MARIO: Motion-Augmented Real-Time Multi-Sensor Inertial Odometry

Supplementary Material

1. PoseNet Evaluation Experiment

To evaluate the quality of our PoseNet, we compare PoseNet with Ego4o [1], which includes an Ego4o-IMU model that takes head IMU as input, and an Ego4o model that additionally takes egocentric RGB images and motion description text as input. We report the Mean Per-Joint Position Error (MPJPE) and Procrustes-Aligned Mean Per-Joint Position Error (PA-MPJPE) of Ego4o-IMU and compare them with those of our PoseNet; both models are evaluated on the Nymeria dataset. Since Ego4o was trained and evaluated on its own curated splits, the numbers may not be directly comparable. We include them only to contextualize the scale of pose accuracy achievable with richer modalities. From Table 1, our method achieves better MPJPE and PA-MPJPE (in mm) than Ego4o-IMU and comparable results to Ego4o, despite Ego4o using more modalities (IMU + RGB + language).

Method	MPJPE	PA-MPJPE
Ego4o-IMU	123.80	85.45
Ego4o	99.68	72.60
PoseNet (ours)	101.24	78.71

Table 1. Comparison of our PoseNet, Ego4o-IMU [1] and Ego4o [1] on MPJPE and PA-MPJPE.

To further demonstrate that the pose prior is dataset-agnostic, we also train our PoseNet on the AMASS dataset and evaluate it on the Nymeria dataset. We observe that PoseNet trained on AMASS improves the performance of both the AirIO and TLIO models.

2. Coordinate Details

We convert all recordings to a Central Pupil Frame (CPF) for consistent training and evaluation. Raw SLAM data provides positions and orientations in an arbitrary world frame, while IMU measurements and velocities are in the device's body frame. This mismatch prevents proper velocity integration during training. We resolve this by choosing a world frame that is expressed in CPF (+X left, +Y up, +Z forward).

CPF (Central Pupil Frame) is a rigid device-centric coordinate system with +X left, +Y up, +Z forward. We describe how raw device data is transformed into the CPF coordinate frame. At a chosen reference time, we (1) project the device's CPF forward direction onto the horizontal plane to define the forward axis of the new world frame, and (2) align the vertical axis with gravity. This creates a unified, gravity-aligned frame that supports geometrically consistent transformations between the device's body frame and the world frame.

Table 2. Results on the Nymeria dataset. Pose_N and Pose_A denotes PoseNet trained on the Nymeria and AMASS training sets, respectively

Model	ATE(m)	RTE-5s (m)	RTE-1s (m)	Drifting(%)
AirIO	6.85	0.363	0.099	3.56
AirIO + Pose_A	5.61	0.315	0.094	3.48
AirIO + Pose_N	5.22	0.323	0.085	2.35
TLIO	10.19	0.322	0.103	6.46
TLIO + Pose_A	7.82	0.309	0.099	4.69
TLIO + Pose_N	7.97	0.281	0.096	4.94

Stage 1: Device → CPF. Using Aria calibration:

$$\mathbf{v}_{\text{CPF}} = R_{\text{CPF}\leftarrow\text{device}}\,\mathbf{v}_{\text{device}}, \tag{1}$$

$$R_{\text{world}\leftarrow\text{CPF}}(t) = R_{\text{world}\leftarrow\text{device}}(t)\,R_{\text{device}\leftarrow\text{CPF}}, \tag{2}$$

$$R_{\text{device}\leftarrow\text{CPF}} = R_{\text{CPF}\leftarrow\text{device}}^{\top}. \tag{3}$$

Stage 2: CPF → CPF-World (gravity-aligned, consistent heading). Choose a reference time t_0 (30 s after sequence start to allow SLAM to settle). Compute the CPF forward direction in world coordinates:

$$\mathbf{z}_{\text{CPF}}^{\text{world}} = R_{\text{world}\leftarrow\text{CPF}}(t_0)\,[0,\,0,\,1]^{\top}. \tag{4}$$

Project to the horizontal plane and normalize:

$$\hat{\mathbf{z}}_{\text{hor}} = \frac{[\,z_x,\,z_y,\,0\,]^{\top}}{\|[\,z_x,\,z_y,\,0\,]\|}, \quad (z_x, z_y, \cdot) = \mathbf{z}_{\text{CPF}}^{\text{world}}. \tag{5}$$

Define CPF-world axes (expressed in world coordinates):

$$\begin{aligned} \mathbf{y}_{\text{cpfw}} &= [0,\,0,\,1]^{\top}, \\ \mathbf{z}_{\text{cpfw}} &= \hat{\mathbf{z}}_{\text{hor}}, \\ \mathbf{x}_{\text{cpfw}} &= \mathbf{y}_{\text{cpfw}} \times \mathbf{z}_{\text{cpfw}}. \end{aligned} \tag{6}$$

Assemble the rotation with columns as the axes in (6):

$$R_{\text{world}\leftarrow\text{cpfw}} = \big[\,\mathbf{x}_{\text{cpfw}}\;\;\mathbf{y}_{\text{cpfw}}\;\;\mathbf{z}_{\text{cpfw}}\,\big]. \tag{7}$$

Transform positions by rotation and centering at $\mathbf{p}_{\text{ref}}$:

$$\mathbf{p}_{\text{cpfw}}(t) = R_{\text{world}\leftarrow\text{cpfw}}^{\top}\big(\mathbf{p}_{\text{world}}(t) - \mathbf{p}_{\text{ref}}\big). \tag{8}$$

3. Additional Trajectory Visualization

In Figure 1, we visualize the predicted trajectories of the base model, the base model + pose, and the base model +

all for four different methods—AirIO, TLIO, EqNIO, and RoNIN-LSTM—on the Nymeria dataset. We observe that, for each model, +pose visibly reduces drift relative to the base model, and +all further aligns the trajectory around ground truth.

References

[1] Jian Wang, Rishabh Dabral, Diogo Luvizon, Zhe Cao, Lingjie Liu, Thabo Beeler, and Christian Theobalt. Ego4o: Egocentric human motion capture and understanding from multimodal input, 2025. 1

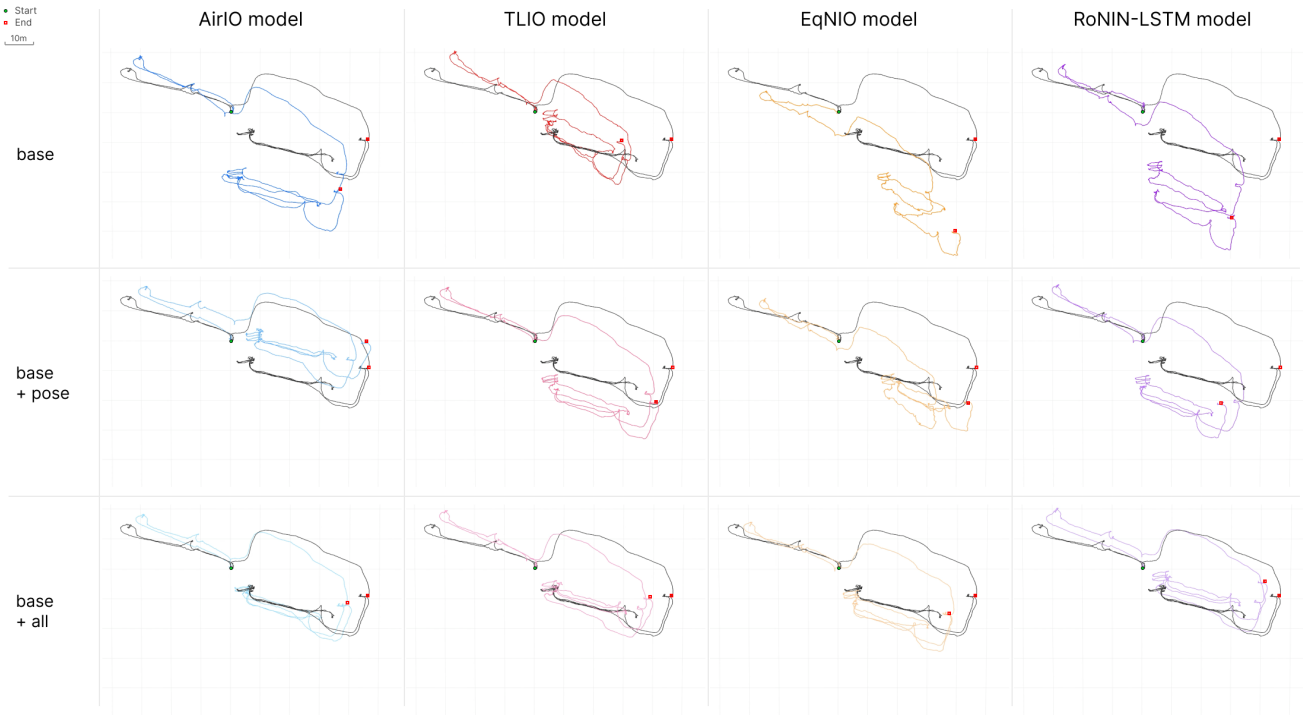

Figure 1. We show the trajectory predictions alongside the ground truth for AirIO, TLIO, EqNIO, and RoNIN-LSTM on one sequence from the Nymeria dataset.