RadarFlowPose: Vision-Inspired Coarse-to-Fine Skeleton Refinement with Flow Matching

Kailu Guo **Akram Alomainy** **Khalid Z. Rajab**
School of Electronic Engineering and Computer Science
Queen Mary University of London
kailu.guo@qmul.ac.uk

Abstract

Radar-based human pose estimation is a promising alternative to camera-based perception, but remains challenging due to sparse observations, noisy measurements, and severe pose ambiguity. Existing methods infer both global skeletal structure and flexible peripheral joints from sparse radar observations, often compromising structural plausibility and fine-grained local accuracy. Motivated by the coarse-to-fine nature of human visual perception, we propose RadarFlowPose, a framework that decomposes radar-based human pose estimation into coarse pose prior generation and flow-based refinement. The coarse model first produces a structurally plausible and temporally coherent pose prior from sparse radar observations, providing a global initialization for subsequent refinement. Instead of estimating the full pose from scratch, the refinement stage focuses on correcting the remaining local ambiguities and motion-induced errors. By incorporating localised Doppler motion cues into the flow-matching process, the model performs motion-aware residual refinement around the coarse prior, leading to more accurate and anatomically consistent pose estimates. Experiments on three benchmark radar pose datasets show consistent effectiveness, especially in motion-sensitive and fine-grained joint-level evaluation.

1 Introduction

Human pose estimation is a fundamental problem in behaviour understanding, human-computer interaction, and intelligent sensing [1, 2], with growing relevance to remote health monitoring [3–6]. Compared with cameras and videos, millimeter-wave (mmWave) radar is well suited to these requirements due to its privacy-preserving nature, robustness under low-light conditions, and robustness to illumination changes and partial occlusions [7, 8]. Despite recent progress, robust pose recovery from mmWave radar point clouds remains challenging [9]. Radar point clouds are sparse, noisy, and incomplete, and are further affected by motion-dependent scattering, viewpoint changes, and multipath effects [10–13]. Such limited and unstable cues make pose recovery ambiguous, especially for weakly observed peripheral joints.

Existing point-cloud-based radar human pose estimation methods have made substantial progress by improving radar point encoding [14, 15] and exploiting temporal or relational cues to improve robustness under sparse and noisy mmWave observations [16]. These designs enable regression-based models to extract useful pose cues from sparse and incomplete radar point clouds [17]. However, most methods still formulate pose estimation as a fixed mapping from radar observations to joint coordinates, making it difficult to simultaneously preserve global skeletal plausibility and recover fine-grained local joint variations.

This limitation motivates us to revisit radar-based human pose recovery from a coarse-to-fine perspective. Human visual perception is often characterized by coarse-to-fine processing: low spatial-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

frequency information provides global cues such as contours, layouts, and overall shapes, while high spatial-frequency information encodes fine details such as edges, textures, and local variations [18]. Inspired by this principle, we argue that radar-based pose estimation should first establish a reliable global skeletal prior before resolving local joint ambiguities. Following this idea, we propose RadarFlowPose, a coarse-to-fine framework for radar-based 3D human pose estimation. Instead of directly regressing the full skeleton in a single step, RadarFlowPose first establishes a coarse pose and then refines it through a flow-based process in skeleton space. Our contributions are summarized as follows.

1. We design a temporally informed coarse pose generator that produces an anatomically plausible skeleton prior from sparse radar observations, capturing the global body configuration before refinement.

2. To the best of our knowledge, we make the first attempt to introduce Flow Matching into radar-based low-dimensional skeleton pose estimation. The proposed refiner performs motion-aware residual correction around the coarse prior.

3. We validate RadarFlowPose on three benchmark radar pose datasets with different skeleton definitions, demonstrating the effectiveness and generality of the proposed framework.

2 Related Work

2.1 Radar-based Human Pose Estimation

For point-cloud-based mmWave human pose estimation, existing methods can be broadly categorized by their point encoding strategies. The first line relies on point-based or neighborhood-based encoders, such as PointNet [19] and DGCNN [20], to extract geometric features from sparse and unordered radar points. Early work such as mm-Pose [14] demonstrates the feasibility of recovering human skeletons from sparse mmWave observations, while later methods improve local aggregation, temporal fusion, and relational reasoning. For example, FUSE [15] exploits multi-frame point cloud fusion for improved robustness, and mmGAT [21] introduces graph attention to capture fine-grained dependencies among radar points. The second line emphasizes global context modeling with attention-based encoders, such as PointTransformer [22], to capture broader dependencies among sparse radar observations. Representative works include HuPR [16], which combines attention-based fusion with pose refinement and establishes a benchmark for mmWave pose estimation, and TransHuPR [23], which explores transformer-based cross-view context aggregation for radar pose recovery.

Despite these advances, most methods still rely on deterministic regression, which can average over ambiguous pose hypotheses under sparse and noisy radar observations. This often leads to over-smoothed predictions and reduced structural fidelity.

2.2 Diffusion Pose Generation

Diffusion models have been widely used for point cloud generation, where data points are progressively corrupted into noise and then recovered through denoising. Prior studies have shown their effectiveness in modeling smooth object surface distributions, such as chairs and airplanes [24, 25]. However, human skeletons differ from general point clouds: they are low-dimensional structured representations with explicit topological and kinematic constraints, rather than unordered surface samples. Recent pose-oriented diffusion methods have explored iterative denoising for 3D human pose estimation, including 2D-conditioned pose lifting [26, 27] and radar-based pose recovery with enhanced radar conditions or coarse pose priors [28–30]. These works demonstrate the potential of generative refinement for ambiguous pose estimation, but mainly rely on reverse denoising processes.

Flow Matching [31] offers an alternative formulation by learning a continuous velocity field that transports samples from an initial distribution to the target distribution. Recent studies have extended FM to structured prediction by incorporating relational information [32], respecting geometric pose spaces [33], or constructing multiple hypotheses for 2D-to-3D pose prediction [34]. Nevertheless, applying FM to articulated 3D skeleton refinement, especially under sparse and noisy radar observations, remains underexplored. Our work addresses this gap by introducing FM as a condition-guided refinement mechanism in skeleton space and analyzing its benefit for radar-based human pose estimation.

3 Method

This section presents the proposed RadarFlowPose framework, as shown in Fig. 1. Sec. 3.1 formulates the radar-based 3D pose estimation problem, Sec. 3.2 introduces the Coarse Pose Generator, Sec. 3.3 presents the structure-conditioned flow refinement process, and Sec. 3.4 describes the training objectives.

3.1 Problem Formulation

Given an input radar sequence $R \in \mathbb{R}^{T\times N\times C}$, our goal is to recover the corresponding 3D human skeleton sequence $X \in \mathbb{R}^{T\times J\times 3}$, where T, J, N, and C denote the sequence length, number of body joints, radar points per frame, and point feature dimension, respectively. Rather than adopting standard Flow Matching that transports samples from a random source distribution [31], we cast pose estimation as a coarse-to-fine skeletal refinement problem. Since the target skeleton is governed by anatomical joints, kinematic connectivity [35], and temporal coherence, random-source transport introduces an unnecessarily broad search space. We therefore decompose pose recovery into two stages: a coarse model predicts a skeletal prior X_c, and a conditional flow refiner transports a perturbed source state X_0 toward the ground-truth pose X_1. In this way, Flow Matching models structured residual refinement rather than unconstrained generation from noise.

Specifically, given the coarse prior X_c, we construct a perturbed source state during training:

$$X_0 = X_c + \sigma_{\text{noise}}(1-t)\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{1}$$

where $\sigma_{\text{noise}} = 0.01$ controls the maximum perturbation magnitude, and $t \sim \text{Beta}(2, 2)$ denotes the sampled flow time. The coarse prior is detached during refinement training to prevent gradient propagation.

We denote the ground-truth pose as $X_1 = X_{\text{gt}}$ and construct an intermediate state using smoothstep interpolation:

$$X_t = (1-\alpha(t))X_0 + \alpha(t)X_1, \qquad \alpha(t) = 3t^2 - 2t^3, \tag{2}$$

where $X_t \in \mathbb{R}^{T\times J\times 3}$. Treating the sampled source X_0 as fixed, the target velocity is

$$v_t^\star = \dot{\alpha}(t)(X_1 - X_0), \qquad \dot{\alpha}(t) = 6t(1-t). \tag{3}$$

We learn a radar- and source-conditioned velocity field $v_\theta(X_t, t, X_0, R)$ using a joint- and time-weighted Flow Matching objective:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{R,X_0,X_1,t}\left[\frac{1}{3J}\sum_{j=1}^{J}\lambda_t(t)\lambda_j \left\|v_{\theta,j}(X_t, t, X_0, R) - v_{t,j}^\star\right\|_2^2\right], \tag{4}$$

where $\lambda_t(t) = 1 + 2t$ emphasizes later refinement states, and λ_j is set to 3 for end-effectors and 1 for other joints.

During inference, refinement starts from the unperturbed coarse pose X_c, using Euler integration as detailed in Sec. 3.3.

3.2 Coarse Pose Generator

Coarse Pose Generator serves as a structural anchor for subsequent refinement. Rather than directly regressing joint coordinates from sparse radar point clouds, we mapped radar observations into a compact latent representation and decode it into a coarse skeleton sequence with explicit kinematic structure. As illustrated in Fig. 2, this stage consists of three steps: Structured Radar Latent Construction (SRL), Latent Temporal Aggregation (LTA), and Kinematic Prior Construction (KPC).

Structured Radar Latent Construction. For each radar frame, let $R_t = \{r_{t,n}\}_{n=1}^{N}$ denote the input point cloud, where each point consists of physical attributes $r_{t,n}^{\text{phys}}$ and a time-related feature $\tau_{t,n}$. We first constructed point features as

$$h_{t,n} = \phi_{\text{phys}}(r_{t,n}^{\text{phys}}) + \phi_{\text{time}}(\tau_{t,n}), \tag{5}$$

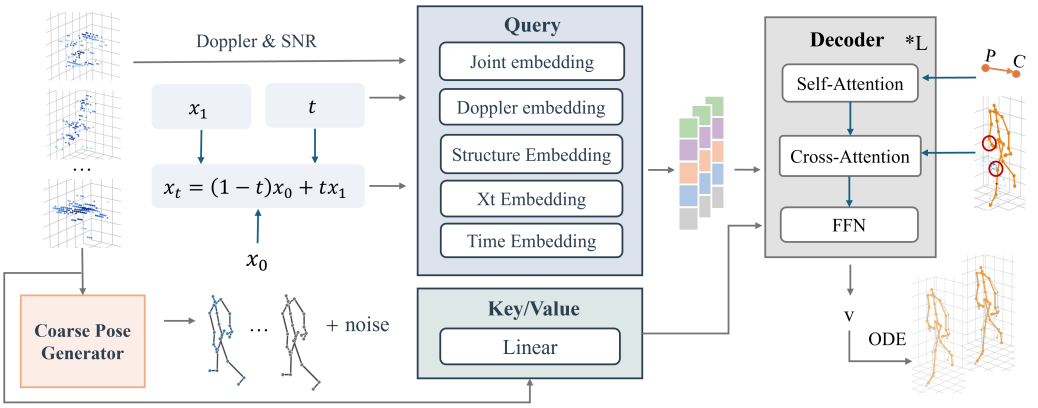


Figure 1: Overall architecture of RadarFlowPose. The coarse pose generator predicts a skeletal prior X_c from radar observations. During training, a perturbed source X_0 is used to construct X_t, while the structure-conditioned flow module predicts the velocity field v using pose-aware queries and radar-based keys/values. During inference, refinement starts from X_c, and Euler integration produces the final 3D pose sequence.

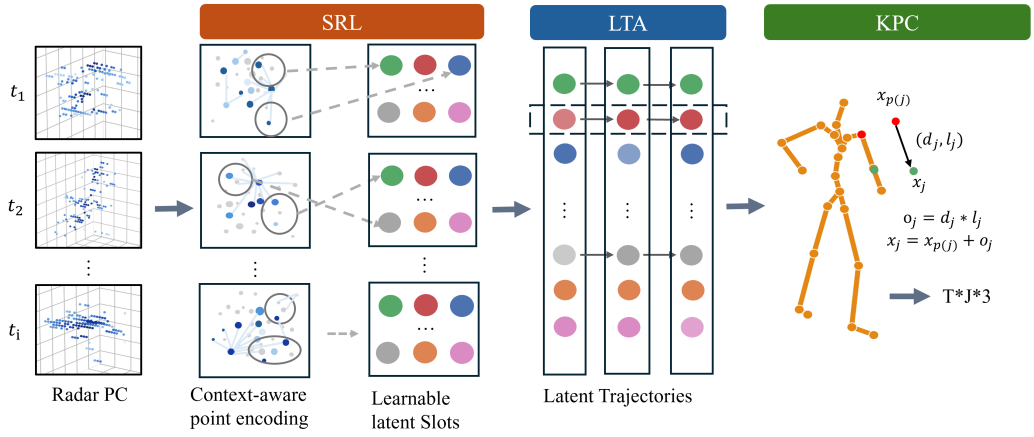


Figure 2: Coarse Pose Generation. The SRL module encodes each radar point cloud into context-aware latent slots, and the LTA module aggregates them over time to form latent trajectories. The KPC module then predicts bone offsets and reconstructs the coarse skeleton through forward kinematics, producing the structural prior for flow refinement.

and denoted the resulting set by $H_t^0 = \{h_{t,n}\}_{n=1}^N$. These point features are then mapped into a compact latent slot representation by aggregating informative sparse point evidence

$$Z_t = \Psi_{\text{slot}}(H_t^0) \in \mathbb{R}^{M \times D}, \tag{6}$$

where M is the number of latent slots and D is the latent dimension. The key role of $\Psi_{\text{slot}}(\cdot)$ is therefore not merely feature extraction, but the conversion of unordered sparse observations into a structured pose-oriented latent space.

Latent Temporal Aggregation. Given the frame-wise latent sequence $\{Z_t\}_{t=1}^T$, we further aggregated temporal context directly in the latent space. For each latent slot m, we collected its trajectory across time,

$$\tilde{Z}^{(m)} = \{z_{t,m}\}_{t=1}^T \in \mathbb{R}^{T \times D}, \tag{7}$$

and refined it by a temporal aggregation operator

$$\hat{Z}^{(m)} = \Psi_{\text{temp}}\left(\tilde{Z}^{(m)}\right). \tag{8}$$

The refined slots together form the temporally informed latent sequence

$$\hat{Z} = \{\hat{Z}_t\}_{t=1}^T, \qquad \hat{Z}_t \in \mathbb{R}^{M \times D}. \tag{9}$$

In Fig. 2, the slots with the same color are linked across frames, emphasizing that temporal interaction is performed slot-wise after latent compression.

Kinematic Prior Construction. We parameterized the coarse skeleton in terms of bone directions and lengths rather than joint coordinates. Specifically, given the temporally aggregated latent representation $\hat{Z}$, a shared kinematic decoder predicts, for each frame t and each non-root joint j, a unit direction $\mathbf{d}_{t,j}$ and a positive length $\ell_{t,j}$. The corresponding bone offset is defined as

$$\mathbf{o}_{t,j} = \ell_{t,j}\mathbf{d}_{t,j}. \tag{10}$$

The coarse skeleton is then recovered by forward kinematics along the kinematic tree:

$$\mathbf{x}^c_{t,0} = \mathbf{0}, \qquad \mathbf{x}^c_{t,j} = \mathbf{x}^c_{t,p(j)} + \mathbf{o}_{t,j}, \tag{11}$$

where $p(j)$ denotes the parent of joint j. This decoding process corresponds to the KPC block in Fig. 2, where the aggregated latent representation is transformed into a structured skeleton through kinematic construction rather than direct coordinate regression. Applying this construction to all frames yields the coarse skeleton sequence prior

$$X_c = \{\mathbf{x}^c_t\}_{t=1}^T \in \mathbb{R}^{T\times J\times 3}, \tag{12}$$

which serves as the structural initialization for the subsequent refinement stage.

3.3 Structure-Conditioned Flow Refinement

Given the coarse prior X_c, we construct the noise-perturbed source state X_0 during training, as defined in Sec. 3.1, to encourage robust local correction. During inference, refinement starts deterministically from the unperturbed coarse pose X_c.

For each joint j, we construct a query token by combining its current pose state, joint identity, structural context, localized Doppler cue, and flow time:

$$q_j = h_j^{\text{state}} + h_j^{\text{id}} + h_j^{\text{struct}} + h_j^{\text{doppler}} + h^{\text{time}}. \tag{13}$$

The structural embeddings encode anatomical roles and kinematic dependencies, while the Doppler embedding provides local motion evidence.

The Doppler component is obtained by distance-weighted aggregation around the current joint state. Let r_n denote the n-th radar point and d_n its Doppler value. We assign each point to joint j with a Gaussian weight:

$$w_{j,n} \propto \exp\left(-\frac{\|x_{t,j} - r_n^{xyz}\|^2}{2\sigma_j^2}\right), \qquad h_j^{\text{doppler}} = \frac{\sum_n w_{j,n}\,\phi_{\text{dop}}(d_n)}{\sum_n w_{j,n} + \varepsilon}, \tag{14}$$

where $x_{t,j}$ and r_n^{xyz} denote the current joint position and radar point location, respectively. The Gaussian width is $\sigma_j = r_j/2$, where r_j is the joint-specific neighbourhood radius. We use a base radius of 0.10 m, scaled to 0.06 m for core trunk joints and 0.15 m for end-effectors. Gaussian weighting emphasizes nearby points, while the radius constrains subsequent cross-attention.

To instantiate the velocity field $v_\theta(X_t, t, X_0, R)$, we use six pre-normalized Transformer decoder layers with an embedding dimension of 512, eight attention heads, and an FFN dimension of 2048. Radar point features are projected into a $BT \times N \times 512$ memory sequence for cross-attention, with joint tokens in Eq. (13) serving as queries. Joint self-attention follows the kinematic tree, allowing each joint to attend only to itself, its parent, and its direct children. Cross-attention is restricted to the joint-specific neighbourhood, with the nearest-point fallback when no radar point lies within the radius. The resulting $BT \times J \times 512$ features are mapped to three-dimensional joint velocities:

$$v_\theta(X_t, t, X_0, R) = \{v_j\}_{j=1}^J, \qquad v_j \in \mathbb{R}^3, \tag{15}$$

which update the pose during refinement.

During inference, we initialize $X^{(0)} = X_c$, with $X_0 = X_c$ fixed as the source condition, and integrate the learned velocity field using Euler updates. For N steps, let $\Delta t = 1/N$ and $t_k = k\Delta t$. The pose is updated as

$$X^{(k+1)} = X^{(k)} + \Delta t\, v_\theta(X^{(k)}, t_k, X_c, R). \tag{16}$$

We use two Euler steps in the main experiments, with pose-dependent Doppler features recomputed at each step.

3.4 Training Objectives

The coarse stage is supervised by an MSE loss on the predicted skeleton prior:

$$\mathcal{L}_{\text{coarse}} = \text{MSE}(X_c, X_{\text{gt}}). \tag{17}$$

This objective encourages the first stage to provide a stable global initialization, while leaving fine-grained correction to the refinement stage.

For the refinement stage, we use the perturbed source state X_0, target pose $X_1 = X_{\text{gt}}$, and intermediate state X_t defined in Sec. 3.1. The coarse prediction is detached during refinement training to prevent gradient propagation to the coarse stage. The refiner is trained using the weighted Flow Matching objective, together with geometric regularization.

The refinement objective combines flow matching and geometric regularization:

$$\mathcal{L}_{\text{refine}} = 100\mathcal{L}_{\text{FM}} + 100\mathcal{L}_{\text{len}} + \mathcal{L}_{\text{dir}}. \tag{18}$$

The bone-length and bone-direction terms regularize the refined skeleton geometry, encouraging the predicted motion to remain anatomically plausible during residual correction.

4 Experiments

4.1 Experimental Setup

Datasets. Since point clouds are the primary data modality in our work, we evaluated our method on three benchmark datasets: **MM-Fi** [36], **Reb3D** [37], and **mmBody** [38]. **MM-Fi** contains over 320K synchronized frames from 40 subjects performing 27 actions, captured in two environments and two facing angles (0° and 90°). **Reb3D** provides over 500K frames from 26 subjects covering daily and rehabilitation actions within two lighting conditions. **mmBody** includes 100 motion sequences from 20 volunteers across 7 scenes.

Implementation Details. We implemented RadarFlowPose using PyTorch 2.5.1 with CUDA 12.1 and conducted all experiments on an NVIDIA H100 96GB GPU. All experiments were trained for **100 epochs** using the **AdamW** optimizer with an initial learning rate of 1×10^{-4} and weight decay of 1×10^{-2}. For the dataset split, we followed the original protocol of each dataset.

Compared Methods. We compare our method with several representative baselines. The TDR+STE framework follows a standard direct regression pipeline with decoupled temporal and spatial modeling [39, 40]. **PointTransformer** [22] and **P4DTransformer** [41] are point-based spatio-temporal transformer baselines, learning dynamic point representations from sequential point clouds and directly regress pose coordinates. **DiffPose** [26] formulates 3D pose estimation as a reverse diffusion process for more reliable prediction. **mmDiff** [29] is a conditional diffusion-based pose estimator for radar point clouds, designed to improve robustness to miss-detection and signal inconsistency.

All compared methods are evaluated using the same dataset splits, point-cloud preprocessing, radar samples, points per frame, and MPJPE/PA-MPJPE metrics, subject to architecture-specific skeleton definitions.

5 Results and Analysis

5.1 Baseline Experiments

Table 1 reports the results on mmBody under five challenging scenarios. RadarFlowPose achieves the lowest average PA-MPJPE of 55.23 mm and the best PA-MPJPE in all five scenarios, indicating more consistent recovery of articulated pose structure. Its MPJPE is competitive but not uniformly the lowest, suggesting that the refinement benefit is more pronounced in relative skeletal configuration than in absolute joint localization.

Table 2 reports quantitative results on MM-Fi under three evaluation settings. RadarFlowPose achieves the lowest PA-MPJPE under all three MM-Fi evaluation protocols. Under the Random

Table 1: Quantitative results on mmBody (mm).

Methods	Metric	Furnished	Rain	Smoke	Dark	Occlusion	Avg.
P4DTransformer [41]	MPJPE	79.34	93.83	99.82	89.41	82.95	89.07
	PA-MPJPE	65.22	75.52	79.23	59.14	60.47	67.91
DiffPose [26]	MPJPE	69.28	77.77	89.01	67.27	74.52	73.31
	PA-MPJPE	52.09	62.60	69.34	49.90	56.20	58.23
mmDiff [29]	MPJPE	71.22	79.31	85.02	67.76	70.58	74.78
	PA-MPJPE	52.04	63.07	67.64	48.36	55.76	57.37
RadarFlowPose*	MPJPE	69.13	79.07	88.12	69.25	80.74	77.26
	PA-MPJPE	**50.21**	**60.42**	**64.94**	**47.39**	**53.22**	**55.23**

* Our proposed model.

Table 2: Quantitative results on MM-Fi (mm).

Methods	Random		Cross-Subject		Cross-Environment	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
PointTransformer [22]	93.09	75.60	95.96	68.70	108.28	88.79
Diffpose [26]	73.44	56.83	70.31	54.12	86.35	66.87
mmDiff [29]	**65.42**	57.49	69.34	59.23	82.63	64.43
RadarFlowPose*	66.09	**50.47**	**63.49**	**51.59**	**73.14**	**56.95**

Table 3: Quantitative results on Reb3D (mm).

Methods	Random		Cross-Light	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
TDR+STE [39, 40]	66.34	56.23	66.65	56.69
PointTransformer [22]	71.44	60.67	72.01	59.83
mmDiff [29]	67.28	55.05	67.01	57.41
RadarFlowPose*	**64.29**	**53.97**	**65.31**	**55.56**

* For Reb3D, mmDiff is evaluated using its supported 17-joint skeleton, whereas RadarFlowPose uses the complete 27-joint skeleton.

split, it reduces the error to 50.47, outperforming PointTransformer, DiffPose, and mmDiff by 33.2%, 11.2%, and 12.2%. RadarFlowPose also shows stronger robustness to environmental changes: from Random to Cross-Environment, its PA-MPJPE and MPJPE increase by only 6.48 mm and 7.05 mm, compared with 6.94/17.21 mm for mmDiff, 10.04/12.91 mm for DiffPose, and 13.19/15.19 mm for PointTransformer.

Table 3 presents the results on Reb3D under the Random and Cross-Light protocols. Across both protocols, RadarFlowPose achieves the lowest MPJPE and PA-MPJPE among all compared methods, showing consistent effectiveness under both standard and lighting-shift settings. In the Random split, it obtains a PA-MPJPE of 53.97, yielding relative improvements of 4.0% over TDR+STE, 11.1% over PointTransformer and 2.0% over mmDiff.

5.2 Why is RadarFlowPose effective?

The results show that RadarFlowPose mainly improves articulated body structure rather than only absolute joint positions, as its gains are more consistent on PA-MPJPE than on MPJPE. This indicates that the coarse prior provides a stable body-level initialization, while the flow refiner corrects residual local ambiguities around it. On MM-Fi, RadarFlowPose has the smallest degradation from Random to Cross-Environment, suggesting stronger robustness to environment-induced radar variations. This can be attributed to LTA, which models temporal motion patterns beyond individual noisy frames, KPC, which provides a kinematically structured prior, and the refiner, which uses localized Doppler cues to focus on motion-relevant residuals. On Reb3D, the diverse daily and rehabilitation motions leave more residual ambiguity after coarse prediction, making structure-conditioned flow refinement consistently beneficial across both protocols. On mmBody, many motions have smaller amplitudes and are already well handled by the coarse stage, so the refinement gain is more limited, although RadarFlowPose still achieves the best average PA-MPJPE.

Table 4: Progressive ablation of the proposed framework on three datasets (mm). Lower is better.

Methods	mmBody		MM-Fi		ReB3D	
	MPJPE↓	PA-MPJPE↓	MPJPE↓	PA-MPJPE↓	MPJPE↓	PA-MPJPE↓
SRL⋆	92.59	68.64	62.65	59.27	75.58	64.55
SRL + LTA⋆	75.16	56.42	65.44	55.50	66.24	56.26
SRL + LTA + KPC⋆	**69.01**	59.54	**62.26**	52.27	64.91	54.76
Stage-1 + Direct-RR⋆	86.29	65.71	69.86	54.10	65.31	56.21
RadarFlowPose⋆	74.27	**55.71**	66.09	**50.47**	**64.29**	**53.97**

⋆ Our proposed modules or model.

Table 5: Ablation of different Doppler aggregation strategies on three datasets (mm).

Doppler mode	mmBody		MM-Fi		ReB3D	
	MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓
None	85.07	62.41	69.63	50.99	67.18	54.30
Global	82.78	64.10	67.89	52.63	67.39	57.61
Localized⋆	**74.27**	**55.71**	**66.09**	**50.47**	**64.29**	**53.97**

⋆ Our proposed localized Doppler aggregation strategy.

Table 6: Sensitivity analysis of the neighbourhood radius on mmBody (mm).

Radius r (m)	MPJPE ↓	PA-MPJPE ↓
0.025	77.32	57.56
0.050	79.63	59.03
0.100	**74.37**	**55.71**
0.150	80.78	60.10
0.200	76.15	56.30

The Gaussian weighting width is set to $\sigma = r/2$. The selected radius is $r = 0.10$ m.

5.3 Ablation Study

5.3.1 Contribution of Coarse Pose Components

Table 4 evaluates the contributions of SRL, LTA, and KPC across the three datasets. LTA provides substantial improvements on mmBody and ReB3D, highlighting the benefit of temporal aggregation under sparse radar observations, although its effect on MM-Fi MPJPE is less consistent. Adding KPC further improves both metrics on MM-Fi and ReB3D. On mmBody, KPC reduces MPJPE from 75.16 to 69.01 mm, while PA-MPJPE increases from 56.42 to 59.54 mm. These results indicate that temporal aggregation and kinematic construction affect absolute localization and structural alignment differently, while together providing a structured Stage-1 initialization for subsequent refinement.

5.3.2 Ablation Study on Doppler Aggregation

We evaluate the effect of Doppler aggregation using three controlled variants: *No Doppler*, *Global Doppler*, and the proposed *Localized Doppler*. All other experimental settings are kept unchanged. In the No Doppler variant, the Doppler token is removed. Global Doppler aggregates Doppler features over the entire radar frame and shares them across all joints, whereas Localized Doppler aggregates Doppler evidence separately around each joint.

As shown in Table 5, Localized Doppler achieves the best MPJPE and PA-MPJPE on all three datasets. Global aggregation does not consistently outperform the No Doppler variant, indicating that frame-level aggregation may introduce motion cues unrelated to individual joints. In contrast, localized aggregation associates Doppler evidence with the current joint locations and provides more relevant joint-specific motion information.

We further examine the sensitivity to the neighbourhood radius r on mmBody. The Gaussian width for Doppler aggregation is set to $\sigma = r/2$, while r also determines the spatial neighbourhood used for cross-attention.

Table 7: Quantitative comparison of initialization strategies across different datasets. Values are Mean ± Std. Bold indicates the best performance (mm).

Method	Metric	mmBody ↓	MM-Fi ↓	Reb3D ↓
Random	MPJPE	77.59 ± 0.26	68.45 ± 0.45	65.22 ± 0.30
	PA-MPJPE	57.64 ± 0.19	52.90 ± 0.26	54.25 ± 0.54
Coarse	MPJPE	**74.37** ± 0.33	**66.09** ± 0.36	**64.29** ± 0.23
	PA-MPJPE	**55.71** ± 0.21	**50.47** ± 0.17	**53.97** ± 0.35

Table 6 shows that the neighbourhood radius affects the quality of the joint-specific Doppler evidence. We use $r = 0.10$ m, which is sufficiently large to include useful local radar returns while limiting interference from adjacent body regions. This setting achieves the best performance among the tested values, with the corresponding Gaussian width set to $\sigma = 0.05$ m.

5.3.3 Contribution of Flow Matching

To isolate the contribution of Flow Matching, we construct a controlled baseline termed One-step Direct Residual Refinement (Direct-RR). It uses the same coarse-pose estimator, radar conditioning, localized Doppler representation, and refinement backbone as RadarFlowPose, but predicts the full pose residual in a single deterministic update. As shown in Table 4, Direct-RR underperforms RadarFlowPose across all three datasets and tends to over-correct joints that are already reasonably estimated by the coarse stage.

We further compare coarse-pose initialization with random initialization. As shown in Table 7, coarse initialization consistently performs better on all three datasets. Starting from the coarse pose places the refinement trajectory near a plausible solution, allowing Flow Matching to focus on the remaining local errors rather than reconstructing the global pose from noise.

These results indicate that Flow Matching is most effective as a continuous, state-dependent refinement process built upon the coarse estimate, rather than as a pose generator from random noise.

5.4 Analysis on Extremity Structure and Motion

Figure 3 presents that the Coarse Pose Generator already provides a strong improvement over P4DTransformer, reducing the mean bone-length error from 29.66 to 14.87 mm and the mean joint-velocity error from 37.86 to 21.33. On top of this, RadarFlowPose further lowers the mean joint-velocity error to 18.70, with the largest gains on highly dynamic joints such as J20 and J21, where the velocity error decreases from 75.93 to 70.14 and from 81.87 to 76.38 compared with Coarse. In contrast, its effect on bone-length error is less uniform. Although it improves 11 out of 21 bones, with representative gains on B1 (16.29 to 13.77) and B5 (10.31 to 7.97), the overall mean bone-length error slightly increases from 14.87 to 16.35 due to larger deviations on several bones, particularly B20 (23.15 to 40.10), B16 (20.75 to 30.51), and B19 (19.58 to 25.74). These suggests that the coarse model mainly establishes strong structural consistency, while RadarFlowPose emphasizes continuous state refinement to better match the observed motion, especially on challenging extremity joints. We visualize four motion categories on Reb3D in Fig. 4. RadarFlowPose generally produces poses that are better aligned with the ground truth and better preserve plausible skeletal geometry. In contrast, PointTransformer occasionally shows less effective, where the estimated current pose is biased by previous frames. Hand joints are the most challenging due to their fine-grained structure and rapid motion, whereas relatively rigid parts such as the legs and chest typically have lower errors.

6 Conclusion

We propose RadarFlowPose, a framework for radar-based 3D pose estimation. The coarse stage provides a plausible global pose prior from sparse radar observations, while the refinement stage corrects the remaining local errors, particularly on ambiguous and fast-moving extremity joints. Extensive experiments across different datasets and settings show that this refinement strategy is effective, especially for motion-sensitive correction and ambiguous extremity joints. **Limitations.**

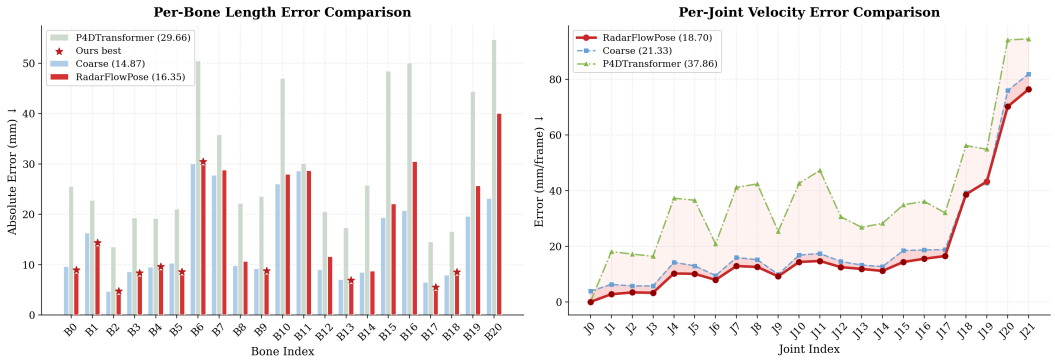


Figure 3: Per-bone length and per-joint velocity errors on mmBody (mm).

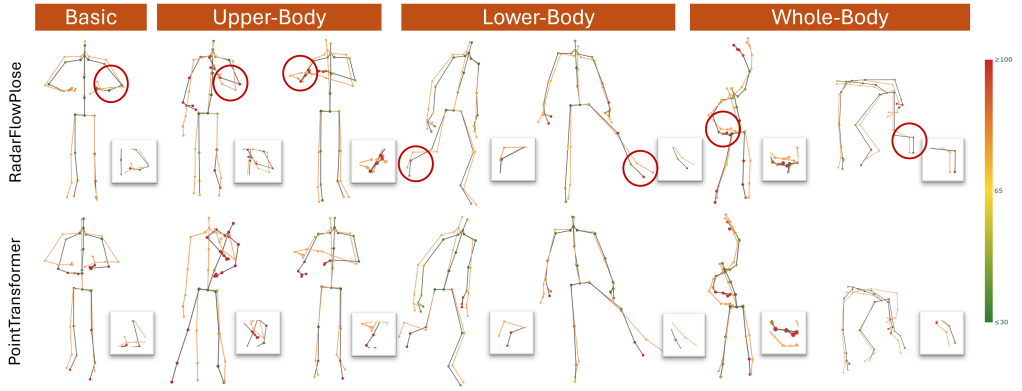


Figure 4: Qualitative comparison across body-part motion categories on Reb3D. GT is shown in orange and predictions in gray. Joint colors indicate error magnitude: red denotes errors above 100 mm and green denotes errors below 30 mm.

The refinement stage improves articulated structure and motion consistency more consistently than absolute joint localization or bone-length accuracy; joints that are already well estimated by the coarse stage can occasionally be over-corrected. Performance also degrades under severe localized radar occlusion, and the current evaluation is limited to the radar configurations and single-person settings represented by the three benchmarks. Future work will investigate rotation-based pose representations, lighter refinement models, multi-person sensing, and longer-term real-world deployment.

References

[1] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017.

[2] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ACM computing surveys*, 56(1):1–37, 2023.

[3] Elif Dogu, Janelle M. Jones, Akram Alomainy, and Khalid Z. Rajab. Millimetre-wave radar for remote health monitoring: Analysing digital biomarkers for real-time insights. *IEEE Internet of Things Journal*, pages 1–1, 2026. doi: 10.1109/JIOT.2026.3680222.

[4] Kailu Guo, Elif Dogu, and Khalid Z Rajab. Automated segmentation of the timed-up-and-go test using millimetre-wave radar sensors for mobility assessment. In *2025 6th International Conference on Computing, Networks and Internet of Things (CNIOT)*, pages 1–6. IEEE, 2025.

[5] Elif Dogu, José A Paredes, Akram Alomainy, Janelle M Jones, and Khalid Z Rajab. Reimagining falls prevention with insights from systems mapping on the use of millimetre-wave radar for remote health monitoring. *Scientific Reports*, 15(1):28842, 2025.

[6] Alan Bannon, Adrien Rapeaux, and Timothy G. Constandinou. Tiresias: A low-cost networked uwb radar system for in-home monitoring of dementia patients. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pages 7068–7072, 2021. doi: 10.1109/EMBC46164.2021.9630533.

[7] Jiahang Wang, Sheng Jin, Wentao Liu, Weizhong Liu, Chen Qian, and Ping Luo. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11855–11864, 2021.

[8] Sizhe An, Yin Li, and Umit Ogras. mri: Multi-modal 3d human pose estimation dataset using mmwave, rgb-d, and inertial sensors. *Advances in neural information processing systems*, 35: 27414–27426, 2022.

[9] Elif Dogu, Jose Paredes, Akram Alomainy, Janelle Jones, and Khalid Rajab. A falls risk screening tool based on millimetre-wave radar. In *Proceedings of the 10th International Conference on Information and Communication Technologies for Ageing Well and e-Health - ICT4AWE*, pages 161–168. INSTICC, SciTePress, 2024. ISBN 978-989-758-700-9. doi: 10.5220/0012618900003699.

[10] Akash Deep Singh, Yunhao Ba, Ankur Sarker, Howard Zhang, Achuta Kadambi, Stefano Soatto, Mani Srivastava, and Alex Wong. Depth estimation from camera image and mmwave radar point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9275–9285, 2023.

[11] Jia Zhang, Rui Xi, Yuan He, Yimiao Sun, Xiuzhen Guo, Weiguo Wang, Xin Na, Yunhao Liu, Zhenguo Shi, and Tao Gu. A survey of mmwave-based human sensing: Technology, platforms and applications. *IEEE Communications Surveys & Tutorials*, 25(4):2052–2087, 2023.

[12] Khalid Z Rajab, Bang Wu, Peter Alizadeh, and Akram Alomainy. Multi-target tracking and activity classification with millimeter-wave radar. *Applied Physics Letters*, 119(3), 2021.

[13] Jaeho Choi, Soheil Hor, Shubo Yang, and Amin Arbabian. Mvdoppler-pose: Multi-modal multi-view mmwave sensing for long-distance self-occluded human walking pose estimation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27750–27759, 2025.

[14] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns. *IEEE Sensors Journal*, 20(17): 10032–10044, 2020. doi: 10.1109/JSEN.2020.2991741.

[15] Sizhe An and Umit Y. Ogras. Fast and scalable human pose estimation using mmwave point cloud. In *Proceedings of the 59th ACM/IEEE Design Automation Conference (DAC)*, pages 889–894. Association for Computing Machinery, 2022. doi: 10.1145/3489517.3530522.

[16] Shih-Po Lee, Niraj Prakash Kini, Wen-Hsiao Peng, Ching-Wen Ma, and Jenq-Neng Hwang. Hupr: A benchmark for human pose estimation using millimeter wave radar. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[17] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[18] Kirsten Petras, Sanne ten Oever, Christianne Jacobs, and Valerie Goffaux. Coarse-to-fine information integration in human vision. *NeuroImage*, 186:103–112, 2019. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2018.10.086. URL `https://www.sciencedirect.com/science/article/pii/S1053811918320706`.

[19] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[20] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38 (5):146:1–146:12, 2019. doi: 10.1145/3326362.

[21] Abdullah Al Masud, Xintong Shi, Mondher Bouazizi, and Tomoaki Ohtsuki. mmgat: Pose estimation by graph attention with mutual features from mmwave radar point cloud. In *IEEE International Conference on Communications (ICC)*, 2024. doi: 10.1109/ICC51166.2024.10622791.

[22] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16259–16268, 2021. doi: 10.1109/ICCV48922.2021.01595.

[23] Niraj Prakash Kini, Ruey-Horng Shiue, ryan chandra, Wen-Hsiao Peng, Ching-Wen Ma, and Jenq-Neng Hwang. Transhupr: Cross-view fusion transformer for human pose estimation using mmwave radar. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024*. BMVA, 2024. URL `https://papers.bmvc2024.org/0686.pdf`.

[24] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2837–2845, 2021.

[25] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*, 2021.

[26] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023.

[27] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14761–14771, 2023.

[28] Lanxin Li, Che Liu, Wenming Yu, and Tie Jun Cui. Diff-hpe: Multi-hypothesis human pose estimation based on mmwave radar and diffusion framework. *IEEE Sensors Journal*, 2025.

[29] Junqiao Fan, Jianfei Yang, Yuecong Xu, and Lihua Xie. Diffusion model is a good pose estimator from 3d rf-vision. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024.

[30] Kailu Guo, Elif Dogu, Akram Alomainy, and Khalid Z Rajab. Diffradarpose: Diffusion-guided 3d human pose reconstruction from sparse radar data. In *2026 20th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5. IEEE, 2026.

[31] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[32] Md Shahriar Rahim Siddiqui, Moshe Eliasof, and Eldad Haber. Graph flow matching: Enhancing image generation with neighbor-aware flow fields. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 25463–25471, 2026.

[33] Wenzhe Ouyang, Qi Ye, Jinghua Wang, Zenglin Xu, and Jiming Chen. Rfmpose: Generative category-level object pose estimation via riemannian flow matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[34] Ti Wang, Xiaohang Yu, and Mackenzie Weygandt Mathis. Fmpose3d: monocular 3d pose estimation via flow matching. *arXiv preprint arXiv:2602.05755*, 2026.

[35] Shuting Hu, Siyang Cao, Nima Toosizadeh, Jennifer Barton, Melvin G. Hector, and Mindy J. Fain. mmPose-FK: A forward kinematics approach to dynamic skeletal pose estimation using mmWave radars. *IEEE Sensors Journal*, 24(5):6469–6481, March 2024. doi: 10.1109/JSEN.2023.3348199.

[36] Jianfei Yang, He Huang, Yunjiao Zhou, Xinyan Chen, Yuecong Xu, Shenghai Yuan, Han Zou, Chris Xiaoxuan Lu, and Lihua Xie. Mm-fi: Multi-modal non-intrusive 4d human dataset for versatile wireless sensing. *Advances in Neural Information Processing Systems*, 36:18756–18768, 2023.

[37] Kailu Guo, Elif Dogu, and Khalid Z Rajab. A spatial millimetre-wave radar point cloud dataset for health applications. *Scientific Data*, 13(1):1160, 2026.

[38] Anjun Chen, Xiangyu Wang, Shaohao Zhu, Yanxu Li, Jiming Chen, and Qi Ye. mmbody benchmark: 3d body reconstruction dataset and analysis for millimeter wave radar. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3501–3510, 2022.

[39] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[40] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European conference on computer vision*, pages 531–548. Springer, 2022.

[41] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.

A Model Details

A.1 Additional Architecture Details

Topology-constrained refinement. In the refinement decoder, we apply a skeleton topology mask to the self-attention among joint tokens. Each joint is allowed to attend to itself and its directly connected kinematic neighbors, i.e., its parent and child joints, while unrelated joints are masked out. The same mask is shared across all decoder layers. This provides a lightweight structural prior and encourages the refinement process to propagate information along the human kinematic tree.

Doppler usage in refinement. Doppler information is injected locally at the joint level rather than pooled globally. For each joint, nearby radar points are selected according to the distance between the current joint state and radar point locations. Their Doppler values are embedded and aggregated with Gaussian distance weights to form a joint-specific Doppler feature. This feature is added to the joint query in the refinement decoder. We also restrict radar cross-attention to local radar points using the same distance-based neighborhood, with a nearest-point fallback when a joint has no radar point within its local radius.

Important hyperparameters. We use 128 radar points per frame. The radar encoder uses a 256-dimensional feature space and compresses each frame into 64 radar tokens. The temporal adapter contains 2 Transformer layers with 4 attention heads. The refinement decoder uses a hidden dimension of 512, 6 Transformer decoder layers, 8 attention heads, and a feed-forward dimension of 2048. The base local radar radius is set to 0.1, with a scale of 0.6 for core joints and 1.5 for end-effectors. For refinement training, the end-effector loss weight is set to 3.0.

A.2 Loss Function

We provide the training objectives used for the two stages of our framework. The first stage is trained with a simple mean squared error (MSE) loss on the coarse skeleton prediction. Given the coarse prediction $X_c \in \mathbb{R}^{B\times T\times J\times 3}$ and the ground-truth pose sequence X_{gt}, we define

$$\mathcal{L}_{\mathrm{coarse}} = \frac{1}{3BTJ}\sum_{b=1}^{B}\sum_{t=1}^{T}\sum_{j=1}^{J}\|X_{c,b,t,j} - X_{\mathrm{gt},b,t,j}\|_2^2 . \tag{A.1}$$

This objective is intentionally simple: the coarse stage is not expected to recover fine-grained pose details, but rather to provide a stable structural anchor that captures the approximate body configuration and prevents the initialization from drifting too far from the target skeleton.

For the refinement stage, we train the model to predict a velocity field that moves an intermediate pose state toward the ground-truth pose. Let x_0 be the noise-perturbed coarse source defined in Sec. 3.1 and x_1 be the ground-truth pose. During training, the underlying coarse prediction is detached so that this loss only updates the refinement module.

We first sample a refinement time $\tau \in [0, 1]$ from $\mathrm{Beta}(2, 2)$. This distribution assigns higher probability to intermediate values of τ, allowing the model to observe more states in the middle of the refinement trajectory. We then construct an intermediate pose between the perturbed source and the target pose using a smooth interpolation:

$$\alpha(\tau) = 3\tau^2 - 2\tau^3, \qquad x_\tau = (1 - \alpha(\tau))x_0 + \alpha(\tau)x_1. \tag{A.2}$$

The corresponding target velocity is given by the derivative of this path:

$$v^\star(\tau) = \dot{\alpha}(\tau)(x_1 - x_0), \qquad \dot{\alpha}(\tau) = 6\tau - 6\tau^2. \tag{A.3}$$

The refiner predicts $v_\theta(x_\tau, \tau, x_0, R)$, and we supervise it with an MSE velocity loss:

$$\mathcal{L}_{\mathrm{vel}} = \frac{1}{3BTJ} \sum_{b,t,j} \lambda_t(\tau)\lambda_j \left\| v_{\theta,b,t,j} - v^\star_{b,t,j} \right\|_2^2. \tag{A.4}$$

Here, $\lambda_t(\tau) = 1 + 2\tau$ gives larger weight to later refinement states, and λ_j is a joint-specific weight, set to 3 for end-effectors and 1 for all other joints. This weighted velocity loss corresponds to $\mathcal{L}_{\mathrm{FM}}$ defined in Sec. 3.1.

To encourage anatomically plausible outputs, we further apply geometric regularization to a one-step endpoint proxy. For training efficiency, we approximate the endpoint obtained from the current intermediate state x_τ as

$$\hat{x}_1 = x_\tau + (1 - \tau)v_\theta. \tag{A.5}$$

This first-order Euler approximation is not the exact endpoint of the nonlinear smoothstep path. This proxy is not used as the final inference result, but provides an efficient way to regularize the local refinement direction during training. We then compute MSE losses on bone lengths and normalized bone directions:

$$\mathcal{L}_{\mathrm{len}} = \mathrm{MSE}\big(\ell(\hat{x}_1), \ell(x_1)\big), \qquad \mathcal{L}_{\mathrm{dir}} = \mathrm{MSE}\big(d(\hat{x}_1), d(x_1)\big), \tag{A.6}$$

where $\ell(\cdot)$ and $d(\cdot)$ denote the bone-length and normalized bone-direction operators, respectively. The final refinement loss is defined as

$$\mathcal{L}_{\mathrm{refine}} = 100\,\mathcal{L}_{\mathrm{vel}} + 100\,\mathcal{L}_{\mathrm{len}} + \mathcal{L}_{\mathrm{dir}}. \tag{A.7}$$

A.3 Dataset details

We evaluate RadarFlowPose on multiple radar-based human pose datasets with different skeleton definitions. mmBody uses 22 joints, MM-Fi uses 17 joints, and Reb3D uses 27 joints. For each dataset, we follow its official joint definition, evaluation protocol and split protocol. Figure 5 illustrates the skeleton topologies across the three different datasets.

RadarFlowPose is trained in a sequence-to-sequence manner. Each motion recording is divided into subsequences, with each subsequence containing approximately 200 frames depending on the duration of the original motion. Because radar point clouds are sparse, we stack three adjacent radar frames and append a temporal index to each point to distinguish its source frame. The radar input at each timestamp is padded or sampled to 128 points, where each point contains 6 channels, including spatial coordinates, Doppler-related measurements, signal strength, and the temporal index.

B Additional Experimental Results

B.1 More Joint and Bone Results on Other Datasets

Figure 6 shows the per-bone length error and per-joint velocity error on MM-Fi. Overall, RadarFlowPose achieves the lowest average bone error among the three methods, with 16.99 mm compared to

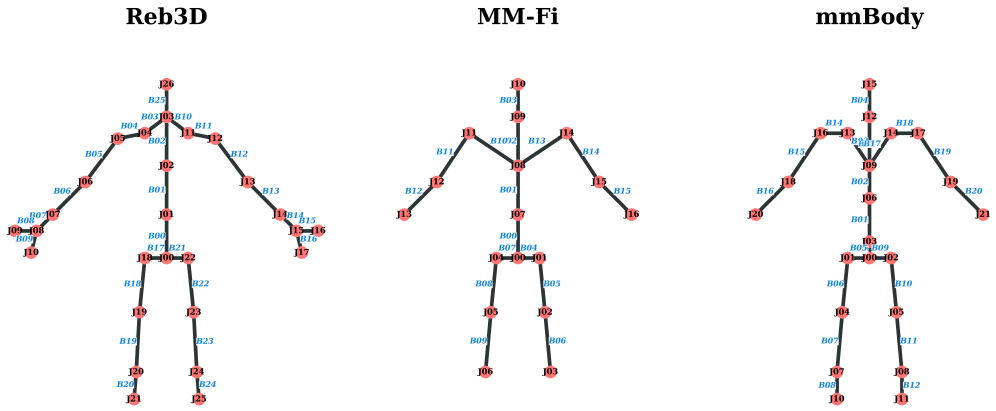


Figure 5: skeleton on three datasets

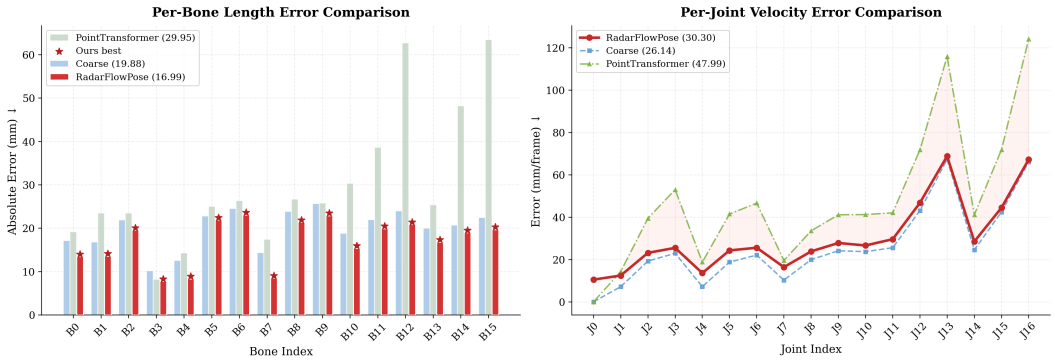


Figure 6: Per-bone length and per-joint velocity error comparison on MM-Fi

19.88 mm for the Coarse Pose Generator and 29.95 mm for PointTransformer. For joint velocity, RadarFlowPose obtains 30.30 mm, which is much lower than PointTransformer (47.99 mm), but slightly higher than the Coarse Pose Generator (26.14 mm).

Figure 7 reports the same comparison on Reb3D. RadarFlowPose also improves the average bone error, reducing it to 11.73 mm compared with 13.06 mm for the Coarse Pose Generator and 15.85 mm for PointTransformer. For velocity error, RadarFlowPose achieves the lowest average error, with 21.02 mm compared with 23.39 mm for the Coarse Pose Generator and 32.59 mm for PointTransformer.

Beyond the average results, the per-joint and per-bone distributions show that the advantage of RadarFlowPose is especially visible around kinematic boundary regions. End-effectors usually correspond to hands, feet, or other distal joints, which have larger motion ranges and are more affected by error accumulation along the skeleton. In the bone-error plots, RadarFlowPose reduces several high-error distal bones, such as the right-side high-index bones on MM-Fi and B15, B20, and B21 on Reb3D. In the velocity plots, the largest peaks also appear around distal joints. RadarFlowPose consistently suppresses these peaks on Reb3D and remains much lower than PointTransformer on MM-Fi, although the Coarse Pose Generator is slightly better in MM-Fi velocity.

Overall, these observations indicate that RadarFlowPose improves distal-joint structural consistency more consistently than temporal velocity accuracy, while still showing clear robustness over PointTransformer at high-motion end-effectors.

B.2 More Visualization

Figure 8 and Figure 9 provide qualitative comparisons across different motion categories, including basic, upper-body, lower-body, and whole-body motions. Across diverse motion patterns, RadarFlow-

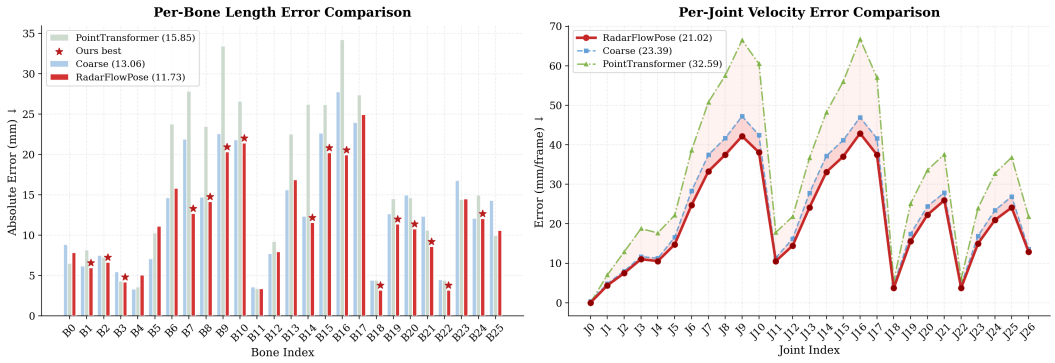


Figure 7: Per-bone length and per-joint velocity error comparison on Reb3D

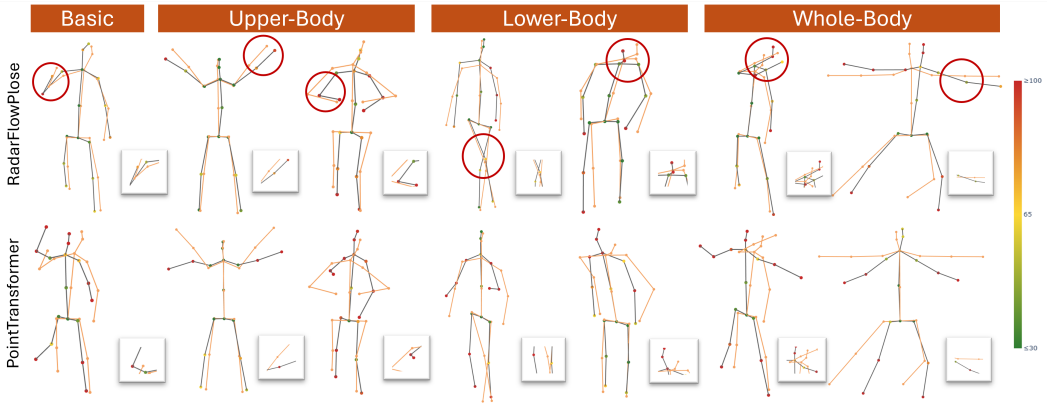


Figure 8: Qualitative Comparison on Different Body-Part Motion Categories in MM-Fi. GT is shown in orange and predictions in gray. Joint colors indicate error magnitude: red denotes errors above 100 mm, and green denotes errors below 30 mm. The top and bottom rows correspond to RadarFlowPose and PointTransformer.

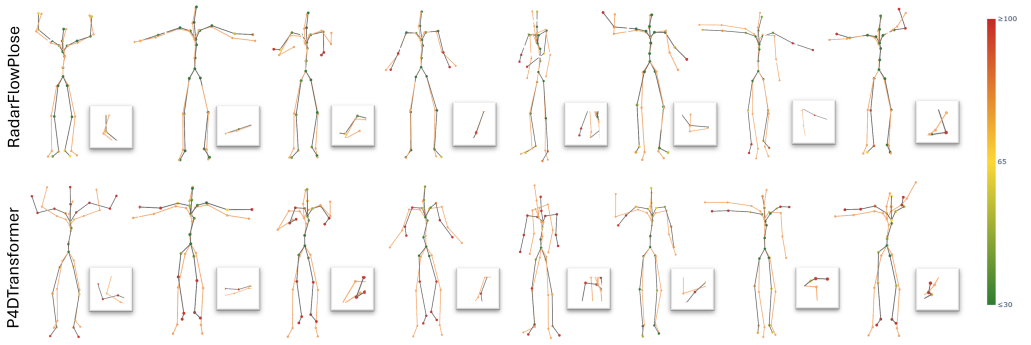


Figure 9: Qualitative Comparison on Different Body-Part Motion Categories in mmBody. GT is shown in orange and predictions in gray. Joint colors indicate error magnitude: red denotes errors above 100 mm, and green denotes errors below 30 mm. The top and bottom rows correspond to RadarFlowPose and P4DTransformer.

Pose generally produces poses that are better aligned with the ground truth and maintains more plausible skeletal geometry.

One noticeable observation is that PointTransformer and P4DTransformer sometimes fail to preserve basic human-body geometry. For example, they may produce over-shortened arms and distorted limb proportions. These errors suggest that directly regressing poses from sparse radar points can struggle to maintain kinematic consistency when the radar observation is ambiguous. In comparison, RadarFlowPose shows smaller local deviations in the highlighted regions and better preserves the continuity and proportion of limbs.

Nevertheless, some errors still remain under challenging poses. When the human body undergoes large orientation changes, or when some body parts are partially occluded or weakly observed by radar, RadarFlowPose may produce less accurate local details. However, even in these difficult cases, it generally preserves the overall body configuration and maintains a plausible global pose structure. The remaining errors are mainly concentrated on fine-grained limb details, such as hands, arms, and legs, rather than causing severe pose collapse.

B.3 Effect of Integration Steps

We evaluate the inference efficiency of multi-step refinement on 50-frame sequences using an NVIDIA A40 GPU with a batch size of 32, as reported in Table 8. Inference latency is measured using `time.perf_counter()` with CUDA synchronization before and after each timed inference, under evaluation mode without gradient computation. Timing includes model inference but excludes data loading, host-to-device transfers, and metric computation. Average sequence latency, per-frame latency, and FPS are calculated from the total inference time and the corresponding numbers of sequences and frames.

As expected, inference time increases approximately linearly with the number of Euler steps, each requiring one evaluation of the refinement network. Increasing the steps from 10 to 20 nearly doubles the sequence latency from 56.59 ms to 112.13 ms. Even at 20 steps, the model achieves 2.24 ms per frame and 445.91 FPS.

Fig. 10 further examines the effect of integration steps on PA-MPJPE. Euler integration approximates the continuous refinement trajectory through discrete velocity updates, with more steps providing finer discretization at higher computational cost. Notably, one-step Euler evaluates the velocity field at $t = 0$, where the smoothstep target velocity vanishes. However, since the exact endpoints are not explicitly supervised during training, the learned velocity need not be zero.

As shown in Fig. 10(a), RadarFlowPose is relatively insensitive to this hyperparameter. Across 1 to 20 steps, PA-MPJPE remains within a narrow range on all three datasets: approximately 53.8–55.0 mm on Reb3D, 50.5–51.0 mm on MM-Fi, and 55.8–56.7 mm on mmBody. This stability suggests that the coarse pose already places the initial state close to a plausible solution. As a result, the learned flow only needs to perform a local residual correction, and even a coarse numerical integration is sufficient to follow the correction direction. Increasing the number of steps only refines the discretization of this short trajectory, so the accuracy changes marginally.

In contrast, Fig. 10(b) shows that the random coarse source is much more sensitive to the number of steps. The error drops sharply from step 1 to the first few steps, indicating that the model initially benefits from iterative refinement. However, after around 3–5 steps, the error starts to increase on MM-Fi and mmBody, reaching around 66–67 mm at 20 steps. This suggests that when the initial state is weakly anchored, longer integration can accumulate trajectory errors or drift away from the desired pose manifold.

Overall, these results indicate that FM is most effective in our framework when used as a continuous, condition-dependent refinement mechanism around a reliable coarse estimate. The integration steps do not need to be large, because the refinement trajectory is short and well anchored. This allows us to select a small number of steps as a practical operating point that balances reconstruction accuracy and inference efficiency.

Efficiency comparison with mmDiff. We additionally compare RadarFlowPose with mmDiff under the same hardware and input protocol. Both models are evaluated on an NVIDIA A40 GPU

Table 8: Runtime analysis under different refinement integration steps. We report inference time per sequence, inference time per frame, and throughput in FPS.

Refinement Steps	Time / Seq. (ms) ↓	Time / Frame (ms) ↓	FPS ↑
1	8.09	0.1619	6178.03
2	13.27	0.2653	3769.12
10	56.59	1.1318	883.54
15	84.16	1.6832	594.09
20	112.13	2.2426	445.91

Table 9: Efficiency comparison with mmDiff on four 297-frame MM-Fi sequences using an NVIDIA A40 GPU.

Method	Steps	Params (M)	GFLOPs	Time/Seq. (ms) ↓	FPS ↑	VRAM (GB)
mmDiff	1	25.867	6787.57	2033.56	146.05	0.9100
RadarFlowPose	1	28.720	2054.99	79.77	3723.16	2.7929
mmDiff	2	25.867	6796.13	2033.75	146.04	0.9100
RadarFlowPose	2	28.720	3938.61	145.95	2034.94	2.7931
mmDiff	20	25.867	6950.30	8706.83	34.11	0.9100
RadarFlowPose	20	28.720	37843.71	1345.75	220.70	2.7931

using the same four 297-frame MM-Fi sequences. Table 9 reports model size, computational cost, sequence-level latency, throughput, and peak GPU memory.

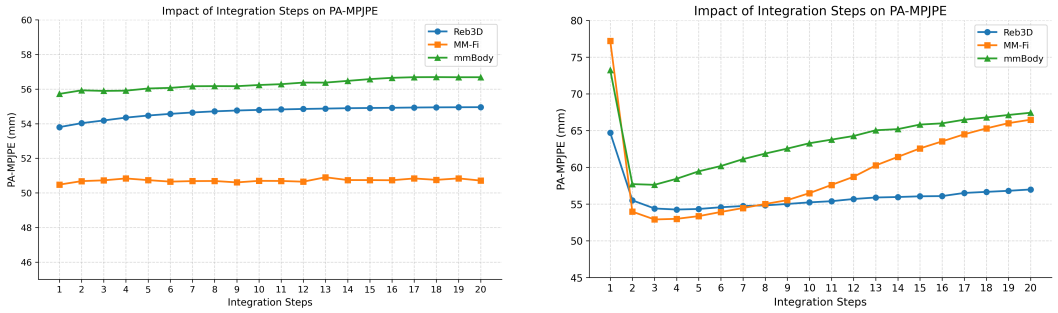
RadarFlowPose is approximately $25.5\times$ faster than mmDiff at one refinement step and $6.5\times$ faster at 20 steps in sequence-level latency. Its higher VRAM usage results from processing the sequence in parallel, whereas mmDiff operates frame by frame. The computational cost of RadarFlowPose increases more clearly with the number of Euler steps because the velocity-field network is reevaluated at each step, but sequence-level inference remains substantially faster under the matched benchmark.

B.4 Effect of Joint Occlusion

To evaluate the stability of RadarFlowPose under partial body occlusion, we design a localised radar occlusion experiment. This setting simulates cases where some body regions are blocked by obstacles or only weakly observed by radar. For a given joint, we define the occlusion region as a $0.5 \times 0.5 \times 0.5$ m^3 cube centered at the joint location, and randomly remove a fixed proportion of radar points inside the cube according to the drop ratio. This allows us to examine both how the model reacts to increasing occlusion severity and which body regions are more sensitive to missing local evidence.

The left part of Fig. 11 fixes the occlusion region around J0 and increases the drop ratio from 0.1 to 0.9. Both MPJPE and PA-MPJPE increase monotonically, with MPJPE rising from about 75.2 mm to 79.8 mm and PA-MPJPE from about 56.5 mm to 61.8 mm. The degradation is mild before a drop ratio of 0.7, while it becomes more visible at 0.8–0.9. This suggests that RadarFlowPose is robust to moderate local sparsity, but severe point removal still reduces useful local evidence for refinement.

The right part of Fig. 11 fixes the drop ratio at 0.8 and applies the same localised occlusion to each joint separately. The impact is joint-dependent: occluding J0–J4 gives the largest errors, around 75–76 mm MPJPE and 62–63 mm PA-MPJPE, whereas joints such as J5, J6, J8, J10, and J11 lead to lower errors around 72–73 mm MPJPE and 59–60 mm PA-MPJPE. This roughly 3–4 mm gap indicates that radar evidence around some structurally important regions is more critical for refinement. Overall, RadarFlowPose degrades smoothly under increasing occlusion and remains reasonably stable, while severe occlusion around key joints still limits fine-grained reconstruction accuracy.



(a) RadarFlowPose (b) Random coarse

Figure 10: Effect of refinement integration steps on PA-MPJPE.

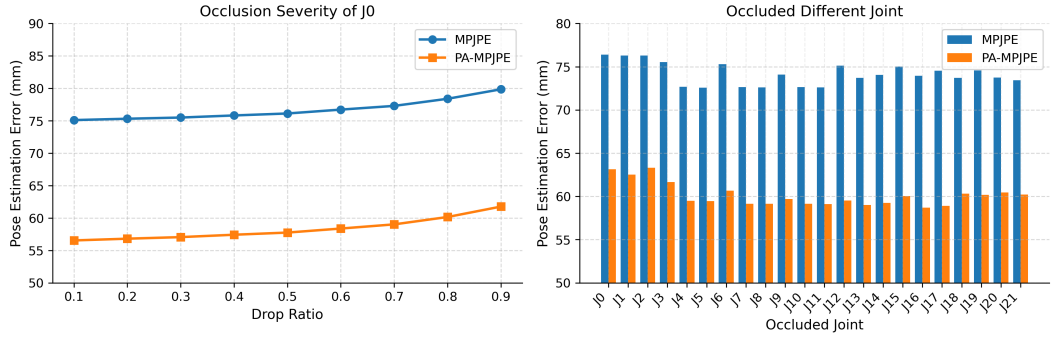


Figure 11: Sensitivity analysis under localised radar occlusion on mmBody. Left: MPJPE and PA-MPJPE when increasing the drop ratio around joint J0. Right: MPJPE and PA-MPJPE when applying localised occlusion to different joints with a fixed drop ratio of 0.8.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction accurately reflect the paper's contributions and are supported by the results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed limitation in Experiment session and Discussion session.

   Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the model architecture, training configuration, parameters, and evaluation protocol in sufficient detail to reproduce the main results, and the code will be released to further support reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our code will be uploaded as the supplementary material, and three datasets are open to get.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide these information in experiment setting and also more details in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report mean and standard deviation over multiple runs for the key experiments supporting the main claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: These information is offered in experiment setting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Radar-based pose estimation can support non-contact human sensing in healthcare, assistive systems, and human-computer interaction. Compared with camera-based sensing, radar does not capture identifiable visual appearance, which helps reduce privacy risks. However, since it can still infer human motion, responsible deployment should require consent, clear usage boundaries, and secure data handling.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk data or models that would require special safeguards for responsible release.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper properly credits the original sources of the existing datasets and codebases used, and their usage follows the corresponding licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide anonymized code for training and evaluation in the supplementary material, together with documentation of dependencies, data preprocessing, configuration files, and instructions for reproducing the reported results.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects requiring IRB approval or equivalent review.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not an important, original, or non-standard component of the core methodology in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.