# BRIDGING 2D OBSERVATIONS AND LEARNED 3D KINEMATICS: TARGET-SPACE ANCHORING AND REFINEMENT FOR 3D HUMAN POSE ESTIMATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Mamba holds great promise for modeling long-range spatiotemporal dependencies in monocular 3D human pose estimation (3D-HPE). However, existing Mamba-based methods generally propagate states over representations derived from 2D poses, where 3D skeletal and trajectory patterns are not explicitly represented during state modeling. To address this limitation, we propose KAR-Mamba, a kinematic anchoring and refinement framework that learns target-space kinematic representations and explicitly uses them to guide Mamba state propagation and updating. First, Codebook-Guided 3D Pose Anchoring leverages learned skeletal and trajectory codebooks to correct initial 3D relations toward representative human motion patterns and construct a kinematically informed pose anchor. Then, Skeletal-Trajectory Aligned Scanning organizes state propagation along skeletal chains and joint trajectories, with anchor-derived kinematic dependencies aligned to the corresponding transitions. Kinematic-Dependency-Modulated State Updating uses these relations to modulate input-dependent state parameters. Finally, the refined representations are decoded into anchor-relative offsets to predict the 3D pose. Experiments on Human3.6M and MPI-INF-3DHP show competitive accuracy and support the usefulness of incorporating learned 3D kinematic representations into refinement. Our code will be made publicly available.

## 1 INTRODUCTION

Monocular 3D human pose estimation (3D-HPE) aims to estimate continuous 3D pose sequences from 2D joint coordinates while preserving skeletal structures and motion dynamics Martinez et al. (2017); Pavllo et al. (2019); Zheng et al. (2021). Achieving reliable 3D-HPE, however, requires more than accurate joint coordinates; it also depends on the kinematic relations among 3D joints. These relations are reflected in the spatial skeletal geometry among joints and the temporal motion changes of individual joints across frames (Chen et al., 2022; Hossain & Little, 2018). Accordingly, they can be characterized by two types of dependencies: spatial skeletal dependencies, which describe local skeletal scales and relative 3D orientations, and temporal trajectory dependencies, which describe the magnitudes and directions of joint motion across frames. Modeling these dependencies can help estimate skeletally consistent and temporally coherent 3D poses Cai et al. (2019); Zhang et al. (2022); Tang et al. (2023); Zhu et al. (2023), thereby enabling broad applications such as quantitative motion analysis (Uhlrich et al., 2023), embodied intelligence, and immersive human-computer interaction (Mehta et al., 2020).

Despite substantial progress in joint-coordinate accuracy, monocular 3D-HPE still produces distorted skeletal structures and unstable motion trajectories, resulting in kinematically inconsistent 3D poses Zhao et al. (2023); Mehraban et al. (2024); Yu et al. (2023). These errors partly reflect the ambiguity of inferring 3D structure and motion from 2D observations: without depth, observed 2D relations alone are insufficient to uniquely determine the corresponding 3D kinematics Li & Lee (2019); Wehrbein et al. (2021); Li et al. (2022); Gong et al. (2023). This motivates incorporating learned 3D kinematic representations that serve as data-driven kinematic priors to provide structured guidance for 3D pose estimation. The challenge is further exacerbated by depth ambiguity, occlusion, and rapid motion, where 2D observations alone may not sufficiently constrain the 3D pose configuration Shan et al. (2023); Chen et al. (2023).



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.

Existing Mamba-based methods have explored different ways to organize pose tokens and incorporate structural or motion cues into state-space modeling Gu & Dao (2024). Scan-oriented approaches redesign joint ordering or traversal, while state-modeling approaches incorporate skeletal or motion information into features and state parameters. Our focus is to complement these observation-derived representations with a kinematically informed 3D pose anchor constructed from learned kinematic representations, and to use anchor-derived kinematic dependencies to guide both state propagation and parameter modulation.

To this end, we complement Mamba's pose features with estimated 3D kinematic relations. Our key insight is to first anchor 2D pose observations in a learned 3D pose representation space, where kinematic patterns can be explicitly modeled and exploited for subsequent Mamba refinement. Spatial transitions along skeletal chains can then be supplied with estimated 3D bone geometry, while temporal transitions along joint trajectories can be supplied with estimated 3D motion Sun et al. (2017); Fang et al. (2018); Zhao et al. (2019); Zou & Tang (2021); Shan et al. (2022).

Based on this idea, we propose KAR-Mamba, a kinematic-dependency-guided Mamba for 3D-HPE that follows an anchor-and-refine framework. Codebook-Guided 3D Pose Anchoring uses skeletal and trajectory codebooks learned from training 3D poses to adjust the initial relation estimates and construct a pose anchor. Spatial skeletal and temporal trajectory relations are then derived from this anchor. Skeletal-Trajectory Aligned Scanning orders tokens along skeletal chains and joint trajectories and pairs within-chain and temporal transitions with their anchor-derived relations. The traversal follows skeletal topology and temporal order, while the accompanying relations provide sample-dependent geometric cues. Kinematic-Dependency-Modulated State Updating further uses these estimated relations to modulate Mamba's input-dependent state parameters. Finally, the refined features are decoded into anchor-relative offsets and added to the anchor to obtain the final 3D pose. KAR-Mamba thus complements conventional 2D-to-3D lifting with refinement informed by learned 3D kinematic representations.

Empirically, we evaluate estimation accuracy on two benchmarks, performance under challenging observation and motion conditions, and spatial and temporal consistency. Ablations assess the contributions of pose anchoring, aligned scanning, and state modulation. Internal analyses examine whether the learned interactions better align with skeletal connectivity and whether modulation reflects differences in 3D relations. Together, these results support the use of learned 3D kinematic representations and anchor-derived relations to improve pose refinement in the evaluated settings.

## 2 Related Work

Relevant studies on monocular 3D-HPE can be broadly grouped into Mamba-based spatiotemporal modeling methods Huang et al. (2025); Zhang et al. (2025); Lu et al. (2025); Cui et al. (2026); Zheng et al. (2026) and kinematic modeling methods Peng et al. (2024); Du et al. (2024); Sun et al. (2024); Liu et al. (2025). The former focuses on improving spatiotemporal dependency modeling through scanning and state modeling, while the latter exploits skeletal, trajectory, or geometric information to enhance 3D pose estimation. Our framework derives estimated skeletal and trajectory relations from a prior-informed 3D pose anchor and uses them to guide both Mamba state propagation and parameter modulation. Detailed discussions are provided in **Appendix A.1**.

## 3 Method

We propose KAR-Mamba, which complements 2D-to-3D lifting with refinement of estimated 3D poses guided by learned kinematic representations. As shown in Fig. 1, given a 2D pose sequence $\mathrm{P}_{\mathrm{2D}} \in \mathbb{R}^{T\times J\times 2}$, where $T$ and $J$ denote the number of frames and joints, an ST-GCN first encodes it into pose features and predicts an initial 3D pose sequence. Our framework comprises three core designs. First, Codebook-Guided 3D Pose Anchoring adjusts the initial skeletal and trajectory relations using learned 3D kinematic codebooks and constructs a pose anchor $\mathrm{P}_{\mathrm{Anchor}} \in \mathbb{R}^{T\times J\times 3}$. From this anchor, spatial skeletal and temporal trajectory relations are derived as explicit kinematic guidance, which define the dependencies used for subsequent refinement. Then, Skeletal-Trajectory Aligned Scanning orders tokens along skeletal chains and joint trajectories and aligns the anchor-derived relations with the corresponding transitions. The scan order is determined by skeletal topology and chronology, while the relation features supply pose-specific 3D kinematic guidance. Meanwhile, Kinematic-Dependency-Modulated State Updating uses the same anchor-derived relations to modulate Mamba's input-dependent state parameters. Finally, the refined features are decoded into anchor-relative coordinate offsets and added to $\mathrm{P}_{\mathrm{Anchor}}$ to obtain the final 3D results.

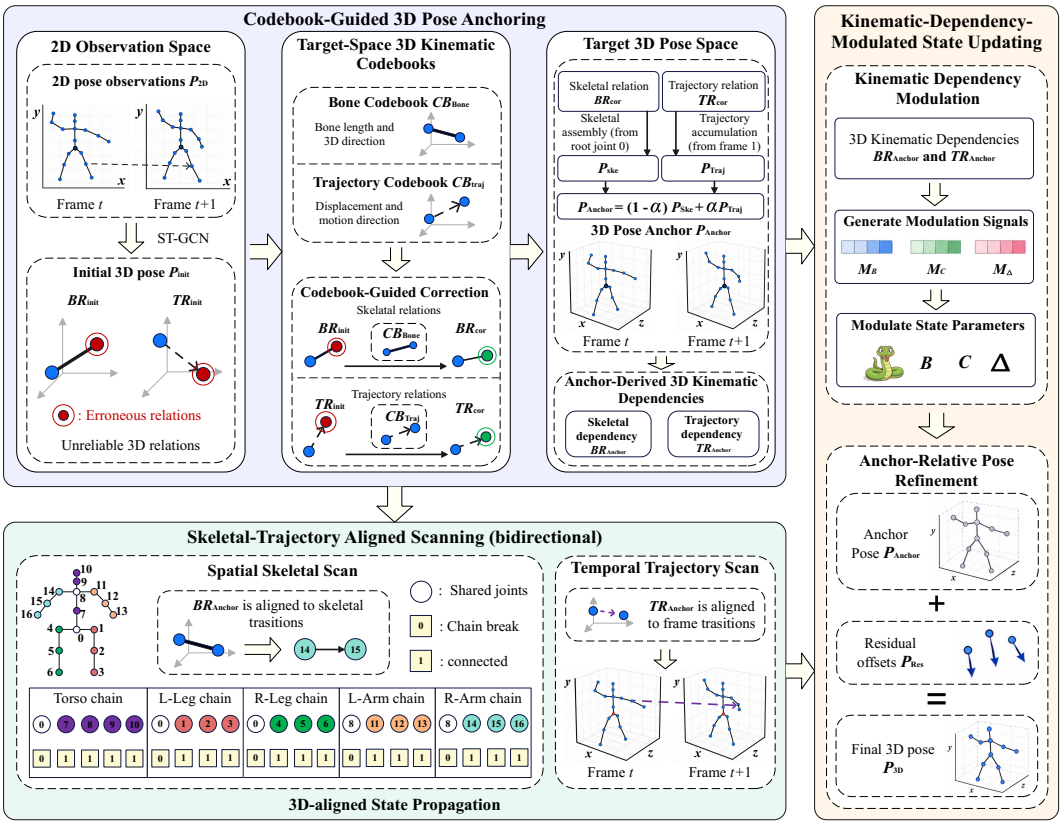



Figure 1: Our framework derives estimated skeletal and trajectory relations from a prior-informed 3D pose anchor to guide aligned scanning and Mamba state modulation for pose refinement.

### 3.1 Codebook-Guided 3D Pose Anchoring

Given $\mathrm{P}_{2\mathrm{D}}$, an ST-GCN predicts an initial 3D pose sequence $\mathrm{P}_{\mathrm{Init}} \in \mathbb{R}^{T\times J\times 3}$, providing an initial estimate in 3D coordinates. Its skeletal and trajectory relations may be inaccurate because of ambiguity in the 2D observations. We correct these relations using learned kinematic codebooks to construct a kinematically informed 3D pose anchor.

We first learn two kinematic codebooks from ground-truth 3D pose sequences during training. Similar to existing supervised 3D-HPE methods, these annotations are only used for learning representations and are unavailable during inference. For each bone and joint trajectory, we maintain $K = 64$ learnable 3D relation prototypes, forming a Bone Codebook $\mathrm{CB}_{\mathrm{Bone}} \in \mathbb{R}^{(J-1)\times K\times 3}$ and a Trajectory Codebook $\mathrm{CB}_{\mathrm{Traj}} \in \mathbb{R}^{J\times K\times 3}$, respectively. The prototypes are initialized by randomly sampling bone vectors and inter-frame joint displacement vectors from the first mini-batch of ground-truth 3D pose sequences, while preserving the corresponding bone and joint identities. From the ground-truth joint positions, we derive the skeletal relations $\mathrm{BR}_{\mathrm{GT}} \in \mathbb{R}^{T\times(J-1)\times 3}$ and trajectory relations $\mathrm{TR}_{\mathrm{GT}} \in \mathbb{R}^{(T-1)\times J\times 3}$. For each relation, the nearest prototype is selected from its corresponding bone- or joint-specific codebook:

$$\widehat{\mathrm{BR}}_{\mathrm{GT}} = \mathcal{Q}\left(\mathrm{BR}_{\mathrm{GT}}, \mathrm{CB}_{\mathrm{Bone}}\right), \qquad \widehat{\mathrm{TR}}_{\mathrm{GT}} = \mathcal{Q}\left(\mathrm{TR}_{\mathrm{GT}}, \mathrm{CB}_{\mathrm{Traj}}\right), \tag{1}$$

where $\mathcal{Q}(\cdot)$ denotes nearest-prototype assignment based on squared Euclidean distance, $\widehat{\mathrm{BR}}_{\mathrm{GT}} \in \mathbb{R}^{T\times(J-1)\times 3}$ and $\widehat{\mathrm{TR}}_{\mathrm{GT}} \in \mathbb{R}^{(T-1)\times J\times 3}$ denote the quantized skeletal and trajectory relations reconstructed by the selected prototypes. The codebooks are optimized by minimizing the error as:

$$\mathcal{L}_{\mathrm{CB}} = \left\|\mathrm{BR}_{\mathrm{GT}} - \widehat{\mathrm{BR}}_{\mathrm{GT}}\right\|_2^2 + \left\|\mathrm{TR}_{\mathrm{GT}} - \widehat{\mathrm{TR}}_{\mathrm{GT}}\right\|_2^2. \tag{2}$$

Through repeated prototype assignment and reconstruction, the prototypes associated with each bone capture recurring patterns of 3D skeletal relations, while those associated with each joint ap-

proximate 3D trajectory patterns in the training data. After training, the learned codebooks are frozen and used as learned kinematic representations.

After learning the kinematic codebooks, we use them to adjust the skeletal and trajectory relations of the initial 3D prediction toward learned patterns. Given $\mathrm{P}_{\mathrm{Init}}$, we extract the initial skeletal relations $\mathrm{BR}_{\mathrm{Init}} \in \mathbb{R}^{T\times(J-1)\times 3}$ and trajectory relations $\mathrm{TR}_{\mathrm{Init}} \in \mathbb{R}^{(T-1)\times J\times 3}$. The corresponding kinematic codebooks are then used to modify these relations through cross-attention-based residual adjustment:

$$\begin{aligned}\mathrm{BR}_{\mathrm{Cor}} &= \mathrm{BR}_{\mathrm{Init}} + \lambda_b \left(\mathrm{Attn}(\mathrm{BR}_{\mathrm{Init}}, \mathrm{CB}_{\mathrm{Bone}}) - \mathrm{BR}_{\mathrm{Init}}\right), \\ \mathrm{TR}_{\mathrm{Cor}} &= \mathrm{TR}_{\mathrm{Init}} + \lambda_t \left(\mathrm{Attn}(\mathrm{TR}_{\mathrm{Init}}, \mathrm{CB}_{\mathrm{Traj}}) - \mathrm{TR}_{\mathrm{Init}}\right),\end{aligned} \tag{3}$$

where $\lambda_b$ and $\lambda_t$ control the strengths of skeletal and trajectory correction, respectively. The attention is performed within the corresponding bone- or joint-specific prototype set, preserving the identity of each skeletal or trajectory relation. The resulting $\mathrm{BR}_{\mathrm{Cor}} \in \mathbb{R}^{T\times(J-1)\times 3}$ and $\mathrm{TR}_{\mathrm{Cor}} \in \mathbb{R}^{(T-1)\times J\times 3}$ combine the initial relation estimates with codebook-weighted patterns.

We then reconstruct the 3D pose anchor from the adjusted relations. As illustrated in Fig. 1, joint 0 serves as the root joint. The adjusted bone relations are assembled from this root along the human topology to obtain the skeletal reconstruction $\mathrm{P}_{\mathrm{Ske}} \in \mathbb{R}^{T\times J\times 3}$, while the adjusted trajectory relations are temporally accumulated from the first-frame pose of $\mathrm{P}_{\mathrm{Init}}$ to obtain the trajectory reconstruction $\mathrm{P}_{\mathrm{Traj}} \in \mathbb{R}^{T\times J\times 3}$. The final pose anchor is defined as:

$$\mathrm{P}_{\mathrm{Anchor}} = (1-\alpha)\mathrm{P}_{\mathrm{Ske}} + \alpha \mathrm{P}_{\mathrm{Traj}}, \tag{4}$$

where $\alpha \in [0, 1]$ balances spatial skeletal reconstruction and temporal trajectory evolution. The resulting $\mathrm{P}_{\mathrm{Anchor}} \in \mathbb{R}^{T\times J\times 3}$ provides an estimated geometric reference informed by the learned priors, from which spatial skeletal dependencies $\mathrm{BR}_{\mathrm{Anchor}} \in \mathbb{R}^{T\times(J-1)\times 3}$ and temporal trajectory dependencies $\mathrm{TR}_{\mathrm{Anchor}} \in \mathbb{R}^{(T-1)\times J\times 3}$ are further derived to align subsequent spatial-temporal scanning and guide Mamba state updating.

### 3.2 Skeletal-Trajectory Aligned Scanning

We organize spatial scanning along skeletal chains and temporal scanning along consecutive frames of the same joint. These traversal orders are fixed by anatomical topology and chronology; pose-specific guidance is supplied by aligning the anchor-derived 3D relations with the corresponding transitions. Alongside the initial 3D prediction $\mathrm{P}_{\mathrm{Init}}$, the ST-GCN encoder provides pose features, which are augmented with spatial and temporal identity embeddings to obtain $\mathrm{X} \in \mathbb{R}^{T\times J\times C}$. The spatial identity distinguishes joint locations, while the temporal identity indicates frame positions, preserving the spatiotemporal semantics of each token after reordering.

We then reorganize X into five local topological chains following the human topology, corresponding to the torso, two arms, and two legs, as shown in Fig. 1. Shared attachment joints are duplicated across chains while preserving their joint identities. The pose features are then reordered according to the resulting chain-wise scan order, yielding $\mathrm{X}_{\mathrm{Chain}} \in \mathbb{R}^{T\times L\times C}$ with $L = 21$ scan positions. The corresponding skeletal and trajectory dependencies are duplicated and reordered consistently with the chain-wise token layout, so that each valid scan transition is paired with its corresponding anchor-derived kinematic dependency. The skeletal dependencies $\mathrm{BR}_{\mathrm{Anchor}}$ describe the estimated 3D relation between each parent-child joint pair, jointly encoding its estimated bone length and direction. They therefore provide an explicit structural reference for within-chain spatial transitions.

The dependencies are then projected into relation features and fused with the corresponding child-joint tokens, supplying skeletal dependency cues along the fixed traversal. Concatenating the five chains also introduces inter-chain transitions that do not correspond to skeletal connections. We therefore introduce binary connection labels, where 0 denotes a chain start and 1 denotes a valid within-chain transition. These labels are mapped into learnable connection features $\mathrm{E} \in \mathbb{R}^{T\times L\times C}$ and fused with $\mathrm{X}_{\mathrm{Chain}}$. The resulting boundary cues let the model distinguish within-chain connections from inter-chain switches. For temporal scanning, $\mathrm{TR}_{\mathrm{Anchor}}$ encodes the estimated displacement magnitude and motion direction of each joint between adjacent frames. Its projected relation features are fused with the corresponding destination joints, providing motion estimates.

Finally, both the spatial skeletal and temporal trajectory paths are scanned bidirectionally, forming forward and backward propagation streams. For reverse scanning, the corresponding kinematic relations and connection indicators are reordered consistently with the reversed scan order, with the

relation directions adjusted accordingly. This design combines skeletal and temporal traversal with anchor-derived geometric estimates to inform state propagation over 2D-derived pose features.

### 3.3 KINEMATIC-DEPENDENCY-MODULATED STATE UPDATING

Kinematic-aligned scanning supplies relation-enriched tokens along skeletal and temporal paths. Beyond this feature-level guidance, we introduce Kinematic-Dependency-Modulated State Updating to directly condition Mamba's input-dependent state parameters on the anchor-derived relations. This explicitly conditions the parameters governing information writing, retention, and readout on anchor-derived 3D kinematic relations.

The same anchor-derived kinematic dependencies used for scanning are further employed to modulate state parameters, as shown in Fig. 1. Following the scan alignment described above, the corresponding skeletal or trajectory dependencies are mapped to parameter modulation signals as:

$$[\mathrm{M}_B, \mathrm{M}_C, \mathrm{M}_\Delta] = \begin{cases} \Phi(\mathrm{BR}_{\mathrm{Anchor}}), & \text{for spatial propagation,} \\ \Phi(\mathrm{TR}_{\mathrm{Anchor}}), & \text{for temporal propagation,} \end{cases} \tag{5}$$

where $\Phi(\cdot)$ denotes a lightweight nonlinear mapping. Spatially, $\mathrm{BR}_{\mathrm{Anchor}}$ provides the estimated 3D skeletal relation for each within-chain transition. Temporally, $\mathrm{TR}_{\mathrm{Anchor}}$ provides the estimated 3D motion relation for transitions between adjacent frames of the same joint. $\mathrm{M}_B$, $\mathrm{M}_C$, and $\mathrm{M}_\Delta$ denote the resulting modulation signals, which are applied to modulate state parameters as:

$$\widehat{\Theta} = \Theta \odot (1 + \sigma(\mathrm{M}_\Theta)), \qquad \Theta \in \{\mathbf{B}, \mathbf{C}, \mathbf{\Delta}\}, \tag{6}$$

where $\sigma(\cdot)$ denotes the Sigmoid function, and $\widehat{\Theta}$ denotes the modulated state parameter. The same anchor-derived kinematic dependencies therefore condition both the features used in propagation and the corresponding input-dependent state parameters.

After refinement through the stacked Mamba blocks, the bidirectional representations are restored to the original joint order, and duplicated shared-joint representations are averaged to recover the $J$-joint layout. The regression head then predicts anchor-relative coordinate offsets $\mathrm{P}_{\mathrm{Res}} \in \mathbb{R}^{T\times J\times 3}$. The final 3D pose is obtained as:

$$\mathrm{P}_{\mathrm{3D}} = \mathrm{P}_{\mathrm{Anchor}} + \mathrm{P}_{\mathrm{Res}}, \tag{7}$$

where $\mathrm{P}_{\mathrm{Anchor}}$ serves as the estimated 3D reference and $\mathrm{P}_{\mathrm{Res}}$ captures the residual correction. Thus, the model predicts a residual adjustment to a prior-informed 3D pose estimate. **Appendix A.2** specifies the complete training objective, including coordinate-, scale-, velocity-, and smoothness-based supervision for the refined pose, together with kinematic-relation and coordinate supervision for the intermediate 3D anchor. Overall, the framework augments 2D-to-3D lifting with learned kinematics and uses anchor-derived estimates in pose refinement, state propagation, and parameter modulation.

## 4 EXPERIMENTS

### 4.1 BENCHMARKS AND EVALUATION SETUP

Experiments are conducted on Human3.6M Ionescu et al. (2014) and MPI-INF-3DHP Mehta et al. (2017). The datasets, protocols, and experimental configurations are detailed in **Appendix A.3**. Sensitivity to codebook correction strength, anchor fusion initialization, and supervision weights is analyzed in **Appendix A.4**.

### 4.2 COMPARISON WITH STATE-OF-THE-ART METHODS

We compare KAR-Mamba with representative state-of-the-art methods from three perspectives. Specifically, we aim to answer the following questions: **(1) Does KAR-Mamba achieve competitive accuracy on the standard Human3.6M benchmark across different model scales? (2) Does KAR-Mamba remain effective on MPI-INF-3DHP under more diverse motions, viewpoints, and scene conditions? (3) Does KAR-Mamba reduce errors on challenging subsets with low-confidence 2D observations and intense motion?** These evaluations assess the effectiveness of KAR-Mamba under standard benchmarks, diverse motion and scene conditions, and challenging observation settings.

#### 4.2.1 DOES KAR-MAMBA PERFORM COMPETITIVELY ON HUMAN3.6M?

As shown in Table 1, KAR-Mamba consistently improves the corresponding PoseMamba baselines across different model scales. Compared with the recent SAMA, KAR-Mamba-S and KAR-Mamba-L achieve lower errors, while KAR-Mamba-B remains comparable, with the same P2 but a slightly higher P1$^\dagger$ and a 0.4 mm higher P1. Notably, KAR-Mamba-L also uses 29.8% fewer parameters and 49.3% fewer MACs than PoseMamba-X while achieving lower errors. These results demonstrate that KAR-Mamba provides consistent gains across model scales with competitive efficiency.

Table 1: Comparison on Human3.6M. $T$: input frames; CE: center-frame estimation; Params: M; MACs: G; MACs/F: M. P1: MPJPE; P2: P-MPJPE; P1$^\dagger$: P1 with ground-truth 2D poses.

| Variants | Method | $T$ | CE | Params | MACs | MACs/F | P1↓ | P2↓ | P1$^\dagger$ ↓ |
|---|---|---|---|---|---|---|---|---|---|
| Small | GLA-GCN (Yu et al., 2023) | 243 | ✓ | 1.3 | 1.5 | 6 | 44.4 | 34.8 | 21.0 |
| | HDFormer (Chen et al., 2023) | 96 | × | 3.7 | 0.6 | 6 | 42.6 | 33.1 | 21.6 |
| | PoseMamba-S (Huang et al., 2025) | 243 | × | 0.9 | 3.6 | 15 | 41.8 | 35.0 | 20.0 |
| | SAMA-S (Lu et al., 2025) | 243 | × | 1.1 | 3.9 | 16 | 40.6 | 34.0 | 20.2 |
| | **KAR-Mamba-S** | **243** | × | **1.1** | **3.8** | **15** | **39.5** | **33.1** | **19.8** |
| Base | STCFormer (Tang et al., 2023) | 243 | × | 4.7 | 19.6 | 80 | 41.0 | 32.0 | 21.3 |
| | PoseFormerV2 (Zhao et al., 2023) | 243 | ✓ | 14.4 | 4.8 | 20 | 45.2 | 35.6 | – |
| | PoseMamba-B (Huang et al., 2025) | 243 | × | 3.4 | 13.9 | 57 | 40.8 | 34.3 | 16.8 |
| | SAMA-B (Lu et al., 2025) | 243 | × | 3.3 | 11.7 | 48 | 37.7 | 32.0 | 13.6 |
| | SasMamba-L (Cui et al., 2026) | 243 | × | 4.1 | 8.6 | 35 | 39.8 | 33.6 | 21.0 |
| | **KAR-Mamba-B** | **243** | × | **4.0** | **13.8** | **56** | **38.1** | **32.0** | **13.8** |
| Large | MotionBERT (Zhu et al., 2023) | 243 | × | 42.3 | 174.8 | 719 | 39.2 | 32.9 | 17.8 |
| | MotionAGFormer-L (Mehraban et al., 2024) | 243 | × | 19.0 | 78.3 | 322 | 38.4 | 32.5 | 17.3 |
| | KTPFormer (Peng et al., 2024) | 243 | × | 35.2 | 76.1 | 313 | 40.1 | 31.9 | 19.0 |
| | TCPFormer (Liu et al., 2025) | 243 | ✓ | 35.1 | 109.2 | 449 | 37.9 | 31.7 | – |
| | PoseMagic (Zhang et al., 2025) | 243 | × | 14.4 | 20.3 | 84 | 37.5 | – | – |
| | PoseMamba-X (Huang et al., 2025) | 243 | × | 26.5 | 109.9 | 452 | 37.1 | 31.5 | 14.8 |
| | SAMA-L (Lu et al., 2025) | 243 | × | 17.3 | 53.2 | 219 | 36.9 | 31.3 | 11.9 |
| | **KAR-Mamba-L** | **243** | × | **18.6** | **55.7** | **229** | **36.8** | **31.2** | **11.7** |

#### 4.2.2 DOES KAR-MAMBA REMAIN EFFECTIVE BEYOND CONTROLLED INDOOR SCENES?

MPI-INF-3DHP contains more diverse motions, viewpoints, and scene conditions than Human3.6M. As shown in Table 2, KAR-Mamba achieves the best PCK and MPJPE among the compared methods while matching the best AUC. In particular, it reduces MPJPE by 2.1% over SAMA and 2.8% over PoseMamba. These results support the effectiveness of the proposed kinematic representation-guided refinement on this benchmark.

Table 2: Comparison on MPI-INF-3DHP.

| Method | $T$ | PCK↑ | AUC↑ | MPJPE↓ |
|---|---|---|---|---|
| GLA-GCN | 81 | 98.5 | 79.1 | 27.8 |
| STCFormer | 81 | 98.7 | 83.9 | 23.1 |
| DiffPose | 81 | 98.0 | 75.9 | 29.1 |
| MotionAGFormer | 81 | 98.2 | 85.3 | 16.2 |
| KTPFormer | 81 | 98.9 | 85.9 | 16.7 |
| TCPFormer | 81 | 99.0 | 87.7 | 15.0 |
| PoseMagic | 81 | 98.8 | 87.6 | 14.7 |
| PoseMamba | 81 | – | – | 14.5 |
| SAMA | 81 | 99.0 | 88.3 | 14.4 |
| **KAR-Mamba** | **81** | **99.1** | **88.3** | **14.1** |

#### 4.2.3 DOES KAR-MAMBA REDUCE ERRORS ON CHALLENGING SUBSETS?

Average performance on Human3.6M may obscure differences on difficult observations and motions. We therefore construct two challenging subsets with low-confidence 2D observations (∼10%) and high-motion intensity (∼25%). Detailed subset construction is provided in **Appendix A.5**. As shown in Table 3, KAR-Mamba-L achieves the lowest errors among the compared methods on both subsets. Compared with PoseMamba-X, it reduces MPJPE by 4.9% and 3.9% under low-confidence and high-motion conditions, respectively, exceeding its 0.8% overall improvement. The gains over SAMA-L similarly increase from 0.3% overall to 2.8% and 2.1% on the two challenging subsets. These larger gains support the usefulness of kinematic representation-guided anchoring and refinement on the evaluated difficult cases.

Table 3: Robustness on challenging Human3.6M subsets. Low Conf. and High Motion denote the lowest-confidence 10% and highest-motion 25% test samples, respectively. Results are P1 (mm); $^*$ denotes reproduced results.

| Method | Low Conf.↓ | High Motion↓ |
|---|---|---|
| DiffPose | 54.4 | 42.9 |
| PoseMagic | 56.2 | 43.5 |
| PoseMamba-X | 55.3 | 43.2 |
| SAMA-L$^*$ | 54.1 | 42.4 |
| **KAR-Mamba-L** | **52.6** | **41.5** |

### 4.3 ANALYSIS OF CODEBOOK-GUIDED 3D POSE ANCHORING

We analyze Codebook-Guided 3D Pose Anchoring from four complementary perspectives. Specifically, we aim to answer the following questions: **(1) Does codebook-guided adjustment reduce errors in the initial 3D kinematic estimates? (2) Do the learned codebooks represent held-out 3D kinematic relations? (3) How does codebook capacity affect the learned kinematic representation and pose estimation? (4) How should the learned codebook relations be incorporated to improve the initial kinematic estimates?** We investigate the first two questions in the following sections, while the analyses of codebook capacity and correction strategy are provided in **Appendix A.6** and **Appendix A.7**.

#### 4.3.1 Does Codebook Anchoring Improve Initial 3D Kinematic Estimates?

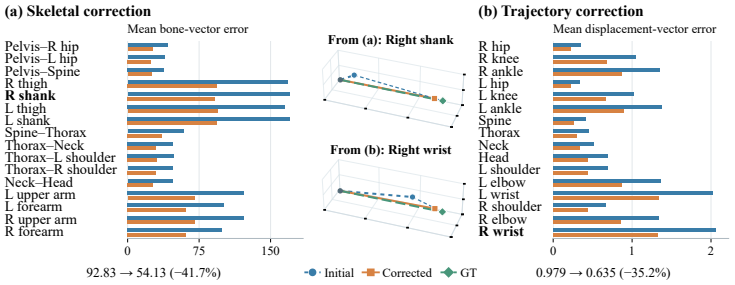



Figure 2: Codebook-guided adjustment of skeletal and trajectory relations. Case residuals are uniformly magnified for visibility.

To evaluate codebook-guided adjustment, we compare skeletal and trajectory relation errors before and after this step. As shown in Fig. 2, the adjusted relations exhibit lower errors relative to the corresponding ground-truth relations. The central panels visualize the skeletal and trajectory relations with the largest initial errors, illustrating the adjustment on these selected cases. These results support the use of learned kinematic prototypes to improve initial 3D relations for subsequent pose anchoring and refinement.

#### 4.3.2 Do the Learned Codebooks Capture Representative 3D Relations?

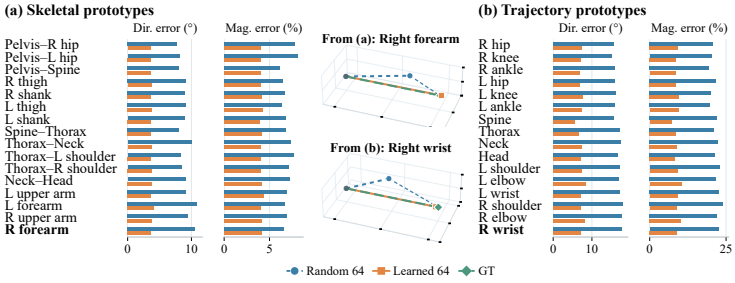



Figure 3: Quality of learned skeletal and trajectory prototypes on held-out 3D relations. Case residuals are uniformly magnified for visibility.

To evaluate the representation quality of the learned codebooks, we compare them with a Random-64 baseline constructed by randomly sampling 64 training relations for each corresponding bone or joint, without optimization. For each held-out ground-truth skeletal or trajectory relation, the nearest prototype is retrieved from the corresponding learned or random prototype set. As shown in Fig. 3, the learned codebooks yield lower nearest-prototype errors than the random baseline. This indicates better representation of the evaluated held-out relations and supports their use as effective learned representations for correcting initial 3D relations toward plausible human kinematic patterns.

### 4.4 Analysis of Skeletal–Trajectory Aligned Scanning

We analyze the proposed scanning from three perspectives: **(1) Does aligned scanning organize local joint interactions in accordance with skeletal connectivity? (2) Does effective state propagation show greater agreement with the human skeleton? (3) Are the learned dependency graphs more consistent across camera views?**

#### 4.4.1 Does our scanning organize interactions along skeletal connections?

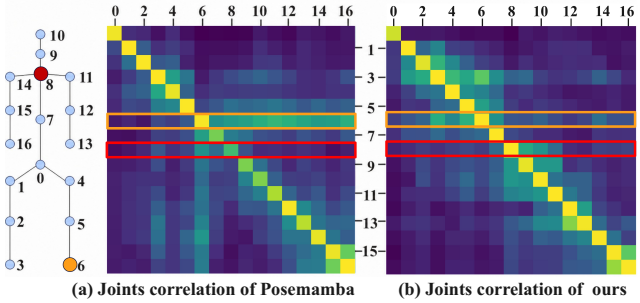



Figure 4: Joint-interaction maps.

In Mamba, the interaction between $\mathbf{B}$ and $\mathbf{C}$ describes the local compatibility between information written by one joint token and read by another. We use these local compatibility maps to examine whether aligned scanning favors skeletal neighbors over cross-chain scan adjacency. Brighter cells indicate stronger local compatibility. Details of the interaction construction and the choice of diagnostic joints are provided in **Appendix A.8.1**. As shown in Fig. 4, joint 8 is a structural attachment, whereas joint 6 terminates a limb chain. Compared with PoseMamba, KAR-Mamba concentrates interactions around skeletal neighbors of joint 8 while weakening cross-chain interactions beyond joint 6. These examples support a local interaction pattern more consistent with skeletal connectivity.

#### 4.4.2 DOES EFFECTIVE STATE PROPAGATION ALIGN WITH THE SKELETAL STRUCTURE?

We construct a joint-dependency graph from effective recurrent state propagation, accounting for input writing, intermediate state transitions, and output reading. We compare its strongest dependencies with the Human3.6M skeleton to assess agreement with skeletal connectivity. Details of the graph construction and quantitative statistics are provided in **Appendix A.8.2**. As shown in Fig. 5, KAR-Mamba has greater overlap between its strongest dependencies and skeletal edges than PoseMamba, with fewer strong nonadjacent connections. This supports a propagation pattern more aligned with skeletal connectivity.

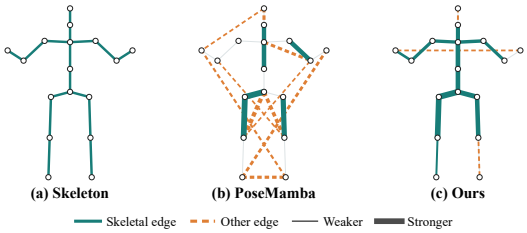



Figure 5: Joint-dependency graphs. Teal lines show skeletal edges, orange lines show non-skeletal edges, and line thickness shows dependency strength.

#### 4.4.3 DOES ALIGNED SCANNING IMPROVE DEPENDENCY CONSISTENCY ACROSS VIEWS?

Aligned scanning combines skeletal topology with anchor-derived estimated 3D relations. We examine whether the resulting dependency graphs remain more stable across camera views of the same 3D pose. This analysis directly evaluates whether the learned dependencies are associated with the underlying 3D structure rather than a specific 2D projection. Fig. 6(a) shows four 2D projections of the same 3D pose. The Top-16 dependency graphs in (b,c) use a fixed joint layout, with line width indicating dependency strength. PoseMamba produces noticeably different strong dependencies across views, including varying cross-region connections, whereas KAR-Mamba preserves more similar torso and limb-chain dependencies. This indicates that aligned scanning reduces the influence of view-dependent 2D patterns and promotes dependency structures consistent with shared 3D kinematic patterns.e7/a The synchronized multi-view construction, evaluation metrics, and complete quantitative results are provided in **Appendix A.8.3**.

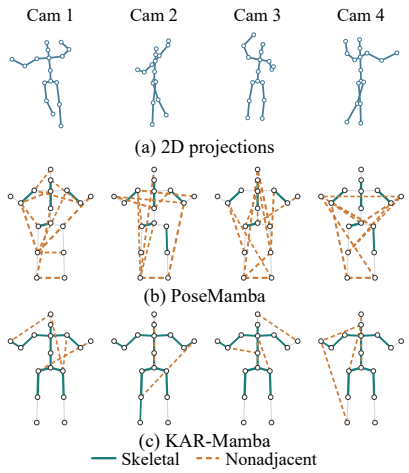



Figure 6: Joint dependencies across camera views.

### 4.5 ANALYSIS OF KINEMATIC-DEPENDENCY-MODULATED STATE UPDATING

The state updating modulation uses anchor-derived skeletal and trajectory dependencies to guide internal state parameters. We examine its behavior through two questions: **(1) Do state-update differences track 3D kinematic differences after modulation? (2) Does the modulation exploit kinematic representations beyond raw 2D features?**

#### 4.5.1 DO STATE-UPDATE DIFFERENCES TRACK 3D KINEMATIC DIFFERENCES?

Kinematic modulation adjusts Mamba's parameters using anchor-derived skeletal and trajectory dependencies. We examine whether larger 3D relation differences are associated with larger state-update differences, with details of the update representation and statistical analysis provided in **Appendix A.9.1**. As shown in Fig. 7(a,c), the state-update discrepancy increases more consistently with larger 3D skeletal or trajectory differences after modulation. Fig. 7(b,d) reports the corresponding Spearman correlations. Both branches show stronger correlations after modulation, and the modulation signals labeled Signal also correlate with the evaluated 3D relations. These associations support a closer correspondence between update configurations and 3D kinematic variation after introducing anchor-derived guidance.

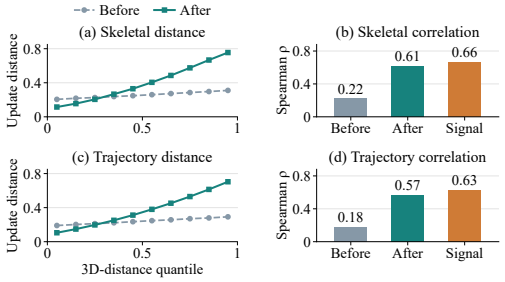



Figure 7: State-update alignment.

### 4.5.2 Does modulation exploit kinematic representations beyond 2D features?

In monocular 3D-HPE, similar local 2D relations can correspond to different 3D structures or motions. We examine whether prior-guided modulation produces different update configurations for locally 2D-similar pairs grouped offline for analysis according to their ground-truth 3D differences, using 2D- and 3D-similar pairs as controls. Details are provided in **Appendix A.9.2**. As shown in Fig. 8(a,c), representative skeletal and trajectory pairs appear similar in 2D but differ in 3D, and their updates become more separated after modulation. Fig. 8(b,d) shows a larger increase in update distance for the 3D-different group than for the controls. This supports an association between prior-guided modulation and grouped offline according to their ground-truth 3D differences among pairs with similar local 2D relations.

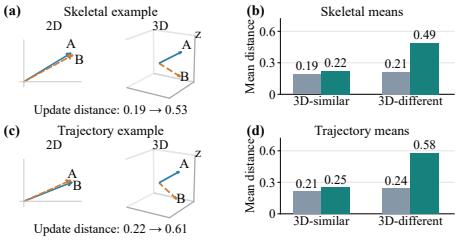



Figure 8: Example pairs and dataset-level update distances for 2D-similar relations.

## 4.6 Ablation Study

We evaluate the three designs and their components using KAR-Mamba-S on Human3.6M (Table 4). B-CB and T-CB denote bone and trajectory codebooks; S-Align and T-Align denote skeletal and trajectory alignment, and CE denotes connection encoding. The $\mathbf{B}$, $\mathbf{C}$, and $\Delta$ columns indicate parameter modulation; ✓/× mark retained/removed components. The full model performs best, supporting complementary component contributions. Detailed definitions and analyses are provided in **Appendix A.10**.

Table 4: Ablations of KAR-Mamba-S on Human3.6M. Results are P1 (MPJPE, mm); lower is better.

| Variants | 3D Pose Anchoring | | Aligned Scanning | | | State Modulation | | | P1↓ |
|---|---|---|---|---|---|---|---|---|---|
| | B-CB | T-CB | S-Align | T-Align | CE | $\mathbf{B}$ | $\mathbf{C}$ | $\Delta$ | |
| Remove all (Baseline) | × | × | × | × | × | × | × | × | 41.8 |
| w/o Codebook Correction | × | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 40.8 |
| w/o Aligned Scanning | ✓ | ✓ | × | × | × | ✓ | ✓ | ✓ | 40.6 |
| w/o Kinematic Modulation | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | 40.1 |
| w/o Bone Codebook | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 40.2 |
| w/o Trajectory Codebook | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 40.0 |
| w/o Skeletal Alignment | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | 40.2 |
| w/o Trajectory Alignment | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ | 40.0 |
| w/o Connection Encoding | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | 39.8 |
| w/o $\mathbf{B}$ Modulation | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ | 39.8 |
| w/o $\mathbf{C}$ Modulation | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | 39.7 |
| w/o $\Delta$ Modulation | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | 39.9 |
| **KAR-Mamba-S (Full)** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **39.5** |

## 4.7 Evaluation of Kinematic Guidance for Pose and Motion Refinement

To examine how KAR-Mamba uses estimated geometry for refinement, we compare observation-derived features, initial 3D relations, and prior-corrected 3D relations while fixing the output pose anchor. This separates the role of kinematic guidance from the quality of the coordinate reference. We further examine temporal consistency over longer sequences, accuracy across actions, and qualitative behavior on challenging poses. The controlled comparison and complementary analyses are detailed in **Appendices A.11, A.12, A.13, and A.14**.

# 5 Discussions and Limitations

The learned codebooks should be viewed as data-driven kinematic representations rather than additional 3D observations. They are learned from training annotations following the standard supervised 3D-HPE setting and are used as compact representations of recurring skeletal and motion patterns during inference.

Although KAR-Mamba relies on learned kinematic prototypes and an estimated pose anchor, these may provide limited guidance for rare or extreme 3D configurations that are poorly represented in the training data, and anchor errors can affect subsequent refinement. More adaptive representations could improve coverage of uncommon motion patterns.

# 6 Conclusion

Our framework augments conventional 2D-to-3D lifting by introducing learned kinematic representations that explicitly guide pose refinement and Mamba state modeling. Codebook-guided anchoring corrects initial skeletal and trajectory estimates, aligned scanning pairs skeletal and temporal traversal with anchor-derived relations, and kinematic modulation conditions state parameters on these estimates. Experiments support the benefits of this design on the evaluated benchmarks and motivate further study of learned 3D priors for spatiotemporal pose refinement.

## REFERENCES


Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2272–2281, 2019.

Hanyuan Chen, Jun-Yan He, Wangmeng Xiang, Zhi-Qi Cheng, Wei Liu, Hanbing Liu, Bin Luo, Yifeng Geng, and Xuansong Xie. Hdformer: High-order directed transformer for 3d human pose estimation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 581–589, 2023. doi: 10.24963/ijcai.2023/65.

Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3D human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):198–209, 2022. doi: 10.1109/TCSVT.2021.3057267.

Hu Cui, Wenqiang Hua, Renjing Huang, Shurui Jia, and Tessai Hayama. Sasmamba: A lightweight structure-aware stride state space model for 3d human pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2721–2730, 2026.

Songlin Du, Zhiwei Yuan, and Takeshi Ikenaga. Kinematics-aware spatial-temporal feature transform for 3d human pose estimation. *Pattern Recognition*, 150:110316, 2024. doi: 10.1016/j.patcog.2024.110316.

Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 6821–6828, 2018. doi: 10.1609/aaai.v32i1.12270.

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 (CVPR)*, pp. 13041–13051, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Mir Rayat Imtiaz Hossain and James J. Little. Exploiting temporal information for 3D human pose estimation. In *Computer Vision – ECCV 2018*, volume 11214 of *Lecture Notes in Computer Science*, pp. 69–86. Springer, 2018. doi: 10.1007/978-3-030-01249-6_5.

Yunlong Huang, Junshuo Liu, Ke Xian, and Robert Caiming Qiu. Posemamba: Monocular 3d human pose estimation with bidirectional global-local spatio-temporal state space model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(4):3842–3850, 2025. doi: 10.1609/aaai.v39i4.32401.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014. doi: 10.1109/TPAMI.2013.248.

Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3D human pose estimation with mixture density network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9887–9895, 2019.

Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13147–13156, 2022.

Jiajie Liu, Mengyuan Liu, Hong Liu, and Wenhao Li. Tcpformer: Learning temporal correlation with implicit pose proxy for 3d human pose estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(5):5478–5486, 2025. doi: 10.1609/aaai.v39i5.32583.

Ye Lu, Jie Wang, Jianjun Gao, Rui Gong, Chen Cai, and Kim-Hui Yap. A structure-aware and motion-adaptive framework for 3d human pose estimation with mamba. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7958–7968, 2025.

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 (ICCV)*, pp. 2640–2649, 2017.

Soroush Mehraban, Vida Adeli, and Babak Taati. Motionagformer: Enhancing 3d human pose estimation with a transformer-gcnformer network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6920–6930, 2024.

Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 Fifth International Conference on 3D Vision (3DV)*, pp. 506–516. IEEE, 2017. doi: 10.1109/3DV.2017.00064.

Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. XNect: Real-time multi-person 3D motion capture with a single RGB camera. *ACM Transactions on Graphics*, 39(4):82:1–82:17, 2020. doi: 10.1145/3386569.3392410.

Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7753–7762, 2019.

Jihua Peng, Yanghong Zhou, and P. Y. Mok. Ktpformer: Kinematics and trajectory prior knowledge-enhanced transformer for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1123–1132, 2024.

Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. In *Computer Vision – ECCV 2022*, pp. 461–478. Springer, 2022.

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 (ICCV)*, pp. 14761–14771, 2023.

Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2602–2611, 2017.

Ziming Sun, Yuan Liang, Zejun Ma, Tianle Zhang, Linchao Bao, Guiqing Li, and Shengfeng He. Repose: 3d human pose estimation via spatio-temporal depth relational consistency. In *Computer Vision – ECCV 2024*, volume 15077 of *Lecture Notes in Computer Science*, pp. 309–325. Springer, 2024. doi: 10.1007/978-3-031-72655-2_18.

Zhenhua Tang, Zhaofan Qiu, Yanbin Hao, Richang Hong, and Ting Yao. 3d human pose estimation with spatio-temporal criss-cross attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4790–4799, 2023.

Scott D. Uhlrich, Antoine Falisse, Łukasz Kidziński, Julie Muccini, Michael Ko, Akshay S. Chaudhari, Jennifer L. Hicks, and Scott L. Delp. OpenCap: Human movement dynamics from smartphone videos. *PLOS Computational Biology*, 19(10):e1011462, 2023. doi: 10.1371/journal.pcbi.1011462.

Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic monocular 3d human pose estimation with normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11199–11208, 2021.

Bruce X. B. Yu, Zhi Zhang, Yongxu Liu, Sheng-hua Zhong, Yan Liu, and Chang Wen Chen. Glagcn: Global-local adaptive graph convolutional network for 3d human pose estimation from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8818–8829, 2023.

Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13232–13242, 2022.

Xinyi Zhang, Qiqi Bao, Qinpeng Cui, Wenming Yang, and Qingmin Liao. Pose magic: Efficient and temporally consistent human pose estimation with a hybrid mamba-gcn network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(10):10248–10256, 2025. doi: 10.1609/aaai.v39i10.33112.

Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3425–3435, 2019.

Qitao Zhao, Ce Zheng, Mengyuan Liu, Pichao Wang, and Chen Chen. Poseformerv2: Exploring frequency domain for efficient and robust 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8877–8886, 2023.

Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11656–11665, 2021.

Kaili Zheng, Feixiang Lu, Yihao Lv, Liangjun Zhang, Chenyi Guo, and Ji Wu. 3D human pose estimation via non-causal retentive networks. In *Computer Vision – ECCV 2024*, volume 15091 of *Lecture Notes in Computer Science*, pp. 111–128. Springer, 2024. doi: 10.1007/978-3-031-73414-4_7.

Zenghao Zheng, Lianping Yang, Jinshan Pan, and Hegui Zhu. Mamba-driven topology fusion for monocular 3d human pose estimation. *Image and Vision Computing*, 168:105927, 2026. doi: 10.1016/j.imavis.2026.105927.

Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15085–15099, 2023.

Zhiming Zou and Wei Tang. Modulated graph convolutional network for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11477–11487, 2021.

# A Appendix

## A.1 Related Work

### A.1.1 Mamba-Based Methods in 3D-HPE

Mamba has recently been introduced into monocular 3D-HPE for its linear computational complexity and long-range dependency modeling. PoseMamba Huang et al. (2025) designed bidirectional global-local spatiotemporal state-space modeling with joint reordering. Pose Magic Zhang et al. (2025) combined Mamba with GCNs to complement global sequence modeling with local skeletal relations. SAMA Lu et al. (2025) incorporated learned joint topology into feature and hidden-state fusion, while deriving motion-adaptive timescales from neighboring-frame joint features. Sas-Mamba Cui et al. (2026) introduced structure-aware stride scanning to preserve local and global pose structures during state propagation. More recently, MDTF Zheng et al. (2026) estimated approximate 3D bone directions and lengths and fused them with joint features to provide explicit skeletal guidance for Mamba-based refinement.

These methods demonstrate the value of structural and motion cues in state-space modeling, ranging from observation-derived features to estimated 3D skeletal geometry. Our framework constructs a geometric reference from learned target-space relations. Bone and joint-displacement prototypes learned from training 3D poses first correct initial skeletal and trajectory estimates, from which a full pose anchor is reconstructed. Its relations enrich tokens along fixed skeletal and temporal scans and directly modulate Mamba's $B$, $C$, and $\Delta$ parameters, while its coordinates provide the reference for residual pose decoding. This connects prior-guided correction of the estimated 3D geometry with its use in both internal state modeling and final coordinate refinement.

### A.1.2 Kinematic Modeling in 3D-HPE

Kinematic modeling has been widely explored to improve the structural and temporal reliability of monocular 3D-HPE beyond coordinate-wise regression. MotionBERT Zhu et al. (2023) learns transferable human-motion representations containing geometric, kinematic, and physical knowledge by recovering 3D motion from noisy and incomplete 2D observations. KTPFormer Peng et al. (2024) incorporates human kinematic topology and joint trajectory priors into attention modeling. Du et al. (2024) construct kinematics-aware features from local and global joint groups and employ spatiotemporal feature transformation to strengthen structural correlations. RePOSE Sun et al. (2024) introduces spatiotemporal relative-depth consistency as supervision to improve 3D pose recovery under challenging observations such as occlusion.

These approaches exploit kinematics through learned representations, topology priors, feature construction, or supervisory constraints. Our learned bone and trajectory codebooks organize recurring 3D relations into prototypes that adjust an initial pose estimate before refinement. The resulting anchor provides a shared geometric reference for the relations conditioning the refinement network and the coordinate offsets predicted at its output.

## A.2 Supervision Objectives

The bone and trajectory codebooks are first optimized using $\mathcal{L}_{\mathrm{CB}}$ defined in Sec. 3.1 and are subsequently frozen. They therefore provide fixed target-space kinematic representations and receive no gradient during main-network training. The pose network is supervised at two levels: the final refined pose is constrained in coordinate, scale, and temporal motion, while the intermediate anchor is constrained in both its 3D coordinates and the kinematic relations extracted from it.

Let $\mathrm{P}_{\mathrm{GT}} \in \mathbb{R}^{T\times J\times 3}$ denote the root-relative ground-truth sequence, and let $\Delta_t(\mathrm{P}) = \mathrm{P}_{2:T} - \mathrm{P}_{1:T-1}$ denote the temporal difference operator. The coordinate loss for the refined pose is the mean per-joint Euclidean distance

$$\mathcal{L}_{\mathrm{pos}} = \frac{1}{TJ}\sum_{t=1}^{T}\sum_{j=1}^{J} \left\|\mathrm{P}_{3\mathrm{D},t,j} - \mathrm{P}_{\mathrm{GT},t,j}\right\|_2 . \tag{8}$$

We additionally use a scale-normalized pose loss

$$\mathcal{L}_{\mathrm{scale}} = \mathcal{L}_{\mathrm{pos}}\left(s^{\star}\mathrm{P}_{3\mathrm{D}}, \mathrm{P}_{\mathrm{GT}}\right), \qquad s^{\star} = \frac{\langle \mathrm{P}_{3\mathrm{D}}, \mathrm{P}_{\mathrm{GT}}\rangle_J}{\|\mathrm{P}_{3\mathrm{D}}\|_{F,J}^2}, \tag{9}$$

where the inner product and squared norm are evaluated over the joints and coordinates of each frame. Temporal motion is supervised by matching the first-order joint displacements:

$$\mathcal{L}_{\text{vel}} = \frac{1}{(T-1)J} \sum_{t=1}^{T-1} \sum_{j=1}^{J} \left\| \Delta_t(\mathrm{P}_{\text{3D}})_{t,j} - \Delta_t(\mathrm{P}_{\text{GT}})_{t,j} \right\|_2 . \tag{10}$$

We further apply a joint-weighted temporal smoothness regularizer

$$\mathcal{L}_{\text{smooth}} = \frac{1}{(T-1)J} \sum_{t=1}^{T-1} \sum_{j=1}^{J} w_j \left\| \Delta_t(\mathrm{P}_{\text{3D}})_{t,j} \right\|_2^2 , \tag{11}$$

where $w_j$ is a fixed joint-dependent weight that assigns larger weights to distal limb joints.

To supervise the estimated pose anchor used by the subsequent Mamba blocks, we constrain both the anchor coordinates and its extracted kinematic attributes. Let $\mathcal{K}_s^+$ and $\mathcal{K}_s^-$ extract the magnitude and direction of the forward and backward skeletal relations, and let $\mathcal{K}_t^+$ and $\mathcal{K}_t^-$ extract the corresponding attributes of the forward and backward trajectory relations. The anchor-level kinematic loss is

$$\mathcal{L}_{\text{phy}} = \frac{1}{4} \sum_{q \in \{s^+, s^-, t^+, t^-\}} \text{SmoothL1}(\mathcal{K}_q(\mathrm{P}_{\text{Anchor}}), \mathcal{K}_q(\mathrm{P}_{\text{GT}})) , \tag{12}$$

and the anchor-coordinate loss is

$$\mathcal{L}_{\text{anchor}} = \frac{1}{TJ} \sum_{t=1}^{T} \sum_{j=1}^{J} \left\| \mathrm{P}_{\text{Anchor},t,j} - \mathrm{P}_{\text{GT},t,j} \right\|_2 . \tag{13}$$

The complete objective for main-network training is

$$\begin{aligned} \mathcal{L}_{\text{total}} = \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{scale}}\mathcal{L}_{\text{scale}} + \lambda_{\text{vel}}\mathcal{L}_{\text{vel}} + \lambda_{\text{smooth}}\mathcal{L}_{\text{smooth}} \\ + \lambda_{\text{phy}}\mathcal{L}_{\text{phy}} + \lambda_{\text{anchor}}\mathcal{L}_{\text{anchor}}. \end{aligned} \tag{14}$$

We set $(\lambda_{\text{pos}}, \lambda_{\text{scale}}, \lambda_{\text{vel}}, \lambda_{\text{smooth}}, \lambda_{\text{phy}}, \lambda_{\text{anchor}}) = (1.0, 0.5, 20.0, 0.5, 1.0, 0.1)$ in all main experiments. The first four terms supervise the refined output, whereas the last two encourage the intermediate anchor and its estimated skeletal and trajectory relations to match the training targets for aligned scanning and state modulation.

### A.3 Datasets, Evaluation Metrics, and Implementation Details

#### A.3.1 Datasets and Evaluation Metrics

Human3.6M is a widely used indoor benchmark for 3D human pose estimation,comprising approximately 3.6 million frames collected from 11 subjectsperforming 15 daily activities, such as walking, sitting, eating, and taking photos. We follow the standard subject split, using S1, S5, S6, S7, and S8 for training, and S9 and S11 for testing. Performance is reported under two standard evaluation protocols. Protocol 1 (P1) uses Mean Per Joint Position Error (MPJPE), measured in millimeters after root-joint alignment between the predicted and ground-truth poses. Protocol 2 (P2) adopts Procrustes-aligned MPJPE (P-MPJPE), where the prediction is aligned to the ground truth through translation, rotation, and scaling before computing the error.

MPI-INF-3DHP consists of approximately 1.3 million frames and covers more diverse viewpoints, scenes, clothing conditions, and motion patterns than Human3.6M. It contains both indoor and outdoor sequences, making it suitable for evaluating model performance under more varied and challenging conditions. Following the commonly used evaluation protocol, we report the 3D Percentage of Correct Keypoints (PCK), theArea Under the PCK Curve (AUC), and MPJPE. PCK measures the proportion of predicted joints whose errors fall within a predefined distance threshold, while AUC summarizes the PCK performance over a range of thresholds.

#### A.3.2 Implementation Details

KAR-Mamba is implemented in PyTorch and trained on a single NVIDIA RTX 4090 GPU. For fair comparison, we follow PoseMamba and adopt the same detected and ground-truth 2D pose inputs, preprocessing pipeline, and data augmentation settings. We build three model variants with different Mamba depths $L$ and channel dimensions $C$: KAR-Mamba-S with $L = 8$ and $C = 64$, KAR-Mamba-B with $L = 8$ and $C = 128$, and KAR-Mamba-L with $L = 16$ and $C = 256$. These

variants are used to evaluate the proposed framework under different model capacities. All models are trained from scratch for 120 epochs using AdamW. The initial learning rate is $5 \times 10^{-4}$ with a weight decay of 0.01, and is multiplied by 0.98 after each training epoch. Unless otherwise stated, all remaining training and evaluation settings are kept identicalacross the three variants.

### A.4 How Sensitive Is KAR-Mamba to Its Hyperparameters and Supervision Weights?

We further examine the sensitivity of KAR-Mamba to the key hyperparameters involved in pose anchoring and training. Specifically, we study the codebook correction strength $\lambda_{\text{cor}}$, the initialization of the learnable anchor fusion weight $\alpha$, and the supervision weights $\lambda_{\text{phy}}$ and $\lambda_{\text{anchor}}$. During each study, only the corresponding parameter is varied while all remaining settings are kept unchanged.

Table 5: Sensitivity analysis of key hyperparameters and supervision weights on Human3.6M. The default settings are highlighted in bold. P1 denotes MPJPE in mm.

| Correction strength | | Anchor fusion | | | Supervision weights | |
|---|---|---|---|---|---|---|
| $\lambda_{\text{cor}}$ | P1↓ | $\alpha_0$ | $\alpha$ | P1↓ | $(\lambda_{\text{phy}}, \lambda_{\text{anchor}})$ | P1↓ |
| 0.1 | 39.9 | 0.3 | 0.39 | 39.6 | $(0.5, 0.1)$ | 39.8 |
| 0.2 | 39.7 | **0.5** | **0.38** | **39.5** | $(1.5, 0.1)$ | 39.7 |
| **0.3** | **39.5** | 0.7 | 0.40 | 39.6 | $(1.0, 0.05)$ | 39.7 |
| 0.4 | 39.6 | | | | $(\mathbf{1.0}, \mathbf{0.1})$ | **39.5** |
| 0.5 | 39.8 | | | | $(1.0, 0.2)$ | 39.6 |

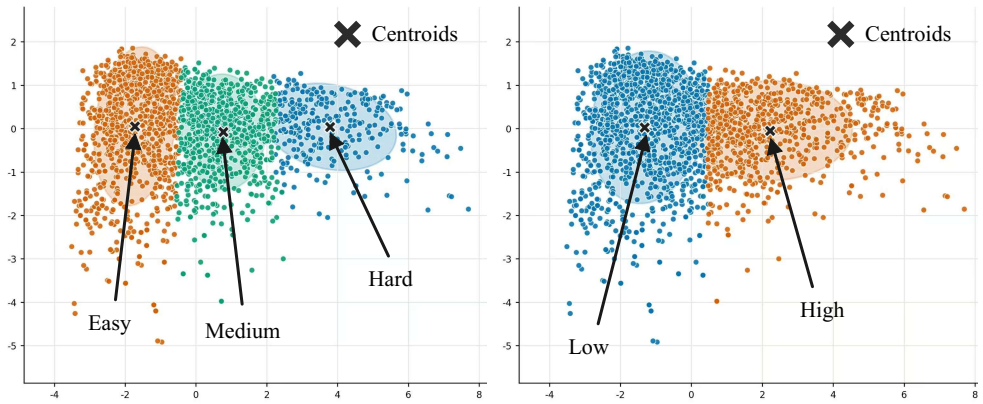
As shown in Table 5, KAR-Mamba remains relatively stable over a reasonable range of hyperparameter choices. For the codebook correction strength, increasing $\lambda_{\text{cor}}$ from 0.1 to 0.3 gradually improves P1 from 39.9 to 39.5 mm, suggesting that an appropriate degree of codebook-guided adjustment benefits the final pose estimates. However, further increasing the correction strength does not yield additional improvements and slightly degrades the performance to 39.8 mm at $\lambda_{\text{cor}} = 0.5$. This suggests that the codebook should serve as a learned prior rather than completely dominate the sample-specific relation. We therefore adopt $\lambda_{\text{cor}} = 0.3$ as the default setting.

We next study the initialization of the learnable fusion weight $\alpha$, which balances the skeletal and trajectory reconstructions when constructing the 3D pose anchor. Initializing $\alpha$ to 0.3, 0.5, or 0.7 results in very similar final performance, with the learned values converging to approximately 0.39, 0.38, and 0.40, respectively. The corresponding P1 varies by only 0.1 mm. This indicates that the final skeletal–trajectory balance is learned during optimization rather than being strongly determined by its initial value. We therefore use the neutral initialization $\alpha_0 = 0.5$.

Finally, we vary $\lambda_{\text{phy}}$ and $\lambda_{\text{anchor}}$ while keeping the remaining training objectives unchanged. The model achieves 39.5 mm under the default setting $(\lambda_{\text{phy}}, \lambda_{\text{anchor}}) = (1.0, 0.1)$. Moderately decreasing or increasing either weight changes the final error by only 0.1–0.3 mm. These results indicate that performance remains stable within the tested range, with the default weights producing the lowest final pose error among the evaluated supervision settings.

### A.5 Construction of Challenging Human3.6M Subsets

The standard Human3.6M test set contains a large proportion of samples with relatively reliable 2D observations and regular motion patterns. Although these samples are important for evaluating overall pose estimation accuracy, their dominance in the test distribution may dilute the performance differences among methods under more challenging conditions. To examine performance on selected difficult cases, we construct two challenging subsets from the Human3.6M test set, focusing on observation unreliability and motion complexity, respectively. These subsets are used only for evaluation and do not alter the original training or testing protocol. Rather than manually defining difficulty through fixed thresholds, we employ unsupervised K-Means clustering to identify challenging samples according to their empirical distributions. The resulting clustering distributions are visualized using PCA in Fig. 9, where PCA is used only for visualization.




**(a) Observation confidence clustering distribution** **(b) Motion intensity clustering distribution**

Figure 9: Visualization of the challenging-subset construction on Human3.6M. (a) Observation-confidence clustering into Easy, Medium, and Hard groups. (b) Motion-intensity clustering into Low and High groups. The Hard and High clusters are selected as the Low-Confidence and High-Motion subsets, respectively. Crosses denote cluster centroids.

For observation difficulty, we characterize each sample using the confidence statistics of its detected 2D joints. Low-confidence detections may reflect localization uncertainty caused by occlusion, self-overlap, appearance ambiguity, or detector failure. Since monocular 3D-HPE relies on these 2D poses as observations, localization errors can affect depth inference and subsequent 3D reconstruction. We perform K-Means clustering based on the 2D joint-confidence statistics and select the cluster with the lowest average confidence as the Low-Confidence subset. As shown in Fig. 9(a), the confidence-based descriptors form three clusters, from which the lowest-confidence cluster is retained for evaluation. This subset contains approximately 10% of the Human3.6M test samples. The remaining higher-confidence clusters are not separately evaluated, since our purpose is not to establish a complete difficulty hierarchy but to specifically examine robustness under highly unreliable observations.

To further characterize challenging cases from the temporal perspective, we additionally consider motion complexity. Even with reasonably accurate 2D detections, rapid or irregular motion can produce substantial temporal variation and make spatio-temporal dependency modeling considerably more difficult. We therefore construct a motion descriptor by integrating four complementary cues: joint velocity, acceleration, motion-direction variation, and frequency-domain motion information. Velocity reflects the magnitude of inter-frame displacement, while acceleration describes how rapidly the motion state changes. Directional variation captures changes in local motion trajectories, whereas high-frequency components characterize rapid and abrupt temporal variations that cannot be fully represented by displacement magnitude alone. These cues summarize different aspects of motion dynamics. We perform K-Means clustering using this motion descriptor and select the cluster with the highest motion intensity as the High-Motion subset. As illustrated in Fig. 9(b), the samples are separated into Low- and High-Motion clusters, with the latter accounting for approximately 25% of the Human3.6M test set. The Low-Motion cluster is used only to establish the unsupervised partition and is not reported as an additional evaluation subset.

The use of unsupervised clustering avoids manually specifying fixed confidence or motion thresholds, whose effectiveness may depend on the numerical scale and distribution of the corresponding descriptors. Instead, the challenging groups are identified according to the empirical distribution of the test samples. As shown in Fig. 9, both observation-confidence and motion-intensity features exhibit clear clustering structures in the projected space. We retain the lowest-confidence and highest-motion clusters for the subset evaluation. In this way, the Low-Confidence and High-Motion subsets complement the standard Human3.6M evaluation by specifically revealing model behavior when 2D observations become unreliable or the underlying human motion becomes highly dynamic. The clustering procedure is only used to partition evaluation samples and does not involve model selection or hyperparameter tuning.

### A.6 How does codebook capacity affect prototype coverage and utilization?

We first examine how the capacity of the kinematic codebooks affects their ability to represent diverse 3D relation patterns. A small codebook may provide insufficient coverage of skeletal and trajectory variations, whereas an excessively large one may introduce redundant prototypes that are only sparsely activated. We therefore vary the number of prototypes associated with each bone and joint trajectory as $K \in \{16, 32, 64, 128, 256\}$, while keeping all other training and evaluation settings unchanged.

We evaluate each configuration from three complementary aspects: prototype utilization, relation representation quality, and final pose accuracy. Prototype utilization is measured using an entropy-based effective utilization score computed from the assignment frequencies of held-out 3D relations, where a larger value indicates that the available prototypes are used more evenly. Following Sec. 4.3.2, we additionally report the 3D direction error and relative magnitude error between each held-out relation and its nearest prototype.

Table 6: Influence of codebook size on kinematic representation and pose estimation on Human3.6M. Util. denotes entropy-based effective prototype utilization. Dir. and Mag. denote the 3D direction error and relative magnitude error of the nearest prototype, respectively.

| $K$ | Bone Codebook | | | Trajectory Codebook | | | P1↓ |
|---|---|---|---|---|---|---|---|
| | Util.↑ | Dir.↓ | Mag.↓ | Util.↑ | Dir.↓ | Mag.↓ | |
| 16 | 94.8% | 5.8° | 5.6% | 92.6% | 10.5° | 12.7% | 40.2 |
| 32 | 89.7% | 4.5° | 4.7% | 86.9% | 8.4° | 10.1% | 39.8 |
| **64** | **82.6%** | 3.7° | 4.1% | **78.8%** | 7.2° | 8.9% | **39.5** |
| 128 | 68.9% | 3.4° | 3.9% | 64.7% | 6.8° | 8.5% | 39.6 |
| 256 | 52.1% | **3.2°** | **3.8%** | 47.6% | **6.6°** | **8.3%** | 39.8 |

As shown in Table 6, increasing the codebook size from 16 to 64 consistently improves the representation of both skeletal and trajectory relations. For the bone codebook, the direction error decreases from $5.8^\circ$ to $3.7^\circ$, while the relative magnitude error decreases from 5.6% to 4.1%. The trajectory codebook exhibits the same trend, with the direction and magnitude errors decreasing from $10.5^\circ$ and 12.7% to $7.2^\circ$ and 8.9%, respectively. These improvements are accompanied by a reduction in P1 from 40.2 to 39.5 mm, suggesting that improved kinematic coverage can benefit subsequent pose refinement.

Further increasing $K$ to 128 or 256 yields only marginal reductions in nearest-prototype representation errors, while prototype utilization drops substantially. At $K = 256$, the effective utilization decreases to 52.1% for the bone codebook and 47.6% for the trajectory codebook, and the final P1 increases to 39.8 mm. These results suggest diminishing benefits from additional, more sparsely used prototypes. Among the tested settings, $K = 64$ provides the best balance among kinematic coverage, prototype utilization, and downstream pose accuracy.

### A.7 How Should the Kinematic Codebooks Adjust Initial 3D Relations?

The learned kinematic codebooks provide representative 3D relations learned from training data, but directly replacing the relation of the current sample with a prototype may discard sample-specific skeletal geometry or motion variation. We therefore compare different strategies for using the learned prototypes to adjust the initial skeletal and trajectory relations. All variants use the same codebooks and downstream network, differing only in how the codebook relations are introduced during pose anchoring.

No Correction directly retains the relations derived from the initial 3D prediction without using the codebooks. Hard Prototype Replacement replaces each initial relation with its nearest prototype from the corresponding bone- or joint-specific codebook. Soft Prototype Replacement instead uses the attention-weighted prototype aggregation as the adjusted relation, allowing multiple representative patterns to contribute. In contrast, our Residual Attention Correction retains the initial sample-specific relation and uses the codebook aggregation only to adjust it toward learned representative 3D relations:

$$\mathbf{R}_{\mathrm{Cor}} = \mathbf{R}_{\mathrm{Init}} + \lambda_{\mathrm{cor}} \left( \mathrm{Attn}(\mathbf{R}_{\mathrm{Init}}, \mathbf{CB}) - \mathbf{R}_{\mathrm{Init}} \right), \tag{15}$$

where $\mathbf{R}$ denotes either a skeletal or trajectory relation.

As shown in Table 7, directly using the initial relations results in an anchor error of 43.4 mm and a final P1 of 40.8 mm. Replacing each relation with its nearest prototype reduces the errors to 42.8

and 40.4 mm, respectively, supporting the usefulness of the learned codebooks as 3D kinematic representations. Soft prototype aggregation further improves the final P1 to 39.9 mm, showing that combining multiple relevant prototypes is more effective than selecting a single discrete relation in this comparison. However, directly replacing the current relation with the aggregated prototype does not retain an explicit residual path from the initial relation.

The proposed residual correction achieves the lowest anchor and final errors of 41.3 and 39.5 mm, respectively. Compared with soft prototype replacement, the only difference is that the current relation is explicitly retained while the codebook provides a controlled adjustment toward learned representative 3D relations. These results favor retaining the estimated relation and using the codebook as a prior-guided adjustment rather than as a direct substitute in the evaluated setting.

Table 7: Comparison of different codebook correction strategies on Human3.6M. Anchor P1 denotes the MPJPE of $\mathbf{P}_{\mathrm{Anchor}}$ before Mamba refinement.

| Correction Strategy | Preserve Current | Anchor P1↓ | Final P1↓ |
|---|---|---|---|
| No Correction | ✓ | 43.4 | 40.8 |
| Hard Prototype Replacement | × | 42.8 | 40.4 |
| Soft Prototype Replacement | × | 41.9 | 39.9 |
| **Residual Attention Correction** | ✓ | **41.3** | **39.5** |

## A.8 Additional Analysis of Skeletal-Trajectory Aligned Scanning

### A.8.1 Local Joint-Interaction Analysis

We analyze local joint interactions to examine whether aligned scanning favors skeletal connectivity over adjacency introduced by the scan order. The interaction between Mamba's $\mathbf{B}$ and $\mathbf{C}$ parameters characterizes local write–read compatibility between joint tokens and serves as a proxy for their local interaction strength.

We focus on joints 8 and 6 because they provide two complementary diagnostic cases. Joint 8 is an attachment joint between the torso and upper limbs and is used to examine whether local interactions concentrate on skeletal neighbors. Joint 6 terminates a limb chain, after which the scan switches to another anatomical chain, and is therefore used to examine whether interaction across that scan boundary is reduced. The connection encoding distinguishes within-chain transitions from switches between chains.

As shown in Fig. 4, PoseMamba exhibits more dispersed interactions around joint 8 and retains noticeable interactions beyond joint 6. In contrast, KAR-Mamba concentrates the interactions around kinematically related joints while weakening the cross-chain interaction after joint 6. These examples show a local interaction pattern that more closely follows skeletal connectivity.

### A.8.2 Effective State-Propagation Graph Analysis

To examine whether these local relations remain meaningful after recurrent state propagation, we construct a joint-dependency graph from the effective propagation between joints. The propagation strength jointly reflects information written by the source joint, transferred through intermediate states, and finally read by the receiving joint. It therefore describes the dependency formed through the complete state propagation rather than only a direct local interaction.

Because the spatial scan contains five anatomical chains and duplicates several attachment joints, we first restore the forward and backward propagation results to a common scan order and aggregate them across frames, Mamba blocks, and test sequences. The 21 scan positions are then merged according to joint identity to recover the original 17-joint layout. After removing self-connections and combining the two propagation directions, we retain the 16 strongest undirected dependencies, matching the 16 parent–child edges in the Human3.6M skeleton.

As shown in Fig. 5, only 6 of PoseMamba's strongest dependencies coincide with true skeletal edges, whereas 13 of KAR-Mamba's strongest dependencies do. The dependency mass on one-hop skeletal neighbors also increases from 39.40% to 66.50%, while the mass beyond two skeletal hops decreases from 31.90% to 9.30%. These results show that KAR-Mamba shifts effective state propagation toward skeletal neighbors while reducing the mass assigned to joints farther apart on the skeleton.

### A.8.3 Cross-View Dependency Consistency

We group synchronized Human3.6M samples by subject, action, subaction, and frame, keeping the underlying 3D motion fixed while varying the camera projection.

For each camera, we extract a joint-dependency graph. Graph Cosine compares the complete weighted graphs, and Top-16 Jaccard measures the overlap of their 16 strongest dependencies. We evaluate ground-truth (GT) 2D joints to isolate projection changes and detected 2D joints to include observation noise.

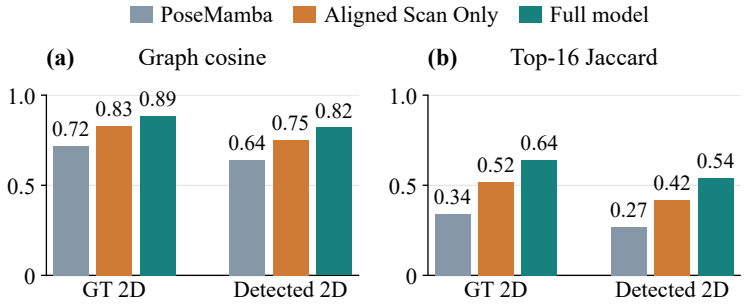



Figure 10: Cross-view dependency consistency with ground-truth and detected 2D inputs.

Figure 10 reports mean scores over synchronized camera groups. Aligned Scan Only denotes the model without kinematic parameter modulation. Both metrics improve with aligned scanning and improve further with the full model under both input settings. Consistent with the examples in Fig. 6, these results support more stable dependencies across different projections of the same 3D motion.

## A.9 Additional Analysis of Kinematic-Dependency-Modulated State Updating

### A.9.1 Correspondence Between 3D Kinematic Relations and State Updates

This experiment examines the correspondence between 3D kinematic differences and Mamba's state-parameter differences before and after anchor-derived modulation. To isolate the effect of kinematic modulation, we extract the state parameters before and after modulation from the same model while keeping the pose features, scan order, connection encoding, and network weights unchanged. Thus, the difference between the two parameter configurations reflects the modulation introduced by the anchor-derived relations.

For each aligned state transition, we separately normalize $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{\Delta}$ and combine them into a state-update representation. Cosine distance between two such representations measures the difference in their update configurations, while normalization reduces the influence of absolute parameter scale. To avoid differences between anatomical locations dominating the comparison, spatial relations are compared only within the same bone identity, and temporal relations only within the same joint identity.

The 3D kinematic difference considers both magnitude and direction. For skeletal relations, these correspond to bone scale and 3D orientation; for trajectory relations, they correspond to displacement magnitude and motion direction. Within each identity, we compare the 3D kinematic difference with the state-update distance and use Spearman correlation to measure whether they change consistently. Fig. 7(a,c) further groups relation pairs into ten bins according to their 3D kinematic differences to show how the update distance changes as the 3D relation becomes increasingly different, while Fig. 7(b,d) summarizes the corresponding identity-wise correlations.

Together, the trend and correlation analyses show that, after modulation, larger 3D skeletal or trajectory differences are associated with larger state-update differences. This supports an association between the evaluated 3D relation differences and Mamba's modulated update parameters.

### A.9.2 Construction and Analysis of 2D-Similar but 3D-Different Relations

This experiment examines whether prior-guided modulation produces different parameter responses for relations that are similar in 2D but differ in ground-truth 3D geometry or motion. We construct relation pairs that are highly similar in 2D but either similar or different in 3D, so that the observed 2D relation similarity is controlled while the ground-truth 3D relation varies.

For the spatial branch, relation pairs are formed within the same bone identity and camera stream using body-scale-normalized 2D bone vectors. For the temporal branch, pairs are formed using

normalized inter-frame 2D joint displacements. We first retain the 10% of pairs with the smallest 2D distance to ensure strong similarity in the observed 2D relations. These pairs are then divided offline according to their corresponding ground-truth 3D relations: the 20% with the largest 3D distance form the 3D-different group, representing similar 2D observations with clearly different underlying 3D structures or motions, while the 20% with the smallest 3D distance form the 3D-similar control group. Self-pairs and temporally adjacent near-duplicates are excluded, and all identities contribute equally to the final statistics.

This construction compares two groups selected for high local 2D similarity but differing in ground-truth 3D relations. Comparing their changes in state-update distance examines the association between prior-guided modulation and these 3D differences.

Fig. 8 combines representative examples with dataset-level statistics. The left panels show skeletal and trajectory pairs that are similar in 2D but clearly different in 3D, while the right panels compare the average update distances of the 3D-similar and 3D-different groups. Both the examples and the dataset-level results show that the 3D-different group becomes much more separated after modulation, whereas the 3D-similar control group changes only slightly. This supports the ability of anchor-derived modulation to produce parameter differences associated with the ground-truth-defined groups, even when their local 2D relations are highly similar.

### A.10 Detailed Ablation Analysis

We expand the ablation study in Sec. 4.6 by analyzing the contribution of the three proposed stages and their individual components: Codebook-Guided 3D Pose Anchoring, Skeletal-Trajectory Aligned Scanning, and Kinematic-Dependency-Modulated State Updating. All experiments are conducted on Human3.6M using KAR-Mamba-S. Each variant removes the components indicated in Table 4, while the remaining settings are kept unchanged. Removing all three designs gives a baseline P1 of 41.8 mm, compared with 39.5 mm for the full model.

Removing the codebook-guided correction, while retaining the initial 3D prediction as the pose reference, increases the error from 39.5 mm to 40.8 mm, supporting the contribution of codebook-guided anchoring to final pose accuracy. We further examine the two types of kinematic references used in pose anchoring. Removing the bone codebook increases P1 to 40.2 mm, while removing the trajectory codebook increases it to 40.0 mm. These results indicate that skeletal geometry and temporal motion provide complementary information for constructing the 3D pose anchor.

We next evaluate the internal components of Skeletal-Trajectory Aligned Scanning. Removing skeletal alignment or trajectory alignment increases P1 to 40.2 mm and 40.0 mm, respectively, supporting the contributions of the corresponding spatial and temporal alignment designs. Removing connection encoding results in 39.8 mm, supporting the usefulness of distinguishing skeletal transitions from inter-chain switches. When the entire aligned scanning module is removed, the error increases to 40.6 mm, supporting the overall benefit of aligned scanning with anchor-derived relations.

Finally, we examine the contribution of different selective-state parameter modulations. Removing the entire kinematic modulation increases P1 to 40.1 mm. Removing the modulation of $\mathbf{B}$, $\mathbf{C}$, or $\Delta$ results in P1 errors of 39.8 mm, 39.7 mm, and 39.9 mm, respectively, while jointly modulating all three parameters achieves the best performance of 39.5 mm. This suggests that these parameters provide complementary mechanisms for incorporating anchor-derived 3D kinematic relations into Mamba's state updating.

### A.11 Does Prior-Corrected Kinematic Guidance Improve Pose Refinement?

KAR-Mamba uses its pose anchor both as the coordinate reference for residual prediction and as the source of skeletal and trajectory relations for refinement. Improved final accuracy may therefore reflect either a better starting pose or more useful geometric guidance. To distinguish these effects, we keep the output anchor unchanged and vary the representation that guides token modeling and state updating. This comparison examines whether learned target-space relations help the framework refine an imperfect pose estimate beyond the benefit of anchor-relative prediction.

We use KAR-Mamba-S on Human3.6M with $T = 243$. A common front end is first trained with the anchor-coordinate and kinematic-relation objectives in Appendix A.2, independently of the refinement network, with codebooks learned from training 3D poses. The front end is then frozen, providing identical pose features, initial predictions $\mathrm{P}_{\mathrm{Init}}$, and corrected anchors $\mathrm{P}_{\mathrm{Anchor}}$ for all variants. Each refinement network is trained separately with the same pose supervision, scanning

layout, and training budget. All variants retain the same feature-fusion and $\mathbf{B}$, $\mathbf{C}$, and $\Delta$ modulation mechanisms, with condition mappings of the same hidden width and depth.

We compare three sources of guidance. *Observation-derived features* use paired contextual joint features along skeletal edges and adjacent frames, before incorporating anchor geometry. *Initial 3D relations* use bone vectors and inter-frame joint displacements extracted from $\mathrm{P}_{\mathrm{Init}}$. *Prior-corrected 3D relations* use the corresponding relations from $\mathrm{P}_{\mathrm{Anchor}}$. Each source supplies both relation-feature fusion and state-parameter modulation. All variants predict residual offsets from the same corrected anchor, including the variant guided by initial 3D relations.

Table 8: Pose and motion refinement with a common output anchor on Human3.6M. Anchor P1 and final P1 are MPJPE in mm; MPJVE is in mm/frame.

| Guidance source | Anchor P1 | Final P1↓ | MPJVE↓ |
|---|---|---|---|
| Observation-derived features | 42.0 | 40.9 | 1.70 |
| Initial 3D relations | 42.0 | 40.5 | 1.66 |
| **Prior-corrected 3D relations** | 42.0 | **40.0** | **1.61** |

Table 8 organizes the comparison around the same coordinate reference. Comparing observation-derived features with initial 3D relations tests the value of explicitly representing geometry for refinement. Comparing initial and prior-corrected relations further tests whether correcting that geometry improves its usefulness as guidance. The position and velocity errors measure the effectiveness of prior-corrected kinematic guidance when the output anchor is held fixed. This comparison evaluates their use across the refinement framework; feature-level and state-level contributions are examined separately in the component ablations.

### A.12 Does KAR-Mamba Maintain Temporal Consistency over Longer Sequences?

KAR-Mamba uses estimated trajectory relations derived from the pose anchor to guide temporal modeling. We therefore examine whether these relations improve pose and motion accuracy over different temporal ranges. Specifically, we evaluate spatial position error and temporal velocity error under sequence lengths of $T = 243$ and $T = 351$. MPJPE measures the spatial reconstruction error, while MPJVE measures the discrepancy between predicted and ground-truth joint velocities.

Table 9: Spatial and temporal motion accuracy on Human3.6M under different sequence lengths. MPJPE is reported in mm and MPJVE in mm/frame. Lower is better.

| | $T = 243$ | | $T = 351$ | |
|---|---|---|---|---|
| Method | MPJPE↓ | MPJVE↓ | MPJPE↓ | MPJVE↓ |
| PoseMamba-S | 41.8 | 1.72 | 41.5 | 1.65 |
| SAMA-S | 40.6 | 1.66 | 40.3 | 1.59 |
| KAR-Mamba-S w/o T-Align | 40.0 | 1.66 | 39.8 | 1.58 |
| **KAR-Mamba-S** | **39.5** | **1.58** | **39.1** | **1.47** |

As shown in Table 9, KAR-Mamba consistently achieves lower MPJPE and MPJVE under both sequence lengths. When the sequence length increases, the advantage of the full model becomes more evident, while removing trajectory-aligned scanning leads to larger velocity errors. These results support the benefit of anchor-derived trajectory modeling for position and velocity accuracy over the evaluated sequence lengths.

### A.13 Does KAR-Mamba Remain Effective across Different Actions?

Average performance on the complete test set may hide the behavior of a model under different motion patterns. We therefore provide an action-wise evaluation on all Human3.6M actions to examine whether the benefit of prior-guided 3D kinematic modeling is consistent across diverse motion conditions.

As shown in Table 10, KAR-Mamba-S improves on PoseMamba-S across all evaluated actions, which involve different spatial structures and temporal dynamics. This pattern is consistent with the intended roles of skeletal and trajectory priors in pose refinement. The results support the effectiveness of the proposed kinematic modeling across the evaluated Human3.6M actions.

Table 10: Action-wise MPJPE (mm) on Human3.6M. Lower is better. The best result for each action is highlighted in bold.

| Method | Venue | $T$ | Dir. | Disc. | Eat | Greet | Phone | Photo | Pose | Pur. | Sit | SitD. | Smoke | Wait | WalkD. | Walk | WalkT. | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MHFormer | CVPR'22 | 351 | 39.2 | 43.1 | 40.1 | 40.9 | 44.9 | 51.2 | 40.6 | 41.3 | 53.5 | 60.3 | 43.7 | 41.1 | 43.8 | 29.8 | 30.6 | 43.0 |
| MixSTE | CVPR'22 | 243 | 37.6 | 40.9 | 37.3 | 39.7 | 42.3 | 49.9 | 40.1 | 39.8 | 51.7 | **55.0** | 42.1 | 39.8 | 41.0 | 27.9 | 27.9 | 40.9 |
| P-STMO | ECCV'22 | 243 | 38.9 | 42.7 | 40.4 | 41.1 | 45.6 | 49.7 | 40.9 | 39.9 | 55.5 | 59.4 | 44.9 | 42.2 | 42.7 | 29.4 | 29.4 | 42.8 |
| STCFormer | CVPR'23 | 243 | 39.6 | 41.6 | 37.4 | 38.8 | 43.1 | 51.1 | 39.1 | 39.7 | 51.4 | 57.4 | 41.8 | 38.5 | 40.7 | 27.1 | 28.6 | 41.0 |
| GLA-GCN | ICCV'23 | 243 | 41.3 | 44.3 | 40.8 | 41.8 | 45.9 | 54.1 | 42.1 | 41.5 | 57.8 | 62.9 | 45.0 | 42.8 | 45.9 | 29.4 | 29.9 | 44.4 |
| KTPFormer | CVPR'24 | 243 | 37.3 | 39.2 | 35.9 | 37.6 | 42.5 | 48.2 | 38.6 | 39.0 | 51.4 | 55.9 | 41.6 | 39.0 | 40.0 | 27.0 | **27.4** | 40.1 |
| PoseRetNet (Zheng et al., 2024) | ECCV'24 | 243 | 36.9 | 40.1 | 38.7 | 38.3 | 42.9 | 48.6 | 38.2 | 40.0 | 52.5 | 55.4 | 42.3 | 38.7 | 39.7 | **26.2** | 27.8 | 40.4 |
| PoseMamba-S | AAAI'25 | 243 | 39.5 | 41.6 | 39.9 | 35.7 | 43.5 | 52.7 | 40.2 | 37.9 | 53.1 | 63.4 | 42.4 | 40.0 | 39.6 | 29.3 | 29.2 | 41.8 |
| SasMamba | WACV'26 | 243 | 39.0 | 43.0 | 39.2 | 36.1 | 43.2 | 51.3 | 40.0 | 37.2 | 52.7 | 61.7 | 43.1 | 38.9 | 39.8 | 28.3 | 28.6 | 41.5 |
| **KAR-Mamba-S** | – | 243 | **36.5** | **39.1** | **35.8** | **34.8** | **41.8** | **48.0** | **37.4** | **37.1** | **50.8** | 58.1 | **41.5** | **38.3** | **38.5** | 27.3 | 27.5 | **39.5** |

## A.14 Qualitative Visualization and Failure Cases

We further examine the qualitative behavior of KAR-Mamba through representative predictions and failure cases. Fig. 11 compares the predicted poses with competing methods, while Fig. 12 illustrates representative challenging cases.

As shown in Fig. 11, KAR-Mamba produces predictions that more closely match the ground-truth skeletal configurations in the displayed examples, especially in articulated regions with large orientation changes. These examples illustrate the local reconstruction quality of the combined anchoring and refinement framework.

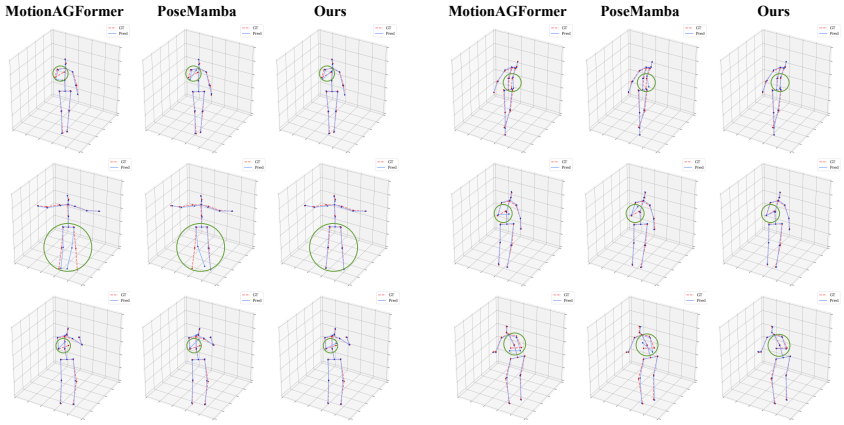



Figure 11: Qualitative comparison with MotionAGFormer and PoseMamba on representative test samples. Red dashed lines denote the ground-truth 3D poses and blue solid lines denote the predictions. Green circles highlight representative regions with noticeable differences in local pose reconstruction. KAR-Mamba shows closer alignment with the ground truth in the highlighted regions of these examples.

Despite these improvements, Fig. 12 shows that errors remain in rare and highly articulated configurations. In such cases, similar 2D observations may correspond to substantially different 3D relations, and the finite kinematic prototypes may provide insufficient coverage for these uncommon poses. These examples illustrate remaining limitations of prior-guided pose refinement.

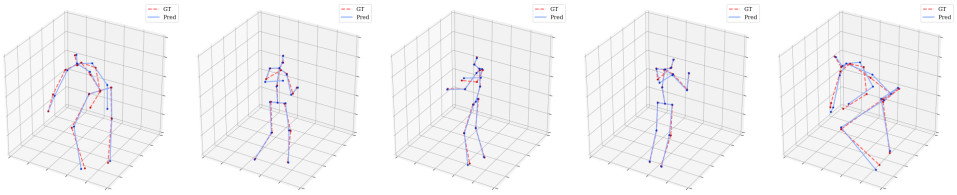


Figure 12: Representative failure cases of KAR-Mamba. Red dashed lines denote the ground-truth 3D poses and blue solid lines denote the predictions. Although the overall skeletal structures remain plausible, noticeable errors remain in the 3D orientations of highly articulated limbs, revealing remaining challenges for uncommon or extreme kinematic configurations.

### A.15 AI USAGE DISCLOSURE

During manuscript preparation, we used large language models to assist with language revision, LaTeX typesetting, and checks of the logical consistency of the presentation. The authors take full responsibility for the final content of this paper.