HUMANOIDPOSE: MONOCULAR 3D POSE ESTIMATION FOR HUMANOID ROBOTS

Anonymous authors
Paper under double-blind review

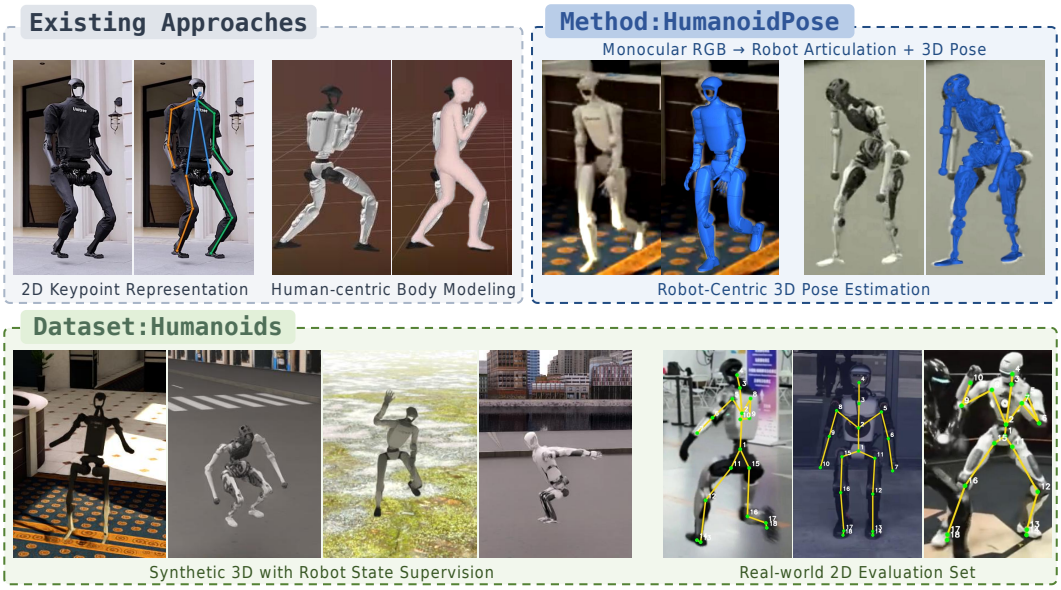


Figure 1: **Overview.** To address the gap in monocular 3D humanoid robot pose estimation(HRPE), we introduce **Humanoids**, a dedicated dataset and benchmark spanning 12 humanoid embodiments, with a synthetic split for 3D supervision and a real-world split for evaluation, and **HumanoidPose**, a framework that recovers robot articulation and camera-relative pose from monocular observations.

ABSTRACT

As humanoid robots increasingly enter real-world environments, recovering their 3D whole-body pose from monocular imagery is becoming increasingly important for external state perception, motion analysis, and robot learning. However, Humanoid Robot Pose Estimation (**HRPE**) lies at the intersection of conventional Robot Pose Estimation (RPE) and Human Pose Estimation (HPE), while not being fully addressed by either paradigm: RPE primarily focuses on non-humanoid robots, particularly fixed-base manipulators, whereas HPE is built upon human-specific skeletal topologies, body models, and motion priors. Despite recent progress in humanoid 2D keypoint detection and HPE adaptation, humanoid-native 3D pose recovery remains insufficiently studied, and a dedicated benchmark with suitable training data and evaluation protocols is still lacking. To address this gap, we introduce **Humanoids**, a benchmark for monocular 3D humanoid robot pose estimation, and propose **HumanoidPose**, a framework for this task. Humanoids provides synthetic 3D supervision and real-world evaluation across diverse humanoid embodiments, motions, and visual conditions, while HumanoidPose combines human-inspired visual perception with robot **articulation modeling** and explicit **kinematic grounding**. Experiments on *Humanoids* and the *DHRP-H1* benchmark show that *HumanoidPose* outperforms representative HPE and RPE approaches in articulation recovery, 3D pose estimation, and real-world evaluation. Ablation studies further show the contribution of robot-native articulation modeling and explicit kinematic grounding. The dataset and code will be publicly released to support future research in the community.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Humanoid robots are increasingly entering real-world environments shared with humans (Luo et al., 2026; Bjorck et al., 2025). In such settings, understanding their whole-body state from external visual observations becomes important for making robot behavior observable and interpretable to people and surrounding systems. Beyond external state perception, motion analysis, and human–robot interaction, such visual understanding may also facilitate robot learning and skill transfer from the observed behavior of other humanoid robots (Fu et al., 2024; Radosavovic et al., 2024; Allshire et al., 2025; Mao et al., 2024; Xie et al., 2025; Liao et al., 2025).

Despite its apparent similarity to existing pose-estimation problems, Humanoid Robot Pose Estimation (**HRPE**) lies at the intersection of conventional Robot Pose Estimation (RPE) and Human Pose Estimation (HPE), while not being fully addressed by either paradigm. RPE methods primarily focus on non-humanoid robots, particularly fixed-base manipulators, and do not readily handle the floating-base, freely moving, high-DoF whole-body motion of humanoids (Labbé et al., 2021; Ban et al., 2024; Tian et al., 2024; Goswami et al., 2025). HPE methods are closer in motion structure, but are built upon human-specific skeletal topologies, joint definitions, body models, and motion priors that do not directly correspond to robot articulation (Goel et al., 2023; Loper et al., 2015). Existing humanoid-specific efforts mainly focus on 2D keypoint detection, specialized sensing setups, or the direct application of existing HPE methods to humanoid robots (Heo et al., 2024; Cho et al., 2024; Young et al., 2025; Li et al., 2026), leaving monocular recovery of robot articulation and root pose insufficiently studied. As summarized in Table 2, progress is further limited by the lack of datasets and benchmarks that jointly provide monocular RGB, robot joint states, root pose, and accurate 3D supervision while covering multiple humanoid embodiments.

To address these gaps, we introduce **Humanoids**, a dataset and benchmark for monocular 3D humanoid robot pose estimation, and propose **HumanoidPose**, a framework for this task.

Humanoids covers **12** humanoid embodiments such as Unitree G1, PND AdamLite, and Booster T1, with diverse motions and visual conditions. It contains 3,706 source motions and 6,885 synthetic clips (**732K RGB frames**) with synchronized robot joint states, root poses, and FK-derived 3D keypoints, together with a real-world evaluation set of **365 clips** with manually verified 2D annotations. The synthetic annotations are generated and verified through geometric **consistency checks** based on forward kinematics and reprojection, while the real-world annotations undergo **quality control** through temporal and kinematic consistency validation. A canonical 2D keypoint representation provides a unified visual interface for different humanoid embodiments.

HumanoidPose directly estimates robot **articulation** and root pose from monocular RGB. Predicting 3D keypoints alone does not explicitly recover the underlying mechanical joint states. Modeling robot articulation and root pose instead allows the recovered state to be mapped to 3D geometry through embodiment-specific forward kinematics, linking visual pose recovery to the robot's mechanical structure. Target-centric visual features and canonical 2D keypoints are fused through keypoint-conditioned **joint queries** to recover robot articulation, while visual, structural, and bounding-box cues jointly initialize the root pose. The state is refined through explicit **robot kinematics** and reprojection, without human body models or human-to-robot retargeting.
Experiments on *Humanoids* and the external *DHRP-H1* benchmark demonstrate that *HumanoidPose* outperforms representative HPE and RPE approaches across robot articulation recovery, 3D pose estimation, and real-world evaluation.

Our main contributions are summarized as follows:

- We formulate monocular 3D Humanoid Robot Pose Estimation (**HRPE**) as a distinct perception problem and introduce **Humanoids**, a dataset and benchmark providing synthetic 3D supervision and real-world evaluation for diverse humanoid embodiments, motions, and visual conditions.
- We propose **HumanoidPose**, a robot-centric framework that directly estimates robot articulation and root pose from monocular RGB, combining keypoint-conditioned articulation prediction with explicit kinematic grounding and bounded refinement.
- We evaluate representative RPE and HPE approaches under a unified HRPE protocol, including synthetic-to-real evaluation. Controlled analyses demonstrate the effectiveness of direct robot articulation modeling and explicit kinematic grounding.

2 Related Work

Table 1: **Comparison with related methods.** Global pose denotes global root pose; robot kinematics denotes explicit use of the robot kinematic model.

Method	Formulation			Capabilities		
	Target	Input	Representation	Temporal	Global Pose	Robot Kinematics
Human Pose Estimation						
GVHMR (Shen et al., 2024)	Human	RGB video	SMPL-X + trajectory	✓	✓	-
WHAM (Shin et al., 2024)	Human	RGB video	SMPL + trajectory	✓	✓	-
HMR2 (Goel et al., 2023)	Human	RGB crop	SMPL	-	✓	-
Robot Pose Estimation						
HoRoPose (Ban et al., 2024)	Non-Hum.	RGB + intrinsics	Joint state	-	✓	✓
RoboPose (Labbé et al., 2021)	Non-Hum.	RGB + intrinsics	Joint state	-	✓	✓
RoboPEPP (Goswami et al., 2025)	Non-Hum.	RGB + intrinsics	Joint state	-	✓	✓
(Heo et al., 2024)	Humanoid	RGB image	2D keypoints	-	-	-
HumanoidPose (Ours)	Humanoid	RGB video	**Joint state**	✓	✓	✓

2.1 Robot Pose Estimation

Robot Pose Estimation (RPE) aims to recover robot articulation and spatial pose from visual observations. Existing methods explore render-and-compare optimization, keypoint-based 3D lifting, holistic feed-forward pose estimation, and self-supervised visual representation learning (Labbé et al., 2021; Tian et al., 2024; Ban et al., 2024; Goswami et al., 2025). However, existing RPE methods primarily target non-humanoid robots, particularly fixed-base manipulators, and do not readily extend to recovering robot articulation and root pose for floating-base humanoids undergoing high-DoF whole-body motion (Table 1).

Humanoid 2D keypoint estimation has been studied (Heo et al., 2024; Young et al., 2025), while full-body pose estimation with head-worn cameras for digital human-augmented robotic telepresence has been investigated (Cho et al., 2024). However, these approaches focus on 2D landmarks or specialized sensing, rather than monocular recovery of 3D robot articulation states and camera-relative global pose.

2.2 Human Pose and Motion Recovery

Human Pose Estimation (HPE) and Human Mesh Recovery (HMR) are closer to humanoid pose estimation in whole-body motion structure. Methods built on human-body models such as SMPL (Loper et al., 2015) include optimization-based fitting such as SMPLify (Bogo et al., 2016), video-based recovery such as VIBE (Kocabas et al., 2020), and world-grounded methods such as WHAM (Shin et al., 2024) and GVHMR (Shen et al., 2024), which support monocular 3D body and motion recovery. However, their human-specific skeletal topologies, joint definitions, body models, and motion priors do not directly correspond to robot articulation.

The Motion Turing Test (MTT) (Li et al., 2026) represents humanoid motions through human-body pose models for motion analysis and evaluation. However, monocular recovery of robot articulation and root pose remains insufficiently studied. HumanoidPose instead directly recovers these robot-native states from monocular RGB observations.

2.3 Pose and Motion Datasets for Humans and Robots

Human datasets such as Human3.6M (Ionescu et al., 2014), AMASS (Mahmood et al., 2019), and Motion-X (Lin et al., 2023) provide large-scale pose and motion supervision, while robot datasets such as DREAM (Lee et al., 2020) and RoboNet (Dasari et al., 2019) mainly focus on non-humanoid embodiments. Recent humanoid datasets, including DHRP (Heo et al., 2024), HumanoidRobotPose (Amini et al., 2021), PHUMA (Lee et al., 2025), and HHMotion (Li et al., 2026), further advance humanoid robot research from perception, motion, and control perspectives.

As summarized in Table 2, existing datasets remain limited in jointly providing visual observations, 3D pose annotations, and robot-state supervision across diverse humanoid embodiments. **Humanoids** is designed to address this gap.

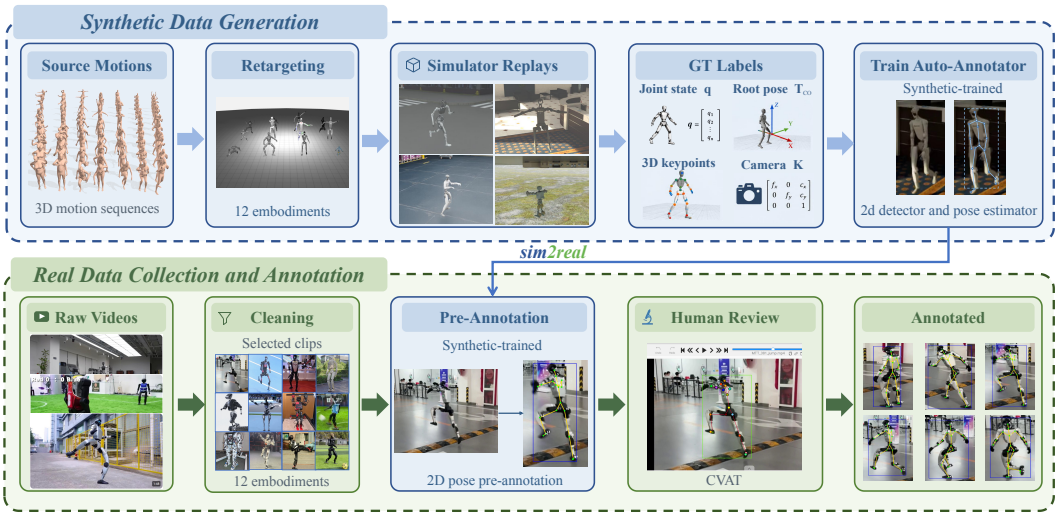


Figure 2: ***Humanoids* Dataset Construction Pipeline**. From left to right, the pipeline illustrates synthetic 3D data generation and real-world 2D annotation, where synthetic-trained 2D pose estimator enables sim-to-real pre-annotation.

Table 2: Comparison with related datasets. The "-" symbol indicates that the corresponding data type or annotation is unavailable.

Dataset	Domain	Data Composition				Real/Sim	#Emb.	#Dur	#Clips
		RGB	2D KP	3D Pose	Robot State				
AMASS(Mahmood et al., 2019)	Human	-	-	✓	-	Real	-	40+h	26.3k
Human3.6M(Ionescu et al., 2014)	Human	✓	✓	✓	-	Real	-	-	-
Motion-X(Lin et al., 2023)	Human	✓	-	✓	-	Real	-	144.2h	81,084
LAFAN1	Human	-	-	✓	-	Real	-	4.6h	-
DREAM(Lee et al., 2020)	Non-Humanoid	✓	✓	✓	✓	Real+Sim	3	-	-
RoboNet(Dasari et al., 2019)	Non-Humanoid	✓	-	-	✓	Real	7	-	15M+
DHRP(Heo et al., 2024)	Humanoid	✓	✓	-	-	Real+Sim	11	-	462
HumanoidRobotPose(Amini et al., 2021)	Humanoid	✓	✓	-	-	Real	10+	-	23
BONES-SEED(Luo et al., 2026)	Human-Humanoid	-	-	✓	✓	Real	-	-	-
PHUMA(Lee et al., 2025)	Human-Humanoid	-	-	✓	✓	Sim	2	72.96h	76.01k
HHMotion(Li et al., 2026)	Human-Humanoid	✓	-	✓	-	Real+Sim	11	1.39h	1,000
Humanoids (Ours)	Humanoid	✓	✓	✓	✓	Real+Sim	**12**	**7.3h**	**7,250**

3 HUMANOIDS: DATASET AND BENCHMARK

3.1 OVERVIEW

Existing datasets provide limited support for humanoid robot 3D pose estimation (Table 2). We therefore introduce **Humanoids**, a dataset and benchmark for monocular 3D humanoid robot pose estimation with RGB observations, robot joint states, root poses, and FK-derived 3D keypoints. It covers 12 humanoid embodiments, 3,706 unique source motions, 6,885 motion clips, and 732K RGB frames across diverse motions and visual conditions. In addition, we provide a real-world evaluation set containing 365 RGB clips, 56,437 frames, and 12 embodiments, with manually verified 2D annotations using a Canonical-18 keypoint representation (Appendix C) for evaluating real-world generalization. We denote the synthetic split **Humanoids-Sim** and the real-world set **Humanoids-Real**.

3.2 DATA CONSTRUCTION

3.2.1 SYNTHETIC DATA

Human motions from the HDM05, CMU, and BMLrub subsets of AMASS (Mahmood et al., 2019) are retargeted to 12 diverse humanoid embodiments (listed in Appendix A) using GMR (Araujo et al., 2025) and replayed in Isaac Sim (NVIDIA, 2025) and MuJoCo (Todorov et al., 2012). Robot

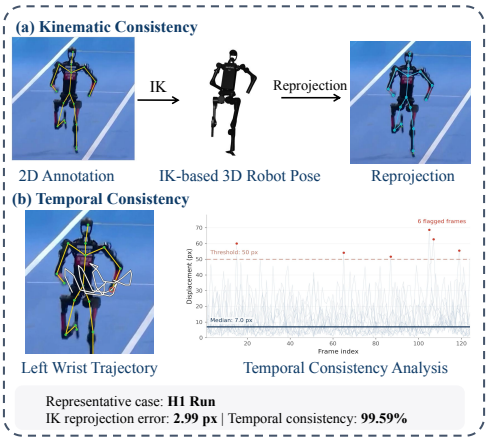


Figure 3: Real-world annotation quality verification

Table 3: Synthetic annotation quality.

Metric	Mean	P95	Max
FK align. (mm)	2.83×10^{-5}	7.16×10^{-5}	2.50×10^{-4}
Reproj. error (px)	3.40×10^{-4}	1.98×10^{-3}	1.48×10^{-2}
Ground pen. (mm)	1.56×10^{-6}	1.41×10^{-5}	8.46×10^{-5}
Joint-limit viol. (deg)	2.01×10^{-11}	1.21×10^{-15}	4.78×10^{-5}

Table 4: Real-world annotation quality.

Metric	Mean	P95	Temp. Consistency
Keypoint velocity (px/frame)	15.12	34.01	97.45%
IK reproj. (px)	4.29	15.10	–

joint states, root poses, and FK-derived Canonical-18 3D keypoints are recorded directly from simulation to provide synchronized robot-state and 3D pose supervision. The detailed definition of the Canonical-18 keypoint representation is provided in Appendix C. We include diverse simulated environments and randomize robot placements, camera viewpoints, distances, and lighting conditions, and render 640×480 RGB frames with synchronized camera parameters. After clip extraction and quality control, the synthetic set contains 6,885 clips and 732K frames.

3.2.2 Real-world data

We collect public and manufacturer-provided humanoid videos covering the same 12 robot embodiments as the synthetic benchmark, with diverse viewpoints, environments, and motions. We remove sequences with severe occlusions or poor visibility, and segment videos into 2–8s clips containing individual continuous motions. For annotation, we leverage an off-the-shelf detector (Redmon et al., 2016) and keypoint model (Xu et al., 2022) trained on the synthetic benchmark to initialize robot bounding boxes and Canonical-18 keypoint predictions on real-world videos. The annotations are then manually verified and corrected in CVAT. To avoid circular bias between annotation and evaluation, the deployed 2D frontend differs from the model used to initialize the annotations. The resulting bounding boxes and Canonical-18 keypoints support image-space evaluation under real-world conditions.

3.3 Dataset Quality

Synthetic data. Because annotations are obtained directly from simulator states, we verify their consistency through FK reconstruction, camera reprojection, ground-contact consistency, and trajectory analysis. Table 3 summarizes the resulting statistics: the FK alignment error is 2.83×10^{-5} mm on average with a maximum of 2.50×10^{-4} mm, reprojection error stays below 1.5×10^{-2} px, ground penetration stays below 8.5×10^{-5} mm, and joint-limit violations are negligible. No hard failures were observed in the final release.

Real-world annotations. Without metric 3D ground truth, we evaluate annotation reliability through temporal and kinematic consistency. The verified annotations exhibit low temporal outlier rates, indicating stable keypoint trajectories across frames. We further fit robot configurations through inverse kinematics and project the resulting 3D poses back to the image plane. The resulting reprojection errors show that the annotations can be largely explained by feasible robot kinematic configurations. Table 4 reports the resulting statistics: keypoint velocities average 15.12 px/frame with a temporal-consistency rate of 97.45%, and the IK reprojection error averages 4.29 px (P95 15.10 px), confirming stable keypoint trajectories and feasible kinematic configurations.

3.4 BENCHMARK PROTOCOL

The Humanoids benchmark provides two evaluation tracks: synthetic 3D pose recovery and real-world image-space evaluation.

For the synthetic benchmark, we provide motion-disjoint train, validation, and test splits, such that clips derived from the same source motion are kept within a single split. Each sample is associated with RGB observations, camera parameters, robot states, and FK-derived 3D keypoints. The default evaluation follows the provided robot kinematic specification and reports articulation and 3D pose metrics.

For the real-world evaluation set, metric 3D ground truth is unavailable. Therefore, we evaluate image-space pose recovery using manually verified Canonical-18 2D keypoint annotations and report 2D pose metrics.

4 HUMANOIDPOSE: ROBOT-NATIVE HUMANOID POSE RECOVERY

Building on the Humanoids benchmark, we present **HumanoidPose**, a monocular framework for estimating robot-native articulation and camera-relative root pose. As shown in Fig. 4, HumanoidPose consists of three modules: a frozen *visual frontend* (Sec. 4.2) for extracting bounding boxes, Canonical-18 keypoints (Appendix C), and spatio-temporal visual tokens; *robot pose initialization* (Sec. 4.3) for estimating articulation and root pose with separate branches; and *kinematic grounding and bounded refinement* (Sec. 4.4) for refining the initialized state using embodiment-specific kinematics and image-space cues.

4.1 PROBLEM FORMULATION

We study *Humanoid Robot Pose Estimation (HRPE)*, which aims to recover the joint configuration and camera-relative root pose of a humanoid robot from monocular RGB observations.

Each humanoid embodiment r is described by its kinematic model $\mathcal{M}_r = (\mathrm{FK}_r, q_{\min}, q_{\max})$, where $\mathrm{FK}_r : \mathbb{R}^{D_r} \to \mathbb{R}^{J\times 3}$ maps a joint configuration to the Canonical-18 keypoints (Appendix C) in the robot root frame, where D_r is the number of robot joints and J is the number of keypoints. Given a monocular RGB sequence window $I_{1:T}$, HRPE estimates

$$(q, R, t) = \mathcal{F}_{\theta_r}(I_{1:T}; \mathcal{M}_r), \tag{1}$$

where $q \in [q_{\min}, q_{\max}]$ denotes the joint configuration at the target time step and $(R, t) \in \mathrm{SO}(3)\times\mathbb{R}^3$ denotes the camera-relative root pose. The corresponding 3D robot structure is obtained deterministically through $\mathrm{FK}_r(q)$, thereby respecting the embodiment-specific kinematic structure and link geometry. We use the same architecture for all embodiments, but train a separate set of model parameters for each embodiment.

4.2 VISUAL FRONTEND

Given an RGB sequence, a detector (Redmon et al., 2016) and a 2D pose estimator (Khirodkar et al., 2026), pre-adapted on synthetic humanoid data, provide the target bounding box b and Canonical-18 keypoints $\hat{x} \in \mathbb{R}^{J\times 2}$. A frozen V-JEPA encoder (Bardes et al., 2024) extracts spatio-temporal visual tokens z. All frontend modules remain fixed during HRPE training. Together, z, $\hat{x}$, and b provide complementary visual, structural, and geometric cues for subsequent robot pose initialization.

4.3 ROBOT POSE INITIALIZATION

Joint articulation is primarily determined by local limb configurations, whereas camera-relative root pose depends more strongly on global placement, scale, and orientation in the image. We therefore initialize them through two separate branches:

$$q_0 = f_q(z, \hat{x}; r), \qquad R_0 = f_R(z, \hat{x}, b; r), \qquad t_0 = f_t(z, \hat{x}, b; r). \tag{2}$$

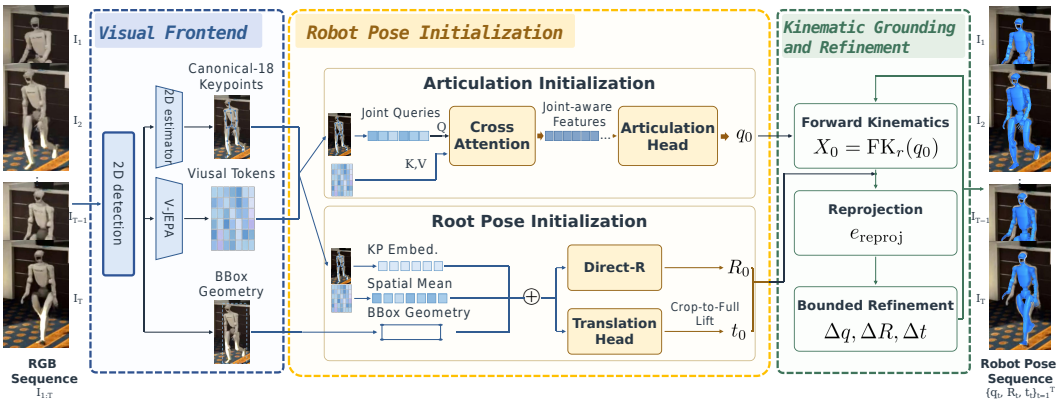


Figure 4: Overview of HumanoidPose. The visual frontend extracts Canonical-18 keypoints, bounding-box geometry, and spatio-temporal visual tokens. Keypoint-conditioned joint queries initialize robot articulation q_0, while a separate root branch estimates the camera-relative root pose (R_0, t_0). The initial state is grounded through embodiment-specific forward kinematics and image-space reprojection, followed by bounded residual refinement to obtain the final robot pose.

Articulation branch. The articulation branch predicts robot joint configurations through keypoint-conditioned queries. Queries constructed from the Canonical-18 keypoints attend to visual tokens through cross-attention (Vaswani et al., 2017), yielding $h = \mathrm{CrossAttn}\big(Q(\hat{x}), z\big)$, where each query corresponds to a robot joint and retrieves joint-specific evidence from the visual tokens. The resulting representation is decoded into q_0.

Root branch. The root branch estimates camera-relative orientation and translation by combining visual features with the Canonical-18 keypoints and bounding-box geometry. The resulting initial state (q_0, R_0, t_0) is subsequently grounded through the robot kinematic model.

4.4 Kinematic Grounding and Bounded Refinement

The initialized articulation is grounded in the embodiment-specific robot geometry through forward kinematics, yielding $X_0 = \mathrm{FK}_r(q_0)$, where X_0 denotes the metric 3D robot structure in the root frame.

The FK-derived structure, projected keypoints, and image-space residuals form geometric features ϕ for refinement. Rather than predicting a new pose from scratch, the refinement module predicts bounded residual corrections to the initial state (q_0, R_0, t_0) to obtain the final pose (q, R, t). The residual parameterization and feature construction are detailed in Appendix D. The bounded refinement restricts corrections around the initial estimate, while joint-limit constraints maintain embodiment-specific feasibility.

4.5 Training Objective

Given the ground-truth state (q^*, R^*, t^*), HumanoidPose is supervised with complementary objectives in joint, Cartesian, camera, and image spaces:

$$\mathcal{L} = \lambda_q \mathcal{L}_q + \lambda_{\mathrm{FK}} \mathcal{L}_{\mathrm{FK}} + \lambda_R \mathcal{L}_R + \lambda_t \mathcal{L}_t + \lambda_{\mathrm{cam}} \mathcal{L}_{\mathrm{cam}} + \lambda_{\mathrm{repr}} \mathcal{L}_{\mathrm{repr}} + \lambda_\Delta \mathcal{L}_\Delta. \tag{3}$$

Here, $\mathcal{L}_q$ and $\mathcal{L}_t$ are Smooth-ℓ_1 losses, $\mathcal{L}_R$ is a geodesic rotation loss, and $\mathcal{L}_{\mathrm{cam}}$ and $\mathcal{L}_{\mathrm{repr}}$ enforce camera-space and image-space consistency. The FK loss $\mathcal{L}_{\mathrm{FK}} = \| \mathrm{FK}_r(q) - \mathrm{FK}_r(q^*) \|_1$ directly supervises the FK-derived 3D structure, while $\mathcal{L}_\Delta$ regularizes the residual corrections. Detailed loss definitions are provided in Appendix D.3.

Table 5: **Comparison with existing pose estimation methods on Humanoids-Sim.** ⋆ denotes robot-specific adaptation while preserving the original inference pipeline.

Category	Method	MPJAE (°)↓	MPJPE (mm)↓	PA-MPJPE (mm)↓	G-MPJPE (mm)↓	PCK@0.3 (%)↑
HPE	HMR2 (Goel et al., 2023) + GMR	29.503	150.772	115.508	2909.486	65.04
	GVHMR (Shen et al., 2024) + GMR	28.471	140.666	113.203	2598.000	70.78
	WHAM (Shin et al., 2024) + GMR	32.754	189.039	163.659	2980.960	62.24
	HMR2⋆ (Goel et al., 2023)	13.634	75.574	67.632	320.149	84.74
	GVHMR⋆ (Shen et al., 2024)	13.522	75.606	68.011	331.569	84.81
	WHAM⋆ (Shin et al., 2024)	16.902	100.052	91.797	803.063	77.88
RPE	RoboPEPP⋆ (Goswami et al., 2025)	15.771	89.200	80.230	2501.417	80.81
	HoRoPose⋆ (Ban et al., 2024)	22.949	145.910	135.039	1346.556	69.72
	RoboPose⋆ (Labbé et al., 2021)	16.139	91.173	81.125	540.262	80.45
HRPE	**HumanoidPose (Ours)**	**12.430**	**59.954**	**52.171**	**183.218**	**89.75**

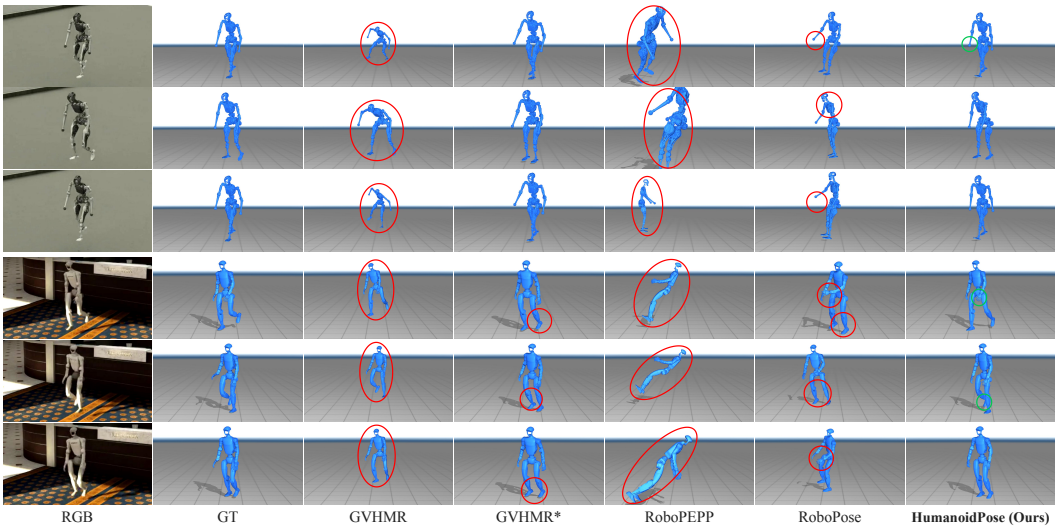


Figure 5: Visual comparison of HumanoidPose against baseline methods on the synthetic Humanoids benchmark. Results are shown for G1 (bottom) and PNDAdamLite (top). Here ⋆ denotes robot-specific adaptation.

5 Experiments

5.1 Experimental Setup

Following Sec. 3.4, we train and evaluate on the synthetic split of **Humanoids** and assess sim-to-real transfer on **Humanoids-Real** and external **DHRP-H1** (Heo et al., 2024). Both real-world evaluations are *zero-shot*, with no real-image adaptation or fine-tuning. Each embodiment uses the same architecture with independently trained parameters; the benchmark measures per-embodiment accuracy. We report macro averages over 12 embodiments and provide per-embodiment results in the appendix. For synthetic evaluation, we report **MPJAE**, **MPJPE**, **PA-MPJPE**, **G-MPJPE**, and **PCK@0.3**. For real-world evaluation, we report image-space metrics, including **2D MPJPE**, **NME**, and **PCK@0.1**. Detailed metric definitions are provided in Appendix G.

5.2 Comparison with Existing Methods

We compare representative HPE and RPE methods under the HRPE protocol. HPE baselines use either (1) direct application with human-to-robot retargeting (e.g., GMR (Araujo et al., 2025)) without additional training, or (2) robot-specific output representations trained on Humanoids. RPE baselines retain their original formulations, with outputs adapted to each humanoid embodiment and training on its corresponding synthetic data. Implementation details are in Appendix H.

As shown in Table 5, HumanoidPose consistently achieves strong performance across articulation estimation, root-relative 3D pose recovery, and global camera-space evaluation. The comparison

Table 6: Real-world image-space evaluation on Humanoids-Real and DHRP-H1 under the zero-shot setting. ⋆ denotes robot-specific adaptation. ↓ lower is better; ↑ higher is better.

Method	Humanoids-Real			DHRP-H1		
	2D MPJPE (px)↓	NME↓	PCK@0.1 (%)↑	2D MPJPE (px)↓	NME↓	PCK@0.1 (%)↑
HPE						
GVHMR+GMR	78.82	0.1371	53.27	97.62	0.1470	62.06
HMR2+GMR	247.77	0.4067	5.77	253.68	0.3285	10.61
WHAM+GMR	197.97	0.3098	15.81	491.47	0.6015	0.99
GVHMR⋆	149.27	0.2432	22.97	98.81	0.1481	60.69
HMR2⋆	162.98	0.2607	15.94	146.48	0.1941	15.73
WHAM⋆	243.53	0.4048	7.49	406.34	0.5003	2.06
RPE						
RoboPEPP⋆	139.18	0.2157	23.44	282.25	0.3684	8.40
HoRoPose⋆	128.32	0.2023	28.72	208.18	0.2830	17.48
RoboPose⋆	152.10	0.2356	20.64	212.69	0.2901	14.73
HRPE						
HumanoidPose	**71.49**	**0.1180**	**61.97**	**87.80**	**0.1363**	**74.20**

Table 7: Ablation study of HumanoidPose components on Humanoids-Sim. ↓ lower is better; ↑ higher is better.

Variant	MPJAE (°)↓	MPJPE (mm)↓	PA-MPJPE (mm)↓	G-MPJPE (mm)↓	PCK@0.3 (%)↑
w/o Robot Artic.	14.668	72.237	61.115	190.705	86.87
w/o Decoup. Root	12.537	60.429	52.664	213.423	89.51
w/o Kinematic Grounding & Refinement	12.851	64.705	56.064	194.082	88.26
Full HumanoidPose	**12.430**	**59.954**	**52.171**	**183.218**	**89.75**

Table 8: Frontend robustness analysis. ↓ lower is better; ↑ higher is better.

Frontend	MPJAE (°)↓	MPJPE (mm)↓	PA-MPJPE (mm)↓	G-MPJPE (mm)↓	PCK@0.3 (%)↑
V-JEPA+Sapiens2	12.430	59.954	52.171	183.218	89.75
DINOv2+Sapiens2	11.497	55.972	49.199	199.485	90.59
V-JEPA+ViTPose-H	11.998	57.037	49.448	180.049	90.42
DINOv2+ViTPose-H	11.444	55.245	48.468	197.274	91.05

highlights two observations. First, direct transfer of HPE models through human-to-robot retargeting suffers from substantial performance degradation, whereas robot-specific adaptation improves their accuracy, indicating that human-centric visual priors require alignment with robot-specific structures. Second, existing RPE methods provide robot-aware representations but are mainly designed for constrained robot systems. HumanoidPose addresses both limitations by directly modeling robot articulation with explicit kinematic grounding.

Figure 5 further illustrates these differences. GVHMR suffers from human-body prior mismatch, particularly for smaller humanoid platforms such as Unitree G1, while RoboPEPP's geometry-based root estimation can degrade on humanoid robots with diverse embodiments. HumanoidPose better preserves robot articulation and camera-relative root pose.

5.3 Real-World Evaluation

Table 6 compares HPE and RPE baselines on **Humanoids-Real** and **DHRP-H1**. As metric 3D robot states are unavailable, we evaluate projected predictions against manual 2D annotations.

HumanoidPose achieves the best accuracy on both benchmarks; among baselines, zero-shot GVHMR+GMR is the strongest in image space.

5.4 Ablation Studies

We conduct ablation studies on Humanoids-Sim (Table 7). Removing robot-native articulation causes the largest degradation, highlighting the importance of directly predicting robot joint states. Removing kinematic grounding and bounded refinement degrades both root-relative and global accuracy. Merging the articulation and root branches mainly hurts global camera-space alignment while leaving articulation nearly unchanged, supporting the decoupled initialization.

We further evaluate the robustness to different perception frontends, including the V-JEPA (Bardes et al., 2024) and DINOv2 (Oquab et al., 2023) encoders and the Sapiens2 (Khirodkar et al., 2026) and ViTPose-H (Xu et al., 2022) 2D pose frontends, in Table 8. Replacing either the visual encoder or the 2D pose frontend maintains comparable performance, indicating that HumanoidPose does not rely on a specific frontend configuration.

6 Conclusion

We introduced *Humanoid Robot Pose Estimation (HRPE)*, a monocular perception task that recovers robot-native articulation and camera-relative floating-base pose for humanoid robots, and a 12-embodiment benchmark with quality-controlled annotations and evaluation protocols. We propose HumanoidPose, combining robot-native articulation prediction, decoupled root initialization, and explicit kinematic grounding. Experiments validate this robot-native design. This work can support future research in humanoid robot pose estimation.

AI USE STATEMENT

Generative AI tools were used to assist with language translation and with implementing and executing code. They were not used to generate research ideas, design experiments, interpret experimental results, or select literature references. All AI-assisted outputs were reviewed and verified by the authors, who take full responsibility for the scientific content, accuracy, and integrity of the work.

ETHICS STATEMENT

This work studies monocular perception of humanoid robots and does not involve the collection or processing of personally identifiable information. The proposed synthetic benchmark is generated in simulation using robot models and motion data, while real-world evaluations focus on humanoid robot observations without human subjects or identity-related annotations.

We acknowledge that visual perception capabilities for humanoid robots may serve as a foundation for future human–robot interaction and autonomous robotic systems. Such technologies should be developed and deployed with careful consideration of safety, reliability, and appropriate human oversight. This work aims to facilitate research on robot state understanding through open evaluation protocols and benchmarks.

REFERENCES

Arthur Allshire, Hongsuk Choi, Junyi Zhang, David McAllister, Anthony Zhang, Chung Min Kim, Trevor Darrell, Pieter Abbeel, Jitendra Malik, and Angjoo Kanazawa. Visual imitation enables contextual humanoid control. *arXiv preprint arXiv:2505.03729*, 2025.

Arash Amini, Hafez Farazi, and Sven Behnke. Real-time pose estimation from images for multiple humanoid robots. In *Robot World Cup*, pp. 91–102. Springer, 2021.

Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *arXiv preprint arXiv:2510.02252*, 2025.

Shikun Ban, Juling Fan, Xiaoxuan Ma, Wentao Zhu, Yu Qiao, and Yizhou Wang. Real-time holistic robot pose estimation with unknown states. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2024.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter V. Gehler, Javier Romero, and Michael J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016.

Youngdae Cho, Wooram Son, Jaewan Bak, Yisoo Lee, Hwasup Lim, and Youngwoon Cha. Full-body pose estimation of humanoid robots using head-worn cameras for digital human-augmented robotic telepresence. *Mathematics*, 12(19):3039, 2024.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. In *Conference on Robot Learning (CoRL)*, 2024.

Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14737–14748. IEEE, 2023.

Raktim Gautam Goswami, Prashanth Krishnamurthy, Yann LeCun, and Farshad Khorrami. Robopepp: Vision-based robot pose and joint angle estimation through embedding predictive pre-training. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6930–6939. IEEE, 2025.

Seokhyeon Heo, Youngdae Cho, Jeongwoo Park, Seokhyun Cho, Ziya Tsoy, Hwasup Lim, and Youngwoon Cha. Diverse humanoid robot pose estimation from images using only sparse datasets. *Applied Sciences*, 14 (19):9042, 2024.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(7):1325–1339, 2014. doi: 10.1109/TPAMI.2013.248.

Rawal Khirodkar, He Wen, Julieta Martinez, Yuan Dong, Zhaoen Su, and Shunsuke Saito. Sapiens2. *arXiv preprint arXiv:2604.21681*, 2026.

Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5252–5262, 2020.

Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Single-view robot pose and joint angle estimation via render & compare. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1654–1663. IEEE, 2021.

Kyungmin Lee, Sibeen Kim, Minho Park, Hyunseung Kim, Dongyoon Hwang, Hojoon Lee, and Jaegul Choo. Phuma: Physically-grounded humanoid locomotion dataset. *arXiv preprint arXiv:2510.26236*, 2025.

Timothy E Lee, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Oliver Kroemer, Dieter Fox, and Stan Birchfield. Camera-to-robot pose estimation from a single image. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 9426–9432. IEEE, 2020.

Mingzhe Li, Mengyin Liu, Zekai Wu, Xincheng Lin, Junsheng Zhang, Ming Yan, Zengye Xie, Changwang Zhang, Chenglu Wen, Lan Xu, et al. Towards motion turing test: Evaluating human-likeness in humanoid robots. *arXiv preprint arXiv:2603.06181*, 2026.

Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241*, 2025.

Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-x: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 36:25268–25280, 2023.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, October 2015. ISSN 0730-0301.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Jiageng Mao, Siheng Zhao, Siqi Song, Tianheng Shi, Junjie Ye, Mingtong Zhang, Haoran Geng, Jitendra Malik, Vitor Guizilini, and Yue Wang. Learning from massive human videos for universal humanoid pose control. *arXiv preprint arXiv:2412.14172*, 2024.

NVIDIA. Nvidia isaac sim. `https://developer.nvidia.com/isaac-sim`, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Science Robotics*, 9(89):eadi9579, 2024.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Zehong Shen, Huaijin Pi, Yan Xia, Zhi Cen, Sida Peng, Zechen Hu, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. World-grounded human motion recovery via gravity-view coordinates. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J Black. Wham: Reconstructing world-grounded humans with accurate 3d motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2070–2080, 2024.

Yang Tian, Jiyao Zhang, Guowei Huang, Bin Wang, Ping Wang, Jiangmiao Pang, and Hao Dong. Robokeygen: robot pose and joint angles estimation via diffusion-based 3d keypoint generation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5375–5381. IEEE, 2024.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30: 5998–6008, 2017.

Weiji Xie, Jinrui Han, Jiakun Zheng, Huanyu Li, Xinzhe Liu, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. Kungfubot: Physics-based humanoid whole-body control for learning highly-dynamic skills. *Advances in Neural Information Processing Systems*, 38:62406–62433, 2025.

Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, 35:38571–38584, 2022.

James Young, Jonathan Roberts, et al. Robots watching robots: Human pose estimation applied to humanoid robots. In *Proceedings-Australasian Conference on Robotics and Automation (ACRA 2025)*. Australian Robotics and Automation Association (ARAA), 2025.

APPENDICES

A PER-ROBOT BREAKDOWN OF THE MAIN EXPERIMENTS

Tables 9–13 break down every Humanoids-Sim method of the main comparison by robot, and Tables 14–16 do the same on the internal Humanoids-Real 2D set. In each table the *Mean* column is the robot-balanced average, i.e. the value reported in the main comparison tables; the per-robot entries are computed under the shared evaluation protocol of Section H. The external DHRP-H1 benchmark is a single image set and is therefore not broken down by robot.

Table 9: Per-robot MPJAE (°) on Humanoids-Sim. Robots: H1=UnitreeH1, G1=G1, BT1=BoosterT1, BK1=BoosterK1, BHL=BerkeleyHumanoidLite, EPM=EngineAI_PM01, FGR3=FourierGR3, FN1=FourierN1, KS45=KuavoS45, PND=PNDAdamLite, STD=StanfordToddy, TK=TienKung. The *Mean* column is the robot-balanced average reported in the main table.

Method	H1	G1	BT1	BK1	BHL	EPM	FGR3	FN1	KS45	PND	STD	TK	Mean
HMR2 + GMR	32.1	38.8	12.9	41.9	31.7	48.8	27.7	30.5	24.1	22.3	16.1	27.0	**29.5**
GVHMR + GMR	48.8	24.0	10.2	39.4	21.0	41.2	33.4	31.1	34.9	21.8	17.1	18.7	**28.5**
WHAM + GMR	44.4	42.6	16.7	52.2	31.0	48.7	23.0	21.1	28.3	24.7	16.5	43.6	**32.8**
HMR2*	16.6	13.4	9.7	14.0	14.7	17.0	11.2	13.2	14.9	12.3	11.8	14.8	**13.6**
GVHMR*	16.4	13.6	8.9	14.1	14.8	16.8	11.4	13.5	14.6	12.1	12.1	14.0	**13.5**
WHAM*	16.0	17.7	13.6	17.1	15.5	23.3	15.4	17.2	20.3	13.4	13.3	20.0	**16.9**
RoboPEPP*	16.4	16.2	11.1	16.3	16.1	22.0	13.2	16.0	18.4	12.9	12.8	17.9	**15.8**
HoRoPose*	19.3	32.6	18.3	20.5	19.8	26.6	18.4	18.8	22.4	15.6	26.3	36.8	**22.9**
RoboPose*	16.3	18.8	12.0	17.5	15.6	22.1	13.8	16.7	17.4	13.7	12.5	17.3	**16.1**
HumanoidPose (Ours)	14.2	13.0	7.7	13.8	14.5	16.3	11.1	13.5	11.2	10.4	10.6	12.8	**12.4**

Table 10: Per-robot MPJPE (mm) on Humanoids-Sim. Robots: H1=UnitreeH1, G1=G1, BT1=BoosterT1, BK1=BoosterK1, BHL=BerkeleyHumanoidLite, EPM=EngineAI_PM01, FGR3=FourierGR3, FN1=FourierN1, KS45=KuavoS45, PND=PNDAdamLite, STD=StanfordToddy, TK=TienKung. The *Mean* column is the robot-balanced average reported in the main table.

Method	H1	G1	BT1	BK1	BHL	EPM	FGR3	FN1	KS45	PND	STD	TK	Mean
HMR2 + GMR	137.4	187.2	47.8	80.6	133.9	273.6	245.6	146.7	139.5	210.1	45.6	161.3	**150.8**
GVHMR + GMR	207.0	144.1	46.4	86.9	68.7	160.4	297.2	141.7	233.7	159.1	48.1	94.8	**140.7**
WHAM + GMR	255.2	280.4	59.5	104.4	142.6	284.2	169.5	117.3	272.1	210.5	42.5	330.2	**189.0**
HMR2*	77.9	83.5	38.2	38.4	54.6	128.3	103.6	71.9	103.7	94.0	34.4	78.4	**75.6**
GVHMR*	77.9	88.6	34.3	38.4	55.4	129.8	107.6	72.2	103.6	91.0	35.5	73.0	**75.6**
WHAM*	75.2	121.1	50.7	47.2	61.6	158.6	159.5	97.0	161.3	106.1	42.2	120.1	**100.1**
RoboPEPP*	77.1	111.9	40.2	42.4	59.8	145.2	128.6	88.1	142.4	98.9	39.2	96.5	**89.2**
HoRoPose*	111.1	226.3	75.8	64.5	69.2	189.4	192.9	101.0	174.5	131.2	57.6	357.5	**145.9**
RoboPose*	74.7	133.4	43.4	50.0	57.4	147.6	138.0	94.7	118.8	101.9	38.8	95.3	**91.2**
HumanoidPose (Ours)	55.9	73.6	24.9	32.5	50.8	100.6	91.4	66.4	69.3	70.3	23.6	60.2	**60.0**

Table 11: Per-robot PA-MPJPE (mm) on Humanoids-Sim. Robots: H1=UnitreeH1, G1=G1, BT1=BoosterT1, BK1=BoosterK1, BHL=BerkeleyHumanoidLite, EPM=EngineAI_PM01, FGR3=FourierGR3, FN1=FourierN1, KS45=KuavoS45, PND=PNDAdamLite, STD=StanfordToddy, TK=TienKung. The *Mean* column is the robot-balanced average reported in the main table.

Method	H1	G1	BT1	BK1	BHL	EPM	FGR3	FN1	KS45	PND	STD	TK	Mean
HMR2 + GMR	128.7	120.6	47.1	80.6	134.6	151.5	168.7	103.6	145.4	129.5	37.0	139.0	**115.5**
GVHMR + GMR	209.4	102.5	37.6	80.7	63.2	96.7	207.8	87.4	228.4	130.2	37.7	76.9	**113.2**
WHAM + GMR	244.1	193.5	61.8	96.2	145.8	208.0	147.0	89.8	259.0	181.9	40.2	296.8	**163.7**
HMR2*	78.7	67.5	36.6	37.4	53.3	108.2	82.6	55.2	109.9	77.9	31.0	73.3	**67.6**
GVHMR*	79.0	72.8	32.4	37.2	53.8	111.2	86.8	57.5	109.9	76.5	32.6	66.4	**68.0**
WHAM*	75.2	101.1	48.2	46.3	61.5	132.7	129.1	86.2	175.5	88.5	37.3	119.9	**91.8**
RoboPEPP*	78.2	93.9	39.2	42.0	58.7	109.6	109.3	75.6	146.7	83.7	34.7	91.2	**80.2**
HoRoPose*	118.0	180.9	75.8	62.9	67.2	149.4	162.5	84.0	177.7	112.8	53.1	376.3	**135.0**
RoboPose*	76.1	106.8	42.2	52.1	56.3	115.7	111.8	76.5	126.4	87.1	34.0	88.6	**81.1**
HumanoidPose (Ours)	54.7	57.5	22.6	30.5	48.5	84.0	69.4	47.3	76.3	59.3	20.2	55.8	**52.2**

Table 12: Per-robot G-MPJPE (mm) on Humanoids-Sim. Robots: H1=UnitreeH1, G1=G1, BT1=BoosterT1, BK1=BoosterK1, BHL=BerkeleyHumanoidLite, EPM=EngineAI_PM01, FGR3=FourierGR3, FN1=FourierN1, KS45=KuavoS45, PND=PNDAdamLite, STD=StanfordToddy, TK=TienKung. The *Mean* column is the robot-balanced average reported in the main table.

Method	H1	G1	BT1	BK1	BHL	EPM	FGR3	FN1	KS45	PND	STD	TK	Mean
HMR2 + GMR	1937	2074	2830	4871	5003	1316	700	2299	919	1671	10940	353	**2909**
GVHMR + GMR	1068	2053	2277	4171	4088	1420	676	2243	971	1796	10158	255	**2598**
WHAM + GMR	2427	2086	2990	4929	4534	1388	675	2344	933	1373	11386	707	**2981**
HMR2*	572	235	214	227	398	273	311	233	387	476	281	235	**320**
GVHMR*	515	292	225	317	399	283	325	263	373	416	328	241	**332**
WHAM*	609	699	813	657	778	1136	878	741	895	1053	555	824	**803**
RoboPEPP*	1581	1762	602	1019	525	2989	1664	844	12450	1273	1340	3967	**2501**
HoRoPose*	873	5982	398	483	404	723	1703	513	1270	1901	803	1106	**1347**
RoboPose*	727	641	439	449	421	506	609	442	655	723	297	574	**540**
HumanoidPose (Ours)	157	176	144	216	174	215	211	173	198	223	146	165	**183**

Table 13: Per-robot PCK@0.3 (%) on Humanoids-Sim. Robots: H1=UnitreeH1, G1=G1, BT1=BoosterT1, BK1=BoosterK1, BHL=BerkeleyHumanoidLite, EPM=EngineAI_PM01, FGR3=FourierGR3, FN1=FourierN1, KS45=KuavoS45, PND=PNDAdamLite, STD=StanfordToddy, TK=TienKung. The *Mean* column is the robot-balanced average reported in the main table.

Method	H1	G1	BT1	BK1	BHL	EPM	FGR3	FN1	KS45	PND	STD	TK	Mean
HMR2 + GMR	69.3	46.7	90.8	85.5	64.3	36.2	50.4	63.2	68.0	45.2	95.2	65.8	**65.0**
GVHMR + GMR	62.1	69.5	89.9	83.9	82.8	62.4	46.6	64.7	59.2	57.6	94.0	76.6	**70.8**
WHAM + GMR	53.2	38.3	86.7	77.6	63.1	43.3	63.9	72.3	53.2	48.9	95.1	51.3	**62.2**
HMR2*	82.5	84.3	93.7	94.8	89.8	72.7	78.6	86.4	76.0	78.3	97.4	82.5	**84.7**
GVHMR*	82.5	82.3	95.2	95.1	89.3	72.4	77.9	85.8	75.8	79.7	97.1	84.7	**84.8**
WHAM*	83.1	71.4	90.1	91.9	87.5	64.2	63.8	77.9	64.5	73.5	94.8	71.9	**77.9**
RoboPEPP*	82.6	74.8	93.2	93.4	87.7	66.2	72.9	81.3	68.2	75.9	94.8	78.8	**80.8**
HoRoPose*	77.0	46.0	82.3	84.0	84.4	58.2	55.3	77.2	60.6	63.1	92.9	55.6	**69.7**
RoboPose*	83.1	70.6	92.4	91.7	89.2	68.5	69.5	78.4	72.3	75.2	96.0	78.5	**80.5**
HumanoidPose (Ours)	90.2	87.4	97.5	96.6	92.1	81.7	82.4	87.9	84.9	87.9	98.8	89.6	**89.8**

Table 14: Per-robot 2D MPJPE (px) on Humanoids-Real. Robots: H1=UnitreeH1, G1=G1, BT1=BoosterT1, BK1=BoosterK1, BHL=BerkeleyHumanoidLite, EPM=EngineAI_PM01, FGR3=FourierGR3, FN1=FourierN1, KS45=KuavoS45, PND=PNDAdamLite, STD=StanfordToddy, TK=TienKung. DHRP-H1 is a single external benchmark and is not broken down by robot; see the main table. The *Mean* column reproduces the main-table value.

Method	H1	G1	BT1	BK1	BHL	EPM	FGR3	FN1	KS45	PND	STD	TK	Mean
GVHMR + GMR	114.8	57.7	56.6	81.3	133.7	40.6	35.3	55.9	152.0	87.2	82.9	47.8	**78.8**
HMR2 + GMR	375.8	284.1	275.2	198.1	233.3	239.9	312.4	148.5	198.2	229.5	330.7	147.8	**247.8**
WHAM + GMR	320.8	188.7	61.9	203.3	196.9	177.8	355.1	133.5	177.8	300.9	92.3	166.7	**198.0**
GVHMR*	92.2	191.2	175.7	219.8	192.3	131.8	168.1	132.5	50.8	117.8	243.9	75.2	**149.3**
HMR2*	140.3	162.5	165.3	213.4	196.3	127.9	112.1	119.4	39.1	279.8	338.0	61.7	**163.0**
WHAM*	243.1	174.7	285.2	366.8	190.3	145.7	469.2	181.9	87.0	182.0	502.2	94.2	**243.5**
RoboPEPP*	186.8	106.3	76.4	119.4	144.2	356.3	204.8	85.9	59.8	142.2	81.7	106.3	**139.2**
HoRoPose*	270.6	124.0	93.2	63.6	128.7	97.5	154.4	121.2	75.8	192.1	90.4	128.5	**128.3**
RoboPose*	138.5	293.4	91.7	191.7	228.3	137.7	211.4	113.7	69.0	146.9	105.1	97.8	**152.1**
HumanoidPose (Ours)	77.7	65.7	54.5	88.1	131.4	53.6	76.8	39.3	45.4	70.8	92.1	62.5	**71.5**

Table 15: Per-robot NME on Humanoids-Real. Robots: H1=UnitreeH1, G1=G1, BT1=BoosterT1, BK1=BoosterK1, BHL=BerkeleyHumanoidLite, EPM=EngineAI_PM01, FGR3=FourierGR3, FN1=FourierN1, KS45=KuavoS45, PND=PNDAdamLite, STD=StanfordToddy, TK=TienKung. DHRP-H1 is a single external benchmark and is not broken down by robot; see the main table. The *Mean* column reproduces the main-table value.

Method	H1	G1	BT1	BK1	BHL	EPM	FGR3	FN1	KS45	PND	STD	TK	Mean
GVHMR + GMR	0.1397	0.0779	0.1244	0.1067	0.1636	0.0624	0.0484	0.1031	0.3933	0.1100	0.1939	0.1222	**0.1371**
HMR2 + GMR	0.4595	0.3817	0.5025	0.2630	0.2941	0.3703	0.4279	0.2881	0.5128	0.2831	0.7731	0.3245	**0.4067**
WHAM + GMR	0.3851	0.2556	0.1309	0.2734	0.2404	0.2721	0.4906	0.2538	0.4600	0.3657	0.2154	0.3751	**0.3098**
GVHMR*	0.1343	0.2613	0.3351	0.2905	0.2187	0.1967	0.2256	0.2362	0.1315	0.1411	0.5697	0.1772	**0.2432**
HMR2*	0.1668	0.2128	0.2944	0.2837	0.2430	0.1943	0.1510	0.2184	0.1013	0.3233	0.7903	0.1487	**0.2607**
WHAM*	0.3499	0.2354	0.5169	0.4876	0.2339	0.2061	0.6261	0.3644	0.2251	0.2117	1.1726	0.2274	**0.4048**
RoboPEPP*	0.2219	0.1428	0.1545	0.1560	0.1737	0.5376	0.2749	0.1609	0.1548	0.1766	0.1907	0.2441	**0.2157**
HoRoPose*	0.3366	0.1690	0.1982	0.0843	0.1485	0.1498	0.2086	0.2184	0.1962	0.2335	0.2105	0.2746	**0.2023**
RoboPose*	0.1673	0.4010	0.1756	0.2514	0.2811	0.2106	0.2908	0.2111	0.1786	0.1827	0.2454	0.2314	**0.2356**
HumanoidPose (Ours)	0.0919	0.0888	0.1077	0.1187	0.1622	0.0847	0.1023	0.0799	0.1175	0.0892	0.2153	0.1574	**0.1180**

Table 16: Per-robot PCK@0.1 (%) on Humanoids-Real. Robots: H1=UnitreeH1, G1=G1, BT1=BoosterT1, BK1=BoosterK1, BHL=BerkeleyHumanoidLite, EPM=EngineAI_PM01, FGR3=FourierGR3, FN1=FourierN1, KS45=KuavoS45, PND=PNDAdamLite, STD=StanfordToddy, TK=TienKung. DHRP-H1 is a single external benchmark and is not broken down by robot; see the main table. The *Mean* column reproduces the main-table value.

Method	H1	G1	BT1	BK1	BHL	EPM	FGR3	FN1	KS45	PND	STD	TK	Mean
GVHMR + GMR	53.05	71.64	59.55	48.36	31.04	84.12	96.22	50.27	5.24	51.66	22.93	65.19	**53.27**
HMR2 + GMR	4.36	5.16	0.61	11.27	9.52	5.84	2.52	11.39	0.00	11.22	0.00	7.32	**5.77**
WHAM + GMR	2.32	21.39	54.62	4.08	15.43	8.03	0.00	36.52	0.00	21.32	11.95	14.05	**15.81**
GVHMR*	49.73	11.10	3.92	0.00	19.02	21.24	24.34	10.97	49.45	47.59	0.00	38.32	**22.97**
HMR2*	20.48	8.64	0.52	0.04	2.59	6.44	35.36	7.79	53.07	11.00	0.00	45.38	**15.94**
WHAM*	1.41	12.25	0.25	0.00	12.72	16.81	2.98	5.43	3.92	18.62	0.00	15.45	**7.49**
RoboPEPP*	21.07	39.64	34.52	33.10	20.74	0.43	10.02	28.73	30.38	28.26	18.75	15.61	**23.44**
HoRoPose*	2.71	29.60	34.74	74.93	29.86	47.80	27.44	10.91	37.71	17.21	17.65	14.12	**28.72**
RoboPose*	34.13	14.66	26.69	14.55	9.39	27.66	9.63	17.58	27.41	30.65	15.81	19.52	**20.64**
HumanoidPose (Ours)	68.50	71.89	63.80	57.45	57.47	74.32	65.00	73.57	60.28	73.16	20.91	57.26	**61.97**

B HUMANOIDS DATASET DETAILS

This appendix provides the full statistics, simulation and rendering configuration, and quality-control procedure behind the **Humanoids** benchmark summarized in the main paper. All counts are reproduced from the released manifest and audit reports; the synthetic component is the `hrpev5_source_aware_hybrid_v1` release.

B.1 DATASET STATISTICS

The synthetic component is constructed from AMASS human motions and replayed in two simulators, yielding 12 embodiments, 3,706 unique source motions, 6,885 clips, and 732,181 RGB frames. Table 17 summarizes the release totals, Table 18 gives the per-robot coverage, and Table 19 reports the motion-held-out split. All clips are rendered at 640×480 with per-clip frame rate and a length of 2–120 frames.

Table 17: Release statistics of the **Humanoids** benchmark. The synthetic component provides synchronized RGB, robot state, and FK-derived 3D keypoints; the real-world component provides image-space evaluation only. Counts are taken from the released manifest.

Component	Clips	Frames
Synthetic — Isaac Sim	4,751	502,816
Synthetic — MuJoCo	2,134	229,365
Synthetic — total	6,885	732,181
Real-world — total	365	56,437

Property	Value
Humanoid embodiments	12
Unique source motions	3,706
Synthetic render duration	6 h 46 m
Real-world duration	31 m 30 s
Resolution	640×480
Frame rate	30 fps
Clip length	2–120 frames
Source subsets (clips)	BMLrub 3,432 / CMU 2,227 / HDM05 1,226

Table 18: Per-robot coverage of the synthetic component. "Isaac / MuJoCo" reports the clip count rendered by each simulator. The 12 embodiments span 19–31 controllable degrees of freedom and are shared across train, validation, and test; this is not an unseen-robot benchmark.

Robot	DoF	Bodies	Clips	Frames	Isaac / MuJoCo
BerkeleyHumanoidLite	22	26	566	60,930	385 / 181
BoosterK1	22	23	561	59,349	386 / 175
BoosterT1	23	24	569	60,125	405 / 164
EngineAI_PM01	24	29	522	55,299	357 / 165
FourierGR3	31	39	560	59,563	394 / 166
FourierN1	23	29	610	65,440	425 / 185
G1	29	28	582	62,081	407 / 175
KuavoS45	28	29	611	65,063	412 / 199
PNDAdamLite	25	26	558	59,255	390 / 168
StanfordToddy	22	33	582	61,238	390 / 192
TienKung	20	24	577	61,281	394 / 183
UnitreeH1	19	20	587	62,557	406 / 181
Total	–	–	6,885	732,181	4,751 / 2,134

B.2 MOTION RETARGETING AND SIMULATION

Human motions from the HDM05, CMU, and BMLrub subsets of AMASS are retargeted to each robot's native joint configuration with GMR, producing robot-native joint angles, root pose, and

Table 19: Motion-held-out split of the synthetic component. All renderings of one source motion, across robots, simulators, and scenes, are assigned to the same split, so no source motion crosses a split boundary.

Split	Clips	Frames	Duration	Source motions
Train	5,400	574,381	5 h 19 m	3,000
Validation	652	69,914	0 h 39 m	330
Test	833	87,886	0 h 49 m	376
Total	6,885	732,181	6 h 46 m	3,706

Table 20: Rendering composition of the synthetic component. Isaac scenes use a static camera within a clip that is re-randomized across clips together with robot placement, distance, and lighting; MuJoCo clips use one of two orbit camera policies. No accepted clip uses synthetic occluders.

Renderer	Scene / camera policy	Clips	Frames
Isaac	`city_anchor` (10 anchors)	1,861	183,513
Isaac	`isaac_simple_warehouse`	835	92,310
Isaac	`isaac_outdoor_rivermark_safe`	760	83,068
Isaac	`brownstone_origin_v1`	607	67,484
Isaac	`residential_lobby_origin_v1`	455	50,763
Isaac	`isaac_office_origin_v1`	233	25,678
MuJoCo	`fixed_orbit`	1,733	185,295
MuJoCo	`weak_follow`	401	44,070

FK-derived body positions. Each trajectory is then replayed in Isaac Sim and MuJoCo, and the synchronized RGB frames, camera parameters, joint states, root poses, and Canonical-18 3D keypoints are recorded directly from the simulator state.

Rendering is randomized across clips. Isaac clips use a static camera within a clip that is re-randomized across clips, together with robot placement, camera distance, and lighting; the replay synchronizes simulation and capture with a single physics step followed by a zero-delta-time capture. MuJoCo clips use either a fixed-orbit or a weak-follow camera policy. No accepted clip uses synthetic occluders. Table 20 lists the scene and camera composition, and the per-robot Isaac/MuJoCo split is given in Table 18.

B.3 Quality Control Procedure

Because all synthetic supervision is derived from simulator state, quality control is defined on the state-to-observation mapping rather than on manual annotation. We verify (i) structural validity (required fields, finite values, tensor shapes), (ii) joint-limit legality, (iii) root-correction invariance (joint angles, root XY, and root rotation are unchanged by grounding, and the recorded Z correction matches the applied offset), (iv) FK consistency between the released body geometry and the robot model, (v) image-space reprojection consistency between the released 3D keypoints, camera, and 2D labels, and (vi) ground non-penetration. Table 21 reports the maximum deviation of each check over the full release.

Trajectory grounding. Foot-ground consistency is handled by a route-based grounding stage. Clips with reliable contact information use a physics-based contact-aware route (PBHC) that preserves the source root; the remaining clips use a support-segment median root-Z fallback, in which the root height is set from the supporting segment while the source joint angles and airborne semantics are preserved. Table 22 compares the grounding routes and shows that the released v3 fallback reduces the support-gap P95 from 8.89 cm to 3.47 cm relative to the constant v2 fallback, while geometry non-penetration and root preservation are maintained.

Teleport and outlier removal. Source trajectories are screened for joint-velocity and root-position discontinuities. Two clips containing confirmed source teleports are excluded from the released manifest (6,887 $\rightarrow$ 6,885), and no further trajectory is altered by the grounding stage.

Table 21: Synthetic quality-control checks over the full release. "Scope" is the audited population; every check reports the maximum observed deviation or the number of failures.

Check	Scope	Result
Required fields / finite / shapes	6,885 clips	0 failures
Joint-limit legality	6,885 clips	0 violations (max 4.78×10^{-5} deg)
Root-correction invariance (q, root XY, rotation)	6,885 clips	0 altered
FK consistency	20.1M body points	max 2.50×10^{-4} mm
2D reprojection consistency	13.0M valid keypoints	max 1.48×10^{-2} px
Ground non-penetration	6,885 clips	0 violations (max 8.46×10^{-5} mm)
Source teleport removal	6,887 $\rightarrow$ 6,885 clips	2 confirmed
Split integrity (motion leakage)	3,706 source motions	0 cross-split

Table 22: Grounding-route summary over the pre-exclusion QC universe (6,887 clips; the released manifest excludes the two teleport clips). "Fallback ratio" is the fraction of clips handled by a fallback route; "root deviation" is the support-gap P95 of the route (PBHC preserves the original root exactly). The released package uses the v3 support-segment fallback.

Version / route	#clips	Penetration	Fallback ratio	Root deviation
PBHC (declared contacts)	1,199	0	0.0%	0 m
Constant fallback (v2)	5,688	0	82.6%	8.89 cm
Support-segment fallback (v3, released)	5,688	0	82.6%	3.47 cm

C CANONICAL-18 KEYPOINT DEFINITION

To enable consistent pose annotation and evaluation, we define a Canonical-18 keypoint representation for humanoid robots. The keypoints describe the semantic full-body structure of a humanoid robot and are instantiated according to each robot's kinematic structure and geometry. Rather than strictly following URDF joint origins, each keypoint is defined according to the semantic center of the corresponding body part. For rigid body components, we prioritize stable geometric or semantic centers that remain consistent across viewpoints.

Figure 6 illustrates the Canonical-18 keypoint layout. The keypoints are designed to capture the major body parts of humanoid robots, including the torso, head, upper limbs, and lower limbs.

The Canonical-18 keypoints are defined as follows:

For joint-like structures (e.g., shoulders, elbows, knees, and ankles), keypoints are placed near the corresponding joint centers. For rigid components (e.g., torso, head, and feet), keypoints represent the semantic center of the visible structure rather than a specific mesh vertex or joint origin. The left/right convention follows the robot's own perspective.

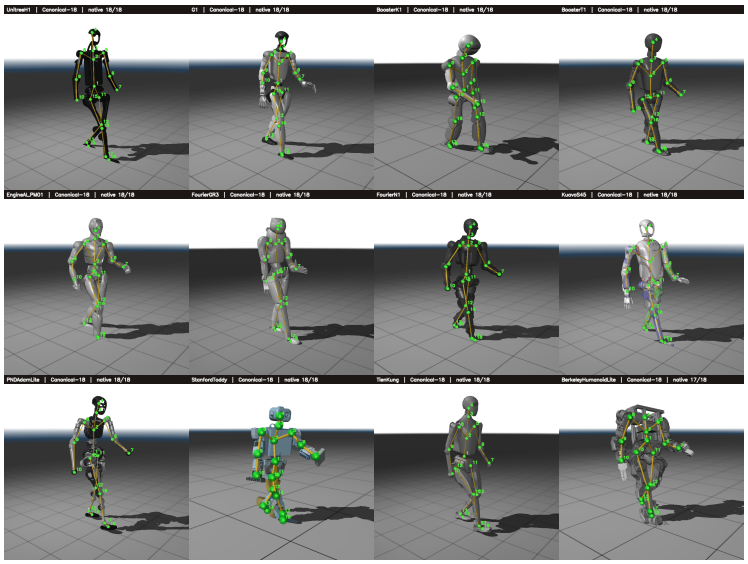


Figure 6: Visualization of the Canonical-18 keypoint representation for humanoid robots.

Index	Keypoint
1	Pelvis
2	Torso
3	Neck
4	Head
5	L-Shoulder
6	L-Elbow
7	L-Wrist
8	R-Shoulder
9	R-Elbow
10	R-Wrist
11	L-Hip
12	L-Knee
13	L-Ankle
14	L-Foot
15	R-Hip
16	R-Knee
17	R-Ankle
18	R-Foot

Table 23: Canonical-18 keypoint definition.

D Implementation Details of Kinematic Grounding and Refinement

D.1 FK-conditioned Geometric Features

The proposed refinement module does not operate on the initial pose alone. Instead, it receives explicit geometric features grounded by the robot kinematic model. Given the initial robot state (q_0, R_0, t_0), we first obtain the metric robot structure through forward kinematics:

$$X_0 = \mathrm{FK}_r(q_0), \tag{4}$$

where $X_0 \in \mathbb{R}^{J\times 3}$ denotes the metric 3D robot structure (Canonical-18 keypoints) in the root frame, determined by the embodiment-specific kinematic tree and link geometry.

The initial 3D structure is transformed into the camera coordinate system:

$$X_{0,j}^{C} = R_0 X_{0,j} + t_0. \tag{5}$$

We then project the grounded structure into the image plane:

$$\hat{x}_0 = \Pi_K(X_0^C), \tag{6}$$

and compute the reprojection residual:

$$e_{\text{reproj}} = \hat{x} - \hat{x}_0. \tag{7}$$

The final refinement feature is constructed by combining the initialized state and FK-derived geometric cues:

$$\phi = [q_0, R_0, t_0, X_0, \hat{x}_0, e_{\text{reproj}}]. \tag{8}$$

This explicit grounding converts the initial neural prediction into a robot-specific geometric representation before refinement.

D.2 Bounded Residual Parameterization

The refinement module predicts bounded corrections instead of directly regressing a new robot pose. Specifically, the predicted residuals are parameterized as:

$$\begin{aligned} \Delta q &= \alpha_q \tanh(g_q), \\ \Delta R &= \text{Exp}([\alpha_R \tanh(g_R)]_\times), \\ \Delta t &= \alpha_t \tanh(g_t), \end{aligned} \tag{9}$$

where α_q, α_R, and α_t control the maximum correction magnitude.

The final robot state is updated as:

$$\begin{aligned} q &= \text{clip}(q_0 + \Delta q, q_{\min}, q_{\max}), \\ R &= \Delta R\, R_0, \\ t &= t_0 + \Delta t. \end{aligned} \tag{10}$$

The bounded formulation preserves the validity of the initial estimate and prevents the refinement module from producing physically implausible states. The joint-limit projection further guarantees that the refined articulation remains within the feasible range of each robot embodiment.

D.3 Training Objective

The refinement module is trained with multiple objectives defined in both joint space and Cartesian space. The overall objective is:

$$\mathcal{L} = \lambda_q \mathcal{L}_q + \lambda_{\text{FK}} \mathcal{L}_{\text{FK}} + \lambda_R \mathcal{L}_R + \lambda_t \mathcal{L}_t + \lambda_{\text{cam}} \mathcal{L}_{\text{cam}} + \lambda_{\text{repr}} \mathcal{L}_{\text{repr}} + \lambda_\Delta \mathcal{L}_\Delta. \tag{11}$$

The joint-space loss is defined as:

$$\mathcal{L}_q = \text{SmoothL1}(q, q^*). \tag{12}$$

To explicitly enforce robot geometry consistency, we additionally apply a forward-kinematic loss:

$$\mathcal{L}_{\text{FK}} = \left\| \text{FK}_r(q) - \text{FK}_r(q^*) \right\|_1. \tag{13}$$

The root pose losses constrain the global camera-relative configuration, with d_{geo} the geodesic distance on $\mathrm{SO}(3)$:

$$\mathcal{L}_R = d_{\text{geo}}(R, R^*), \tag{14}$$

$$\mathcal{L}_t = \mathrm{SmoothL1}(t, t^*). \tag{15}$$

We further apply camera-space body alignment and image-space reprojection constraints:

$$\mathcal{L}_{\text{cam}} = \left\| RX(q) + t - (R^* X(q^*) + t^*) \right\|_1, \tag{16}$$

$$\mathcal{L}_{\text{repr}} = \sum_j w_j \left\| \Pi_K(RX_j(q) + t) - \hat{x}_j \right\|_2. \tag{17}$$

Finally, a residual regularization term encourages minimal corrections:

$$\mathcal{L}_\Delta = \|\Delta q\|_2^2 + \|\Delta R - I\|_F^2 + \|\Delta t\|_2^2. \tag{18}$$

D.4 Inference Behavior

The refinement module is a feed-forward prediction module. During inference, it performs a single residual update from the initialized state. No test-time optimization, inverse kinematics fitting, or parameter update is performed.

Therefore, the proposed refinement differs from optimization-based pose fitting methods: the robot kinematic model is used as an explicit geometric constraint during both training and inference, while the final correction is predicted directly by the learned refinement network.

E Additional Implementation Details

This section lists the concrete network sizes and training settings behind the frozen-model numbers reported in the main paper. All components share the same frozen visual encoder and are trained independently per robot.

E.1 Network Architecture

Visual encoder. We use a frozen V-JEPA ViT-B video encoder with patch size 16, tubelet size 2, embedding dimension 768, and 12 transformer blocks, operating on 16-frame clips at 256×256 resolution. The input frames are the detector crops, normalized with the ImageNet mean/std used during DAPT pre-training. Each 16-frame window is encoded once into 8 tubelets of 16×16 spatial tokens (768-d). The encoder contains 86.8M parameters; all of them are frozen (`requires_grad=False`, eval mode) and only the task heads are optimized.

Articulation head. The native-q branch attends from 18 learnable joint queries (256-d) to the target tubelet's spatial tokens (projected by LayerNorm and a $768 \to 256$ linear layer); each query is conditioned on the corresponding predicted Canonical-18 keypoint through an MLP $3 \to 256$. It consists of one 8-head cross-attention block (256-d, dropout 0.1) followed by a feed-forward block $256 \to 512 \to 256$ (GELU, dropout 0.1), totaling $\sim$3.1M trainable parameters.

Root pose heads. The root rotation is predicted by a two-hidden-layer MLP ($768{+}128{+}8 \to 512 \to 256 \to 6$) with LayerNorm, GELU and dropout 0.1, regressing a 6D rotation ($\sim$0.6M parameters). The camera-relative root translation is predicted by a separate simple head ($768{+}128{+}8 \to 512 \to 512 \to 3$, LayerNorm and GELU) whose three outputs are converted analytically into a full-image camera translation ($\sim$0.7M parameters). Both heads consume the frozen mean-pooled target tubelet token, the predicted Canonical-18, the normalized bounding box and the normalized camera intrinsics.

Refinement module. The bounded refiner is a three-layer MLP (LayerNorm $\rightarrow$ Linear $d_{\text{in}} \rightarrow 384 \rightarrow 256 \rightarrow (d_q{+}6)$, GELU) whose last layer is zero-initialized, with $d_{\text{in}} = d_q + 6 + 3 + 3{\cdot}18 + 2{\cdot}18 + 2{\cdot}18 + 3$ ($\sim$0.17M parameters).

E.2 Training Configuration

Inputs and windows. All branches share the same data interface: 16-frame clips sampled at 4 FPS from 30 FPS source video, 256×256 detector crops, the predicted Canonical-18 keypoints, the normalized bounding box, and calibrated intrinsics. No additional photometric or geometric augmentation is applied beyond the detector crops and the ImageNet normalization; clips are shuffled within each epoch.

Optimization. All heads are optimized with AdamW. The articulation head is trained in mixed precision (AMP) with gradient scaling, and the refinement module is trained on cached frozen features.

Component	LR	Weight decay	Batch size	Epochs
Articulation head (q)	1e$-$4	1e$-$3	8	15
Root rotation head	1e$-$4	1e$-$3	16	15
Root translation head	1e$-$4	1e$-$4	8	15
Refinement module (E1$'$)	2e$-$4	1e$-$3	128	15

Table 24: Optimization settings. All runs use AdamW; the visual encoder is frozen and receives no gradient. Model selection uses the validation split only.

Objectives. The articulation head minimizes $\mathcal{L}_q + 0.1\,\mathcal{L}_{\text{FK}}$ with $\mathcal{L}_q = \text{SmoothL1}(q, q^*)$ and $\mathcal{L}_{\text{FK}} = \|\,\text{FK}_r(q) - \text{FK}_r(q^*)\|_1$; the per-joint smooth-L1 term is re-weighted by the physical joint range. The root rotation head uses a geodesic loss $\mathcal{L}_R = d_{\text{geo}}$ and the translation head a smooth-L1 loss with a normalized target. The refinement module combines $\mathcal{L}_q + 0.1\,\mathcal{L}_{\text{FK}} + 10^{-3}\|\Delta q\|_2^2 + 0.5\,\mathcal{L}_R + 0.5\,\mathcal{L}_t + 0.1\,\mathcal{L}_{\text{cam}} + 0.1\,\mathcal{L}_{\text{repr}}$, with gradients clipped to norm 1.0. Its residual bounds are $\alpha_q = 0.1\,(q_{\max} - q_{\min})$, $\alpha_R = 10^\circ$ and $\alpha_t = 0.2\,\text{m}$.

Computational cost. We use the same two-stage recipe for every robot. The articulation head is trained for 15 epochs over 2,629 clips, each expanded to 8 tubelets, and accounts for almost all of the cost ($\sim$2 h on a single RTX 4090); caching the frozen features for the refinement stage takes $\sim$5 min, and the refinement module itself $\sim$2 min. Training one embodiment therefore takes about 2–3 h on one GPU.

Robot-specific training. Each robot is trained independently with its own `RobotSpec` (kinematic tree, joint limits, and Canonical-18 binding). We report the macro average over the 12 humanoid embodiments used in the main experiments.

F Limitations

HumanoidPose is trained and evaluated on a fixed set of 12 embodiments, and each robot is trained *independently*; we do not study a single model that generalizes across embodiments, and evaluation on unseen robot morphologies (zero-shot cross-embodiment transfer) is left to future work. In addition, our real-world evaluation uses manually verified *2D* keypoints rather than metric 3D ground truth, because accurate 3D annotations are unavailable for real humanoid videos; real-world 3D accuracy is therefore not directly measured. Finally, we assume a single humanoid in view and do not address multi-robot or human-robot interaction scenes.

G Evaluation Metrics

We evaluate humanoid pose recovery from both joint-space and Cartesian-space perspectives. For synthetic evaluation, where robot joint states and FK-derived 3D keypoints are available, we report

joint angle error, 3D keypoint errors, and pose alignment metrics. For real-world evaluation, where metric 3D ground truth is unavailable, we evaluate image-space keypoint recovery using manually verified 2D annotations.

G.1 Synthetic 3D Pose Metrics

Mean Per-Joint Angle Error (MPJAE). MPJAE measures the average angular difference between the predicted and ground truth robot joint configurations:

$$\mathrm{MPJAE} = \frac{1}{J}\sum_{j=1}^{J} \|q_j - \hat{q}_j\|_2 \,, \tag{19}$$

where q_j and $\hat{q}_j$ denote the ground-truth and predicted joint angles, respectively.

Mean Per-Joint Position Error (MPJPE). MPJPE measures the Euclidean distance between predicted and ground-truth 3D keypoints:

$$\mathrm{MPJPE} = \frac{1}{J}\sum_{j=1}^{J} \left\| X_j - \hat{X}_j \right\|_2 \,, \tag{20}$$

where X_j and $\hat{X}_j$ denote the ground-truth and predicted 3D keypoints.

Procrustes-Aligned MPJPE (PA-MPJPE). PA-MPJPE evaluates pose structure after removing global translation, rotation, and scale differences through Procrustes alignment:

$$\text{PA-MPJPE} = \frac{1}{J}\sum_{j=1}^{J} \left\| X_j^* - \hat{X}_j^* \right\|_2 \,, \tag{21}$$

where X^* and $\hat{X}^*$ denote the aligned poses.

Global MPJPE (G-MPJPE). G-MPJPE evaluates the absolute 3D pose accuracy in the camera coordinate system without alignment:

$$\text{G-MPJPE} = \frac{1}{J}\sum_{j=1}^{J} \left\| X_j^C - \hat{X}_j^C \right\|_2 . \tag{22}$$

Percentage of Correct Keypoints (PCK). PCK measures the percentage of keypoints whose Euclidean error is below a distance threshold:

$$\mathrm{PCK@}\alpha = \frac{1}{J}\sum_{j=1}^{J} \mathbb{I}\left(\left\| X_j - \hat{X}_j \right\|_2 < \alpha \right). \tag{23}$$

We report PCK@0.3 for synthetic 3D evaluation.

G.2 Real-world 2D Pose Metrics

Since real-world videos do not provide metric 3D ground truth, we evaluate image-space keypoint recovery using manually verified Canonical-18 2D annotations.

2D Mean Per-Joint Position Error (2D MPJPE). 2D MPJPE measures the average pixel distance between predicted and annotated keypoints:

$$\text{2D-MPJPE} = \frac{1}{J}\sum_{j=1}^{J} \|x_j - \hat{x}_j\|_2 \,. \tag{24}$$

Normalized Mean Error (NME). NME normalizes the 2D keypoint error by the size of the target bounding box:

$$\text{NME} = \frac{\frac{1}{J}\sum_{j=1}^{J} \|x_j - \hat{x}_j\|_2}{d_{\text{bbox}}}, \tag{25}$$

where d_{bbox} denotes the bounding-box normalization factor.

2D Percentage of Correct Keypoints (PCK). PCK measures the ratio of keypoints within a normalized distance threshold:

$$\text{PCK@}\alpha = \frac{1}{J}\sum_{j=1}^{J} \mathbb{I}\left(\frac{\|x_j - \hat{x}_j\|_2}{d_{\text{bbox}}} < \alpha\right). \tag{26}$$

We report PCK@0.1 for real-world evaluation.

H Baseline Details

We evaluate representative *Human Pose Estimation (HPE)* and *Robot Pose Estimation (RPE)* methods under a unified HRPE protocol. Every baseline output is converted into the robot-native joint configuration q, with 3D keypoints obtained through the target embodiment's forward kinematics, and all methods are scored with the same evaluator, target localization, and data splits as **Humanoid-Pose**.

HPE baselines. We consider HMR2 (Goel et al., 2023), GVHMR (Shen et al., 2024), and WHAM (Shin et al., 2024), which produce human-centric whole-body pose representations. We evaluate two settings. (1) *Direct application with human-to-robot retargeting*: no additional training is performed; the original HPE weights, tracker, and keypoint estimator are preserved, and only the input crop is replaced by the shared frozen robot bounding box used for all methods (see the shared evaluation protocol below). The predicted SMPL(-X) bodies are mapped to the target humanoid's native joints with a frozen GMR retargeter (Araujo et al., 2025), followed by target localization and camera-relative root composition. (2) *Robot-specific adaptation*: the output representation is replaced by a robot-specific head that predicts the native joint configuration and the camera-relative root pose, trained on the Humanoids synthetic split while the visual frontend is kept frozen.

RPE baselines. We consider RoboPose (Labbé et al., 2021), HoRoPose (Ban et al., 2024), and RoboPEPP (Goswami et al., 2025). We adapt their output representation to each target humanoid embodiment and train them on the corresponding synthetic data following their original formulations. As these methods target seen embodiments, they are reported only under the adaptation setting.

Shared evaluation protocol. For all methods, the robot is defined by its Canonical-18 keypoint specification and kinematic model, and forward kinematics is computed with the same implementation. Target localization is shared across methods: all baselines use the same frozen robot bounding box for input cropping and the same normalization scale $s = \max(w, h)$ for NME and PCK, so that per-method detectors do not affect the main ranking. For real-world 2D evaluation we assume a fixed intrinsic matrix ($f_x = f_y = \max(W, H)$, $c_x = W/2$, $c_y = H/2$), whereas 3D evaluation uses the benchmark-provided calibrated intrinsics. On the internal real-world set, all twelve embodiments use human-reviewed 2D annotations. The synthetic split is motion-disjoint, the internal

real-world split is source-disjoint, and the external DHRP-H1 benchmark is used only for zero-shot evaluation.

Reporting rules. We never rank zero-shot and adapted baselines together without stating the setting, and we do not mix different crop sources, test subsets, or evaluators in a single ranked table. Failed detections are retained in the denominator, and all methods in a table share the same bounding box, normalization scale, split, and evaluator version.

Fairness of comparison. All methods are trained on the same synthetic split with the same per-embodiment protocol, and are evaluated with the same bounding boxes, normalization, splits, and evaluator. Target localization and evaluation are never tuned per method, and model selection uses the same held-out validation split. The HPE and RPE variants therefore differ from **HumanoidPose** primarily in their pose representation and prediction module, isolating the effect of the proposed robot-native design.

I REAL-2D PCK THRESHOLD SWEEP

Tables 25 and 26 report PCK at thresholds 0.3, 0.1, and 0.05 under exactly the same contract as Table 6: the Humanoids-Real column is the 12-robot robot-balanced average under the shared frozen robot bounding box, and the DHRP-H1 column uses the external instance-id contract. PCK@0.1 is the primary threshold reported in the main paper.

Table 25: PCK threshold sweep on Humanoids-Real (12 robots) under the same evaluation protocol as Table 6. All methods use the same frozen robot bounding box for NME/PCK normalization; 2D MPJPE is in pixels and all PCK values are percentages. PCK@0.1 is the primary threshold.

Category	Method	2D MPJPE (px)↓	NME↓	PCK@0.3 (%)↑	PCK@0.1 (%)↑	PCK@0.05 (%)↑
HPE+GMR	HMR2 + GMR	247.77	0.4067	38.98	5.77	1.61
	GVHMR + GMR	78.82	0.1371	89.49	53.27	25.83
	WHAM + GMR	197.97	0.3098	56.55	15.81	4.90
HPE⋆	HMR2⋆	162.98	0.2607	73.04	15.94	4.46
	GVHMR⋆	149.27	0.2432	69.93	22.97	7.07
	WHAM⋆	243.53	0.4048	49.17	7.49	1.90
RPE	RoboPEPP⋆	139.18	0.2157	78.52	23.44	6.66
	HoRoPose⋆	128.32	0.2023	78.09	28.72	12.06
	RoboPose⋆	152.10	0.2356	74.38	20.64	6.57
HRPE	**HumanoidPose (Ours)**	71.49	0.1180	90.60	61.97	42.13

Table 26: PCK threshold sweep on DHRP-H1 under the same external protocol as Table 6. 2D MPJPE is in pixels and all PCK values are percentages. PCK@0.1 is the primary threshold.

Category	Method	2D MPJPE (px)↓	NME↓	PCK@0.3 (%)↑	PCK@0.1 (%)↑	PCK@0.05 (%)↑
HPE+GMR	HMR2 + GMR	253.68	0.3285	49.47	10.61	2.29
	GVHMR + GMR	97.62	0.1470	87.40	62.06	28.09
	WHAM + GMR	491.47	0.6015	8.70	0.99	0.31
HPE⋆	HMR2⋆	146.48	0.1941	87.48	15.73	1.91
	GVHMR⋆	98.81	0.1481	86.87	60.69	32.60
	WHAM⋆	406.34	0.5003	16.26	2.06	0.46
RPE	RoboPEPP⋆	282.25	0.3684	50.99	8.40	1.53
	HoRoPose⋆	208.18	0.2830	66.18	17.48	5.34
	RoboPose⋆	212.69	0.2901	66.03	14.73	4.05
HRPE	**HumanoidPose (Ours)**	87.80	0.1363	86.41	74.20	49.54