes and treat linear acceleration and angular velocity along the z-axis as scalars (2). We also take the norm of the 2D accelerometer and gyroscope measurements (2), their inner product (1) resulting in invariant scalars $\mathbb{R}^{n\times 5}$.

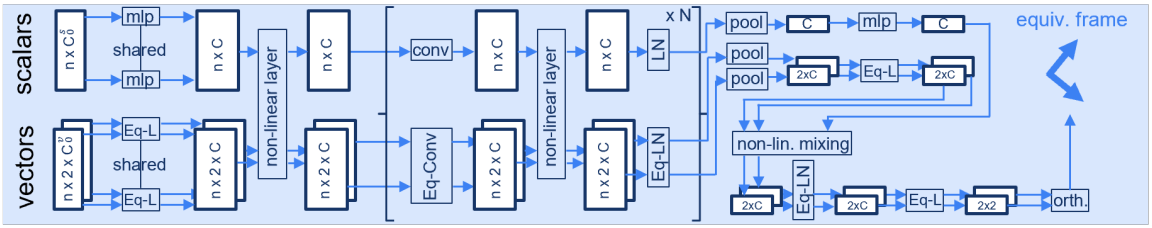

Figure 2: The equivariant network architecture preprocesses the inputs to $n \times C_0^s$ scalars, and $n \times C_0^v$ vectors: Vectors are processed by equivariant linear layers (Eq-L), convolutional layers (Eq-Conv), and normalization layers (Eq-LN), while scalars are separately processed with conventional layers. The vector and scalar features interact only in the non-linear layer in an equivariant way.

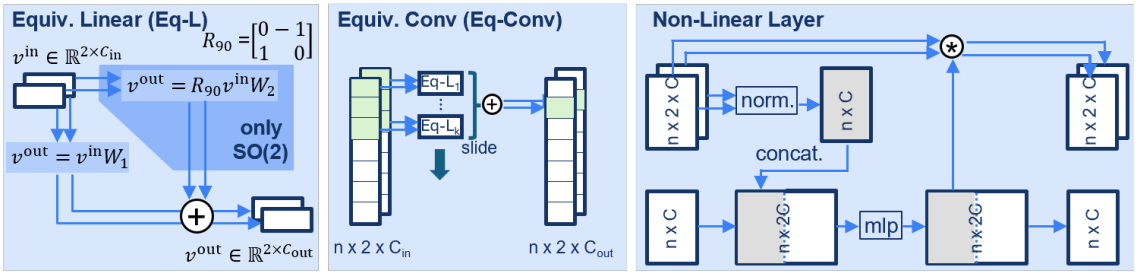

Figure 3: Eq-L (bottom, left) uses two weights W_1, W_2 for SO(2) equivariance, and only W_1 for O(2) equivariance. Eq-L (bottom, middle) uses Eq-L to perform 1-D convolutions over time. The equivariant non-linear layer (bottom, right) mixes the vector and scalar features.

The x and y components of IMU measurements are passed as vector inputs $\mathbb{R}^{n\times 2\times 2}$. The vectors and scalars are then separately passed to the linear layer described in Section **??**. The equivariant network predicting the equivariant frame consists of 1 linear layer, 1 nonlinearity, 1 convolutional block with convolution applied over time, non-linearity, and layer norm. The hidden dimension is 128 and the convolutional kernel is 16 x 1. Finally, the fully connected block of hidden dimension 128 and consisting of linear, nonlinearity, layer norm, and output linear layer follows a pooling over the time dimension. The output of the final linear layer is 2 vectors representing the two bases of the equivariant frame. The input vectors of dimension $\mathbb{R}^{n\times 2\times 2}$ are projected into the invariant space via the equivariant frame resulting in invariant features in $\mathbb{R}^{n\times 4}$. These features are combined with the input scalars and passed as input ($\mathbb{R}^{n\times 6}$) to TLIO or RONIN base architecture. The output of TLIO is invariant 3D displacement and diagonal covariance along the principal axis. The output of RONIN is 2D velocity. The x and y components are back-projected using the equivariant frame to obtain displacement vector d in $\mathbb{R}^2$ and the covariance in the original frame. The covariance is parameterized and processed as mentioned in Section **??**.

$O(2)$- The preprocessing is as described in Section **??** where ω is decomposed to two vectors v_1 and v_2 that have magnitude $|\omega|$. The preprocessed input therefore consists of 3

vectors a, v_1 and v_2. This is then passed to the equivariant network by decoupling the z-axis resulting in vector input $\mathbb{R}^{n\times3\times2}$ which represents 3 vectors in 2D. The scalars passed to the linear layer described in Section **??** consist of the accelerometer z-axis measurement (1), the z component of the two vectors v_1 and v_2 (2), the norm of the vectors (3) and the inner product of the vectors(3) resulting in $\mathbb{R}^{n\times9}$. The network architecture is the same as $SO(2)$ with hidden dimension 64 and 2 convolutional blocks in order to make it comparable in the number of parameters to $SO(2)$ architecture. The invariant features obtained by projecting the three vectors using the equivariant frame are processed as mentioned in Section **??** to obtain 2 vectors in 3D that are fed as input to TLIO and RONIN. The postprocessing is the same as $SO(2)$.

The framework is implemented in Pytorch and all hyperparameters of the base architectures are used to train TLIO and RONIN respectively. The $SO(2)$ architecture has 1821312 while $O(2)$ has 2378368 number of parameters and the base TLIO architecture has 5424646. The baseline TLIO and our methods applied to TLIO were trained on NVIDIA a40 GPU occupying 7-8 GB memory per epoch. The training took 5 mins per epoch over the whole training dataset. We train for 10 epochs with MSE Loss and the remaining 40 epochs with MLE Loss similar to TLIO Liu et al. (2020). RONIN was trained on NVIDIA 2080ti for 38 epochs taking 2 mins per epoch. The loss function used was MSE as mentioned in Herath et al. (2020). The EKF described in TLIO was run on NVIDIA 2080ti with the same initialization and scaling of predicted measurement covariance as in TLIO Liu et al. (2020).

We compare the resource requirements of the SO(2), and O(2) variant of our method coupled with TLIO, with base TLIO without an equivariant frame. We report the floating point operations (FLOPs), the inference time (in milliseconds), and Maximum GPU memory (in GB) during inference, on an NVIDIA 2080 Ti GPU for the neural network averaged over multiple runs to get accurate results. While base TLIO uses 35.5 MFLOPs, 3.5 ms, and 0.383 GB per inference, our SO(2) equivariant method instead uses 531.9 MFLOPs, 4.3 ms, and 0.383 GB per inference. Finally, our O(2) equivariant method uses 638.5 MFLOPs, 4.6 ms, and 0.385 GB per inference. We further evaluate the Maximum GPU memory for the equivariant networks separately and report 0.255 GB per inference for SO(2) equivariant frame prediction and 0.257 GB per inference for O(2) equivariant frame prediction. The Maximum GPU memory is unaffected because the equivariant frame computation utilizes less memory than TLIO.

Finally, we also evaluate our method with a downstream EKF on an NVIDIA 2080 Ti GPU. The EKF incorporates raw IMU measurements for propagation, and displacement measurements from the neural network as measurement updates. For every 20 imu samples, we send the last 200 IMU measurements to the neural network to provide this measurement update. The original TLIO requires 0.492 seconds and 1.113 GB of memory. For the SO(2) variant of our method, we require 0.554 seconds and 1.109 GB of memory to process 1 second of real-world data. For the O(2) variant, we use 0.554 seconds and 1.115 GB of memory, showing that our method is faster than real-time. The increase in memory for the O(2) variant is due to the additional preprocessing step.

With comparable computing resources, our equivariant model outperforms TLIO since we leverage symmetry, which is an intrinsic property in inertial odometry.

A.4. EKF Details

A.4.1. Process Model

The EKF filter states include orientation, translation, velocity, biases of the imu body. The EKF propagation uses raw IMU samples in the local IMU frame, following strap-down inertial kinematics equations:

$$
\begin{aligned}
{}^{w}_{i}\hat{\mathbf{R}}_{\mathbf{k+1}} &= {}^{w}_{i}\hat{\mathbf{R}}_{\mathbf{k}} \exp_{SO(3)}((\omega_{\mathbf{k}} - \hat{\mathbf{b}}_{\mathbf{gk}})\Delta t) \\
{}^{w}\hat{\mathbf{v}}_{\mathbf{k+1}} &= {}^{w}\hat{\mathbf{v}}_{\mathbf{k}} + {}^{w}\mathbf{g}\Delta t + {}^{w}_{i}\hat{\mathbf{R}}_{\mathbf{k}}(\mathbf{a}_{\mathbf{k}} - \hat{\mathbf{b}}_{\mathbf{ak}})\Delta t \\
{}^{w}\hat{\mathbf{p}}_{\mathbf{k+1}} &= {}^{w}\hat{\mathbf{p}}_{\mathbf{k}} + {}^{w}\hat{\mathbf{v}}_{\mathbf{k}}\Delta t + \frac{1}{2}\Delta t^2({}^{w}\mathbf{g} + {}^{w}_{i}\hat{\mathbf{R}}_{\mathbf{k}}(\mathbf{a}_{\mathbf{k}} - \hat{\mathbf{b}}_{\mathbf{a}}\mathbf{k})) \\
\hat{\mathbf{b}}_{\mathbf{g(k+1)}} &= \hat{\mathbf{b}}_{\mathbf{gk}} + \eta_{\mathbf{gdk}} \\
\hat{\mathbf{b}}_{\mathbf{a(k+1)}} &= \hat{\mathbf{a}}_{\mathbf{gk}} + \eta_{\mathbf{adk}}
\end{aligned}
$$

where at timestep k, ${}^{w}_{i}\hat{R}_{k}$ is the orientation estimate of the Kalman filter from IMU frame to the gravity-aligned world frame, $\hat{b}_{gk}$ are the gyroscope biases, Δt is the time interval, ${}^{w}\hat{v}_{k}$ is the velocity estimate, ${}^{w}g$ is the constant gravity vector, $\hat{b}_{ak}$ are the accelerometer biases, ${}^{w}\hat{p}_{k}$ is the position estimate, η_{gdk} and η_{adk} are the IMU noises that are assumed to be normally distributed.

A.4.2. Measurement Model

The measurement model in the EKF uses the displacement estimates provided by the neural network, aligning them in a local gravity-aligned frame to ensure the measurements are decoupled from global yaw information:

$$
\hat{h}(\mathbf{X}) = \mathbf{R}_{\gamma}^{T}(\mathbf{p}_{j} - \mathbf{p}_{i}) = \hat{d}_{ij} + \eta_{ij}
$$

where $\mathbf{R}_{\gamma}$ is the yaw rotation matrix, $\mathbf{p}_{i}$ and $\mathbf{p}_{j}$ are positions of the past and current states, and η_{ij} represents the measurement noise modeled by the network's uncertainty output.

A.4.3. Update Model

The Kalman gain is computed based on the measurement and covariance matrices, and the state and covariance are updated accordingly. The key update equations involve the computation of the Kalman gain ($\mathbf{K}$), updating the state ($\mathbf{X}$), and updating the covariance matrix ($\mathbf{P}$):

$$
\begin{aligned}
\mathbf{K} &= \mathbf{P}\mathbf{H}^{T}(\mathbf{H}\mathbf{P}\mathbf{H}^{T} + \Sigma)^{-1} \\
\mathbf{X} &= \mathbf{X} + \mathbf{K}(\hat{h}(\mathbf{X}) - \hat{d}_{ij}) \\
\mathbf{P} &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}
\end{aligned}
$$

A.5. Evaluation Metrics Definition

We follow most metrics in TLIO Liu et al. (2020) and RONIN Herath et al. (2020), besides MSE loss we reported in the paper. Here we provide the mathematical details of these metrics.

- MSE (m^2): Translation error per sample between the predicted and ground truth displacement averaged over the trajectory. It is computed as $\frac{1}{n}\sum_i^n \| {}^w p_i - {}^w \hat{p}_i \|$. However, it should be noted that MSE mentioned in TLIO Liu et al. (2020) is the same as MSE Loss calculated as the squared error averaged separately for each axis $\frac{1}{n}\sum_i^n \| {}^w p_{i,r} - {}^w \hat{p}_{i,r} \|$ where r is an axis.
- ATE (m): Translation Error assesses the discrepancy between predicted and ground truth (GT) positions across the entire trajectory. It is computed as $\sqrt{\frac{1}{n}\sum_i^n \| {}^w p_i - {}^w \hat{p}_i \|}$
- RTE (m): Following the method described in Cohen and Welling (2016), Relative Translation Error measures the local differences between predicted and GT positions over a specified time window of duration δt (1 minute). $\sqrt{\frac{1}{n}\sum_i^n \| {}^w p_{i+\delta t} - {}^w p_i - ({}^w \hat{p}_{i+\delta t} - {}^w \hat{p}_i) \|}$.
- AYE Absolute Yaw Error is calculated as $\sqrt{\frac{1}{n}\sum_i^n \|\gamma_i - \hat{\gamma}_i\|}$.

A.6. Visualization of TLIO results

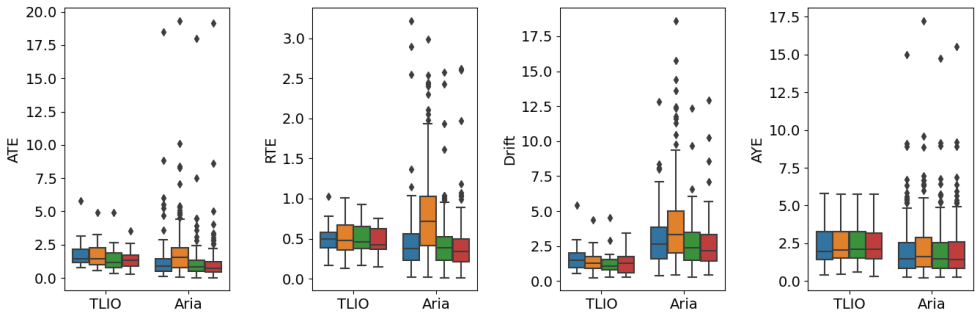

Figure 4: The superior performance of our framework applied to TLIO architecture when compared to baseline TLIO trained with and without augmentations on TLIO and Aria Datasets visualized with a box plot. Blue, Orange, Green and Red indicate TLIO, TLIO-N, TLIO+Eq F. $SO(2)$ and TLIO+Eq F. $O(2)$.

Figure 6 and Figure 7 show only the neural network results compared to ground truth displacements. The ATE and RTE is calculated on the cumulative trajectory obtained form the predicted displacements. Figure 7 is with whisker extended to include the outlier which are commonly calculated as 1.5 * IQR (inter-quartile range). Figure 8 shows the results of EKF without excluding the outliers. We provide more trajectory visualizations of TLIO test data in Figure 5, Figure 9 and Figure 10.

A.7. Augmented TLIO Test Dataset Results and Analysis

We also perform an ablation study on test data augmentation for our model. For neural network results, we apply four random yaw rotations per trajectory and random rotations plus reflection per trajectory. The results are detailed in Table 2. Except for our equivariant

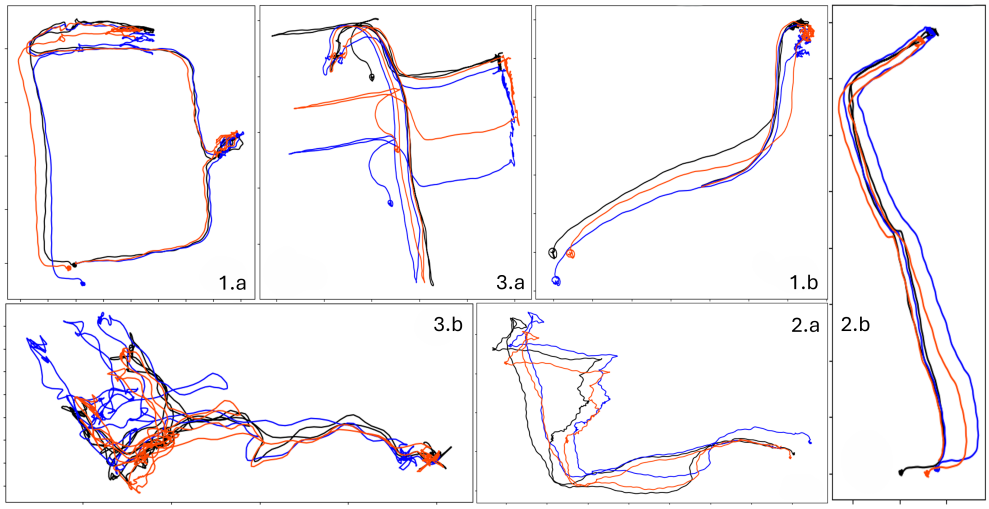

Figure 5: Visualization of final estimated trajectories on TLIO Dataset by baseline TLIO (Blue), our best method applied to TLIO (TLIO+Eq F. $O(2)$)(Red), and the Ground-Truth trajectory (Black). 1.a and 1.b are easy trajectories; 2.a and 2.b are mid-level hard trajectories; 3.a and 3.b are unusual motions not present in the training set are performed.

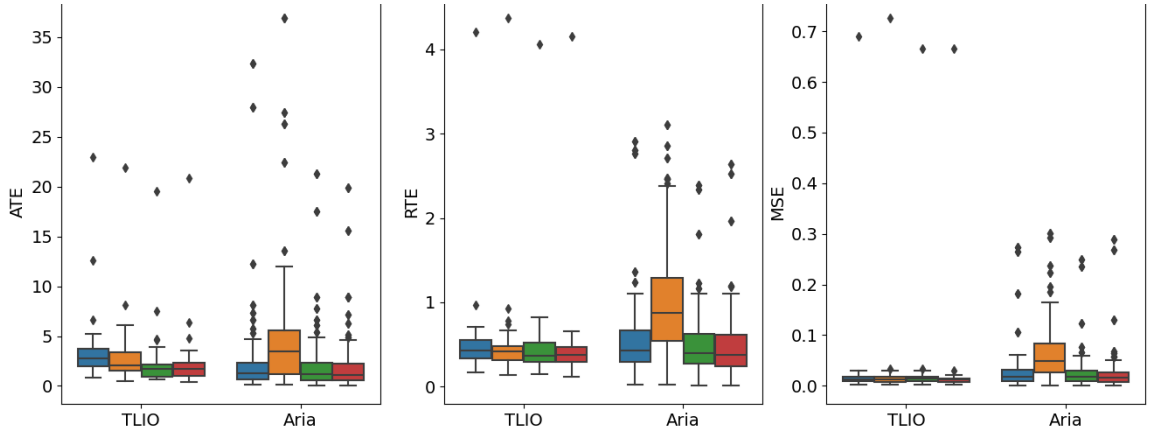

Figure 6: The superior performance of our framework applied to TLIO architecture when compared to baseline TLIO trained with and without augmentations on TLIO and Aria Datasets visualized with a box plot. Blue, Orange, Green and Red indicate TLIO, TLIO-N, TLIO+Eq F. $SO(2)$ and TLIO+Eq F. $O(2)$. ATE, RTE and MSE indicate ATE*, RTE* and MSE* corresponding to only the NN results.

model, all other methods show decreased performance compared to their results on non-

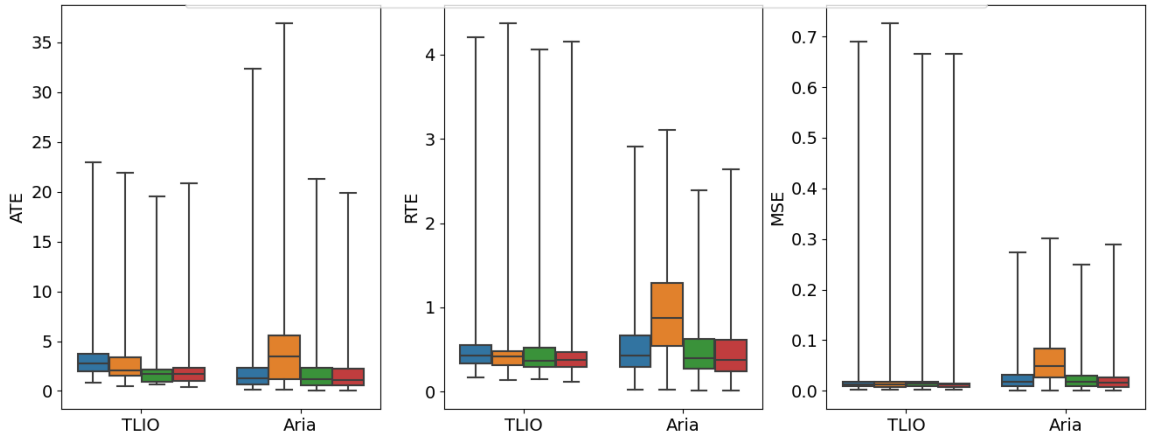

Figure 7: The superior performance of our framework applied to TLIO architecture when compared to baseline TLIO trained with and without augmentations on TLIO and Aria Datasets visualized with a box plot. Blue, Orange, Green and Red indicate TLIO, TLIO-N, TLIO+Eq F. $SO(2)$ and TLIO+Eq F. $O(2)$. ATE, RTE and MSE indicate ATE*, RTE* and MSE* corresponding to only the NN results. The whisker is extended to 1.5 * IQR (inter-quartile range).

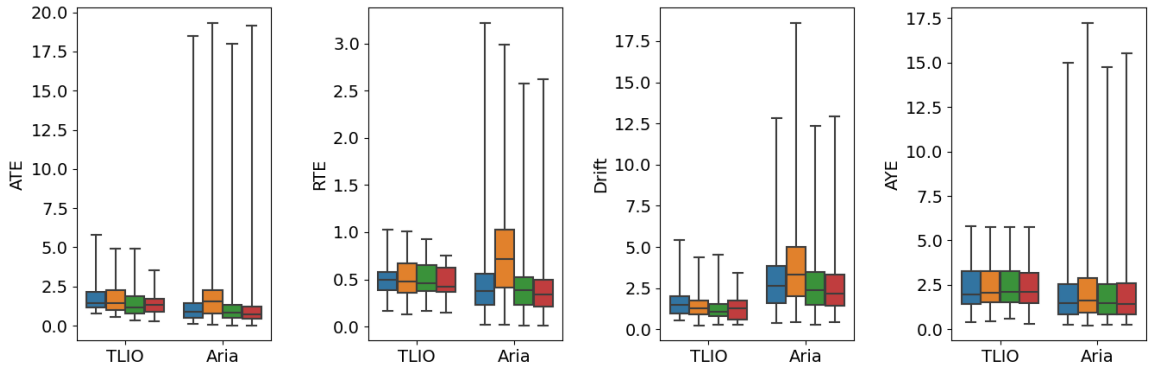

Figure 8: The superior performance of our framework applied to TLIO architecture when compared to baseline TLIO trained with and without augmentations on TLIO and Aria Datasets visualized with a box plot. Blue, Orange, Green and Red indicate TLIO, TLIO-N, TLIO+Eq F. $SO(2)$ and TLIO+Eq F. $O(2)$. The whisker is extended to 1.5 * IQR (inter-quartile range).

augmented test data, whereas our model maintains consistent performance and outperforms the other methods.

For the Extended Kalman Filter (EKF) results, we augment the test data using random $SO(3)$ rotations. Notably, we do not include reflections due to the structural constraints of the Kalman filter. As shown in Table 3, despite the TLIO-NQ model outperforming ours in non-

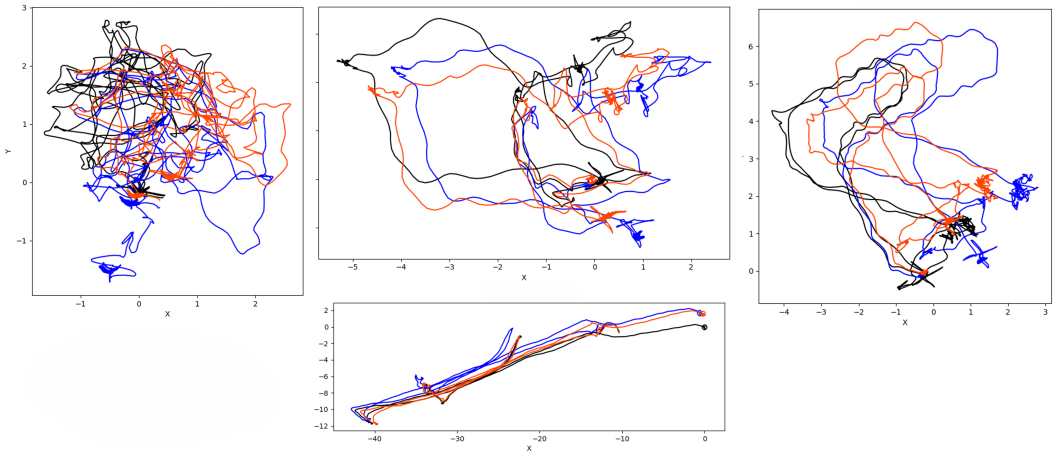

Figure 9: More Visualizations of final estimated trajectories on TLIO Dataset by baseline TLIO (Blue), our best method applied to TLIO (TLIO+Eq F. $O(2)$)(Red), and the Ground-Truth trajectory (Black).

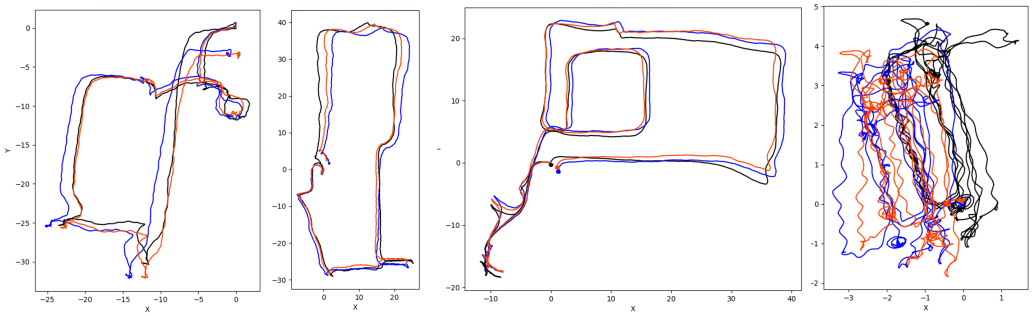

Figure 10: More Visualizations of final estimated trajectories on TLIO Dataset by baseline TLIO (Blue), our best method applied to TLIO (TLIO+Eq F. $O(2)$)(Red), and the Ground-Truth trajectory (Black).

augmented tests on ATE metrics. Our model exceeds TLIO-NQ on the augmented dataset. Our approach not only sets a new benchmark but also maintains consistent performance across random rotations.

A.8. Visualization of RONIN

The visualization of trajectories in RONIN is displayed in Figure 11.

A.9. Ablation Study

In this section, we investigate and motivate the necessity for incorporating equivariance in inertial odometry, the choice of equivariant architecture and covariance. We present all the

	Rotations			Rotations + Reflections		
Model	MSE*	ATE*	RTE*	MSE*	ATE*	RTE*
TLIO	0.0327	3.3180	0.5417	0.0347	2.9110	0.5654
TLIO-N	0.2828	27.7797	3.1390	0.2989	23.4839	3.1313
Deeper TLIO	0.0306	3.0264	0.5300	0.0332	2.3028	0.5592
TLIO-NQ	0.0302	2.6379	0.5025	0.0331	2.3212	0.5446
TLIO-PCA	0.2286	21.3795	2.5288	0.2467	10.1660	2.2283
†Eq F. SO(2)	0.0319	2.3218	0.4957	0.0339	1.8664	0.5178
†Eq F. O(2)	0.0298	2.3305	0.4719	0.0298	1.6418	0.4361

Table 2: Ablation Study For Neural Network with Random Rotation and Reflection Transformation (4 per trajectory) on TLIO test dataset. † represents TLIO+. A lower error indicates a better model. The lowest values are annotated with Red. Our proposed methods are in bold.

Exp	ATE	RTE	Drift	AYE
TLIO	1.6744	0.4944	1.5526	2.7290
TLIO-N	10.3005	3.6263	2.9501	3.3684
Deeper TLIO	1.6447	0.5466	1.2767	2.7279
TLIO-NQ	1.4924	0.5119	1.2721	2.7109
TLIO-PCA	8.5787	2.9962	2.0872	3.0183
†Eq F. SO(2)	1.4850	0.4901	1.3029	2.7615
†Eq F. O(2)	1.4316	0.4592	1.3096	2.7250

Table 3: Results of evaluation of EKF with Random Rotation Transformations (4 per trajectory) on TLIO test dataset (*i.e.*, results on augmentated test dataset). † represents TLIO+. A lower error indicates a better model. The lowest values are annotated with Red. Our proposed methods are in bold.

ablation on TLIO in Table 4 for the neural network and the overall performance when NN is integrated with the EKF. Appendix A.7 contains the results of evaluating all the above models separately on a test dataset augmented with rotations and/or reflections. Appendix A.11 presents the ablation on IMU input sequence length and lastly, in Appendix A.12 we present sensitivity analysis to gravity direction using 5 discrete angles.

Baseline Ablation: **Is yaw augmentation needed when the input is in a local gravity-aligned frame?** We trained TLIO both with and without yaw augmentation using identical hyperparameters and the results in Table 4 revealed that augmentation enhances the network's generalization, improving all metrics for the Aria dataset with the lowest margin of 10% on AYE and highest margin of 65% for MSE*. This underscores the importance of equivariance for network generalization. **Does a Deeper TLIO with a comparable number of parameters match the performance of equivariant methods?** We enhanced the residual depth of the original TLIO architecture from 4 residual blocks of depth 2 each to 4 residual blocks with depth 3 each to align its number of parameters with our Eq

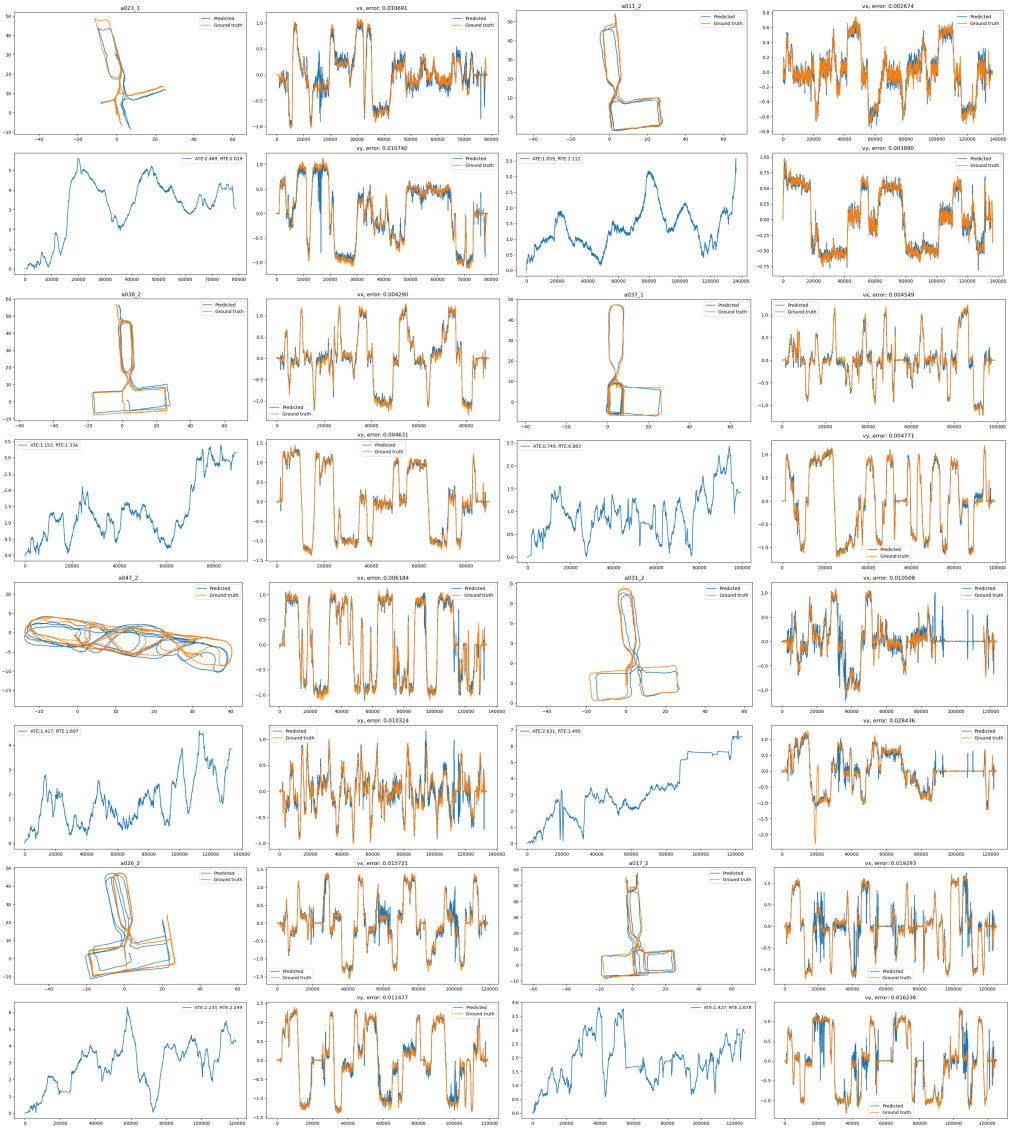

Figure 11: Visualization of RONIN Unseen Test Dataset Trajectories for our best method applied to RONIN, RONIN+Eq F. $O(2)$.

F. $SO(2)$ model. Despite having fewer parameters due to the removal of the orthogonal basis in the $SO(2)$ vector neuron-based architecture, the Eq F. $O(2)$ model still outperformed the augmented TLIO. The data from Table 4 demonstrate that merely increasing the network's size, without integrating true equivariance, is insufficient for achieving precise inertial odometry.

	TLIO Dataset						Aria Dataset					
Model	MSE*	ATE	ATE*	RTE	RTE*	AYE	MSE*	ATE	ATE*	RTE	RTE*	AYE
TLIO	3.242	1.812	3.722	0.500	0.551	2.376	5.322	1.285	2.102	0.464	0.521	2.073
TLIO-N	3.333	1.722	3.079	0.521	0.542	2.366	15.248	1.969	4.560	0.834	0.977	2.309
Deeper TLIO	3.047	1.613	2.766	0.524	0.519	2.397	2.403	1.189	2.541	0.472	0.540	2.081
TLIO-NQ	3.008	1.429	2.443	0.495	0.496	2.411	2.437	1.213	2.071	0.458	0.508	2.096
TLIO-PCA	3.473	1.506	2.709	0.523	0.535	2.459	6.558	1.717	4.635	0.771	0.976	2.232
Eq CNN	3.194	1.580	3.385	0.564	0.610	2.394	8.946	3.223	6.916	1.091	1.251	2.299
TLIO + Eq F. $SO(2)$+S	3.331	1.626	2.796	0.524	0.536	2.440	2.591	1.146	2.067	0.466	0.517	2.089
TLIO + Eq F. $SO(2)$+P	3.298	1.842	2.652	0.588	0.523	2.537	2.635	1.592	2.303	0.585	0.539	2.232
TLIO + Eq F. SO(2)	3.194	1.480	2.401	0.490	0.501	2.428	2.457	1.178	1.864	0.449	0.484	2.084
TLIO + Eq F. $O(2)$+S	3.061	1.484	2.474	0.462	0.481	2.390	2.421	1.175	1.804	0.421	0.458	2.043
TLIO + Eq F. $O(2)$+P	2.990	1.827	2.316	0.578	0.478	2.534	2.373	1.755	1.859	0.564	0.468	2.223
TLIO + Eq F. O(2)	2.982	1.433	2.406	0.458	0.478	2.389	2.304	1.118	1.849	0.416	0.465	2.059

Table 4: Ablation Studies. **TLIO-NQ** is TLIO with non-equivariant frames. **TLIO-PCA** is TLIO using PCA to predict handcrafted frames. **TLIO-N** is TLIO trained without yaw augmentations. **Eq F. $SO(2)$+S** is Equivaraiant frame with circular covariance. **Eq F. $SO(2)$+P** is $SO(2)$ equivariant networks with pearson parameterized covariance. **Eq CNN** is fully equivariant CNN. The same naming conventions apply to the $O(2)$ variants. A lower error indicates a better model. The lowest values are annotated with Red, followed by Orange and Yellow respectively. Our proposed methods are in bold.

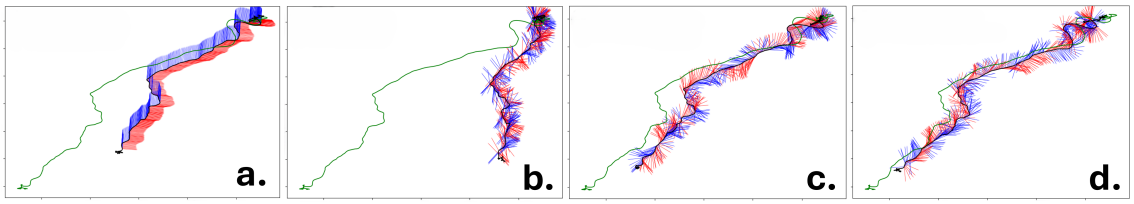

Figure 12: Visualization of one neural integrated trajectory in the Aria Dataset: the green trajectory shows the ground truth, the black trajectories show predictions, and the blue and red vectors represent the predicted frame's basis vectors. a) Non-equivariant MLP b) PCA c) Eq F. *SO(2)* d) Eq F. *O(2)*

Frame Ablation: **Can a non-equivariant MLP predict meaningful frames?** We trained TLIO with augmentation and identical hyperparameters alongside an additional MLP mirroring the architecture of our method to predict a frame and term this baseline TLIO-NQ. We observed that TLIO-NQ tends to overfit to the TLIO dataset, and the predicted frames were not meaningful, as illustrated in Figure 12. **Can frames predicted using PCA (handcrafted equivariant frame) achieve the same performance?** PCA frames underperform on the Aria dataset and perform worse than the original TLIO, likely due to PCA's noise sensitivity and the production of non-smooth frames, as shown in Figure 12. Additionally, PCA cannot distinguish between $SO(2)$ and $O(2)$ transformations. Figure 12 also shows that $O(2)$ does not have frames as smooth as $SO(2)$ as the reflected bends have reflected frames.

Architecture Ablation: **Does a fully equivariant architecture perform better than our frame-based approach ?** We trained a fully equivariant convolutional network using the basic layers described in Section **??**. As shown in Table 4, our frame-based methods are more effective and efficient than the equivariant CNN. We believe the fully equivariant architecture is overly restrictive, while our approach leverages the power of scalars and conventional backbones. Additionally, our method integrates easily with any state-of-the-art neural inertial navigation system, unlike the fully equivariant architecture, which requires redesigning.

Covariance Ablation: **Do we need equivariant covariance?** We investigated the importance of equivariant covariance for both $SO(2)$ and $O(2)$ groups, as described in Section **??**(See Appendix A.1.3 for more details on covariance parameterizations.). In Table 4, the models Eq F. $SO(2)$+S and Eq F. $O(2)$+S are trained with invariant covariance. The results show that equivariant covariance yields better performance, especially when combined with EKF, as it provides a more accurate estimate of the measurement covariance. **Can a full covariance matrix predicted via Pearson parameterization further improve the performance?** In Table 3, Eq F. $SO(2)$+P and Eq F. $O(2)$+P are outperformed by our model in most cases. As mentioned in Section **??**, this experimental result indicates that by aligning the principle axis of the covariance into the basis of the equivariant frame, we intrinsically force the covariance in the equivariant frame to be diagonal, which reduces the ambiguity while training. Prediction of diagonal covariance improves stabilization and convergence in the optimization process as stated in TLIO. The visualization of covariance consistency of our Eq F. $O(2)$ model is in Appendix A.10.

A.10. Covariance Consistency

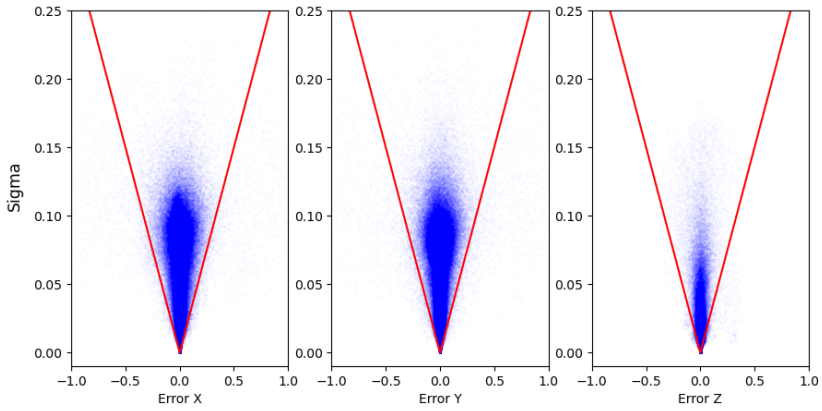

Figure 13: Consistency of Covariance Prediction in the Invariant Space for TLIO test dataset

Similar to TLIO Liu et al. (2020), we plot the prediction error against standard deviation (σ) predicted by the network in the invariant space. As seen in Figure 14 and Figure 13 the covariance prediction of our method is consistently within the 3-σ depicted by the red

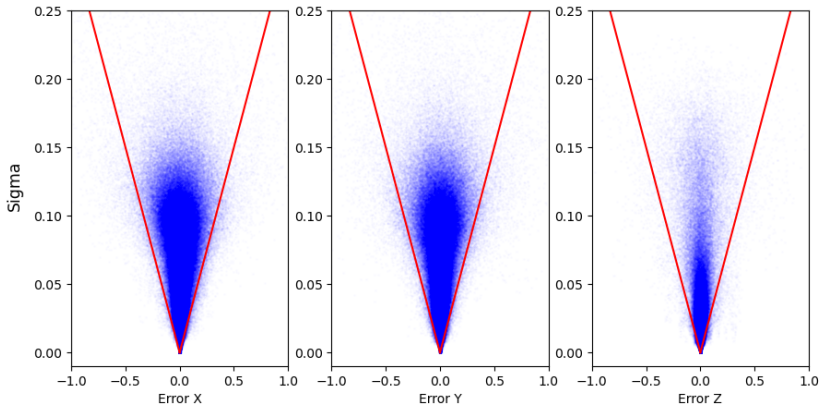

Figure 14: Consistency of Covariance Prediction in the Invariant Space for Aria dataset

lines. These results show that our diagonal covariance prediction in the invariant space is consistent.

A.11. Ablation on IMU sequence length

We aligned the sequence length with baseline models for fair comparison. However, in this Section, we ablate on the sequence length as shown in Table 5 and Table 6. Table 5 varies sequence lengths and displacement prediction windows (e.g., 0.5s displacement with 0.5s of 200Hz IMU data results in a sequence length of 100). Table 6 fixes the prediction window at 1s and varies the context window (e.g., a 2s context window with 200Hz IMU data results in a sequence length of 400). Our results confirm TLIO Liu et al. (2020) Sec. VII A.1: increasing the context window reduces MSE but not ATE. A lower MSE loss over the same displacement window does not translate to a lower ATE. Thus, the addition of the equivariant framework does not change the characteristics of the base (off-the-shelf) model used.

A.12. Sensitivity analysis to gravity direction perturbation

Similar to Wang et al. (2023), which indicates that the equivariance of $SO(2)$ can even help the rotation around another axis which is close to z, we believe that embedding equivariance wouldn't harm the performance of the model when there is a slight perturbation which is inline with the experimental results as seen in Table 7.

Table 7 presents the sensitivity analysis to gravity direction perturbation, applied for 5 discrete angles. We also present results for Eq F. $O(2)$ model trained without the gravity direction perturbation of (-5°,5°) during training. We observe the same trend of stability in MSE* as reported in TLIO Liu et al. (2020) when trained with gravity direction perturbation.

Exp	Displacement Window (s)	TLIO Dataset			Aria Dataset		
		MSE*	ATE*	RTE*	MSE*	ATE*	RTE*
TLIO	0.5	1.132	2.029	0.340	1.038	1.489	0.332
TLIO	1	3.242	3.722	0.551	5.322	2.103	0.521
TLIO	2	9.862	5.102	0.944	6.717	3.452	0.970
†Eq F. $SO(2)$	0.5	1.124	0.711	0.175	1.040	0.673	0.190
†Eq F. $SO(2)$	1	3.194	2.401	0.501	2.457	1.864	0.484
†Eq F. $SO(2)$	2	10.019	3.862	0.797	6.569	2.745	0.774
†Eq F. $O(2)$	0.5	1.040	0.595	0.136	1.002	0.589	0.148
†Eq F. $O(2)$	1	2.982	2.406	0.478	2.304	1.849	0.465
†Eq F. $O(2)$	2	9.804	4.268	0.762	6.112	2.556	0.709

Table 5: Results for ablation on changing prediction displacement window on TLIO architecture. † represents TLIO+

Exp	Context Window (s)	TLIO Dataset			Aria Dataset		
		MSE*	ATE*	RTE*	MSE*	ATE*	RTE*
TLIO	1	3.242	3.722	0.551	5.322	2.103	0.521
TLIO	2	3.199	2.555	0.511	3.790	2.895	0.713
TLIO	3	3.284	4.463	0.617	3.511	3.014	0.738
†Eq F. $SO(2)$	1	3.194	2.401	0.501	2.457	1.864	0.484
†Eq F. $SO(2)$	2	2.886	1.837	0.429	2.187	1.533	0.444
†Eq F. $SO(2)$	3	2.790	3.090	0.492	1.986	1.684	0.447
†Eq F. $O(2)$	1	2.982	2.406	0.478	2.304	1.849	0.465
†Eq F. $O(2)$	2	2.382	1.895	0.367	1.307	1.382	0.338
†Eq F. $O(2)$	3	2.161	2.083	0.366	0.974	1.672	0.366

Table 6: Results for ablation on changing context window with fixed displacement window of 1s on TLIO architecture. † represents TLIO+

A.13. Social Impact

This work aims to utilize deep learning to mitigate drift in inertial integration for purely inertial odometry, thereby enhancing navigation efficiency and reducing costs. While our research directly contributes positively to navigation solutions and does not have inherently negative social applications, it is important to note that improved tracking and navigation capabilities could potentially be utilized for surveillance purposes, which may raise privacy concerns.

Exp	Gravity Direction Perturbation (in degrees)	TLIO Dataset		
		MSE*	ATE*	RTE*
†Eq F. $SO(2)$	0	3.194	2.401	0.501
†Eq F. $SO(2)$	2	3.201	2.409	0.500
†Eq F. $SO(2)$	4	3.206	2.404	0.498
†Eq F. $SO(2)$	6	3.241	2.442	0.501
†Eq F. $SO(2)$	8	3.298	2.502	0.506
†Eq F. $O(2)$‡	0	2.982	2.406	0.478
†Eq F. $O(2)$‡	2	3.198	2.663	0.498
†Eq F. $O(2)$‡	4	3.742	3.292	0.559
†Eq F. $O(2)$‡	6	4.505	4.228	0.659
†Eq F. $O(2)$‡	8	5.433	5.218	0.768
†Eq F. $O(2)$	0	2.982	1.811	0.332
†Eq F. $O(2)$	2	2.988	1.742	0.321
†Eq F. $O(2)$	4	3.010	1.718	0.308
†Eq F. $O(2)$	6	3.060	1.680	0.293
†Eq F. $O(2)$	8	3.095	1.650	0.283

Table 7: Results for ablation on changing prediction displacement window on TLIO architecture. † represents TLIO+. ‡ implies the network was trained without gravity direction perturbation.