PocketSTPose: Causal Spatiotemporal Refinement for Mobile Sequential 3D Human Pose Estimation

Yuto Nakamizo[1], Rintaro Kimura[1], and Keiji Yanai[1]

The University of Electro-Communications, Chofu, Tokyo, Japan
{nakamizo-y,kimura-r,yanai}@mm.inf.uec.ac.jp

Abstract. In this paper, we propose PocketSTPose, a 3D human pose estimation framework designed for real-time streaming inference on mobile devices. While recent high-accuracy methods assume high-performance GPUs and fixed-length clip processing, mobile-oriented approaches are typically frame-wise and have limited capacity for spatiotemporal modeling. PocketSTPose adopts a two-stage design that decouples lightweight single-image estimation (Stage 1) and state-update-based causal refinement (Stage 2), enabling causal spatiotemporal integration under strict mobile constraints. These design choices enable low-latency causal inference on mobile GPUs while improving temporal consistency over a single-frame baseline.

Keywords: 3D Human Pose Estimation · Mobile Vision · Causal Spatiotemporal Modeling · Real-time Inference

1 Introduction

3D human pose estimation, which aims to predict 3D joint coordinates of a person from monocular videos or single images, is a fundamental technology for a wide range of applications such as VR/AR, sports analytics, surveillance, and rehabilitation [2,20,25]. Recent advances have realized more powerful spatiotemporal integration using Transformers [32], State Space Models (SSMs) [7,11], and graph-structured models, leading to substantial improvements in accuracy [3,6,12,23,27,31,35,36,38,39,41].

However, many of these high-accuracy methods assume high-performance GPUs and fixed-length clip processing, which is mismatched with the computational constraints and input conditions of real-time streaming inference on mobile devices. Existing lightweight models designed for mobile devices [4,25] mainly rely on per-frame estimation and do not integrate inter-joint relationships or temporal context within the model. As a result, challenges remain in terms of structural plausibility and temporal consistency.

In this work, we revisit the design of mobile 3D human pose estimation from the perspective of real-time streaming inference. Our key insight is that, under strict mobile latency constraints, allocating limited computational resources to

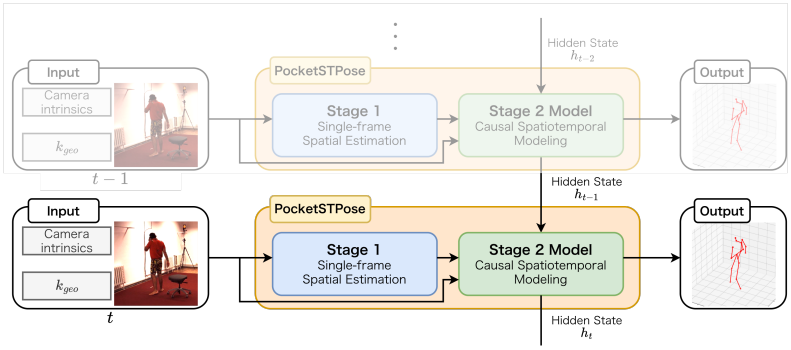


Fig. 1: Overview of the proposed framework, PocketSTPose. A lightweight single-frame estimator (Stage 1) first produces multi-hypothesis joint predictions. A causal refinement module (Stage 2) then incrementally integrates spatiotemporal information via cached hidden states, enabling real-time streaming inference on mobile devices. Note that the faded repeated process indicates past frames summarized in the hidden state.

causal spatiotemporal modeling is more beneficial than simply increasing the capacity of the single-frame spatial backbone. Motivated by this observation, we propose PocketSTPose, a two-stage architecture that reallocates computation from spatial processing to lightweight causal spatiotemporal refinement (Fig. 1). Our method consists of two stages: lightweight frame-wise spatial estimation (Stage 1) and spatiotemporal refinement (Stage 2) that incrementally corrects predictions while maintaining past information as a state. This design enables causal inference without relying on future frames, while achieving low-latency operation on mobile GPUs.

Experimental results show that, without using ground-truth absolute root joint depth, PocketSTPose achieves 50.4 mm MPJPE on Human3.6M Protocol 2 at 14.57 ms per frame on a Snapdragon 865 mobile GPU (FP16), fast enough for real-time 30 fps streaming, and improves temporal consistency over our single-frame (Stage 1-only) baseline.

Our contributions are summarized as follows:

- We propose a causal spatiotemporal modeling framework for 3D human pose estimation under mobile streaming inference, integrating inter-joint relationships and temporal context within the model.
- By combining lightweight single-image spatial estimation with a spatiotemporal module tailored to mobile GPUs, we enable low-latency and stable streaming inference.
- Through extensive experiments and on-device evaluation, we demonstrate a favorable accuracy–efficiency trade-off: our method attains accuracy comparable to the strongest ConvNeXtPose variants while requiring substantially fewer MACs and lower on-device GPU latency. We further verify stable real-time streaming inference on a mobile GPU and show improved temporal consistency over our single-frame (Stage 1-only) baseline.

2 Related Work

2.1 3D Human Pose Estimation

3D human pose estimation aims to predict 3D joint coordinates of a person from either videos or single images. Existing approaches are broadly categorized into methods that directly regress 3D joint coordinates from images [28, 30, 33] and methods that lift 2D joint coordinates to 3D [22, 26, 27, 31, 36–39, 41]. The former enables end-to-end optimization without explicitly separating intermediate representations, whereas the latter is compatible with leveraging large-scale 2D datasets and with temporal modeling based on low-dimensional representations. However, in the latter category, compressing inputs into 2D joint sequences often prevents semantic cues in the image space from propagating to downstream modules, and the resulting information loss has been pointed out [28].

Regression Strategies One-stage 3D human pose estimation typically employs either heatmap-based or direct regression. While heatmap methods preserve spatial prior knowledge, they suffer from high computational and memory-bandwidth costs due to high-resolution feature maps and large-scale Softmax operations. Conversely, direct regression is highly efficient but prone to losing local geometric details. To bridge this gap, Spatial-Aware Regression (SAR) [33] regresses offsets for local features, maintaining both spatial awareness and efficiency. Motivated by these trade-offs, our approach is designed to balance lightweight computation with precise information preservation.

Spatiotemporal Integration Leveraging spatiotemporal information is effective for improving the accuracy of 3D human pose estimation, and it has been widely adopted especially in two-stage methods. Transformer-based approaches such as the PoseFormer family, MixSTE, and MotionBERT demonstrate strong performance. By modeling inter-joint and temporal relationships, these methods improve accuracy; however, most of them assume offline processing with fixed-length clips and are thus not well suited for low-latency streaming inference on sequentially arriving frames.

Recently, State Space Models (SSMs) [7, 11] have attracted attention as they enable efficient sequential updates and can model long-range dependencies with linear complexity in sequence length. In particular, S6 [11] and Mamba [11], the latter being an architecture built upon S6, introduce input-dependent state updates and have increasingly been applied to 3D human pose estimation [5, 6, 12]. However, many existing applications assume bidirectional processing or offline configurations, and designs that primarily target causal streaming inference under mobile constraints remain underexplored.

Causal/online 3D pose is itself established—e.g., the causal-convolution variant of VideoPose3D [26] enables online inference without future frames. Our contribution is thus not causal inference per se, but its combination with an explicit mobile latency budget, which prior causal methods do not target. Throughout, "causal" denotes temporal causality (no future frames), not causal inference.

2.2 3D Human Pose Estimation for Mobile Devices

Representative studies targeting real-time 3D human pose estimation on mobile devices include MobileHumanPose [4] and ConvNeXtPose [25]. Looking further back, BlazePose [2] and MoVNect [13] also focus on real-time pose estimation in mobile settings, where 3D estimation is treated as a target task or discussed as a possible extension. While these methods achieve low-latency inference with lightweight backbones, their primary focus is improving the efficiency of single-image estimation, leaving limited room for incorporating spatiotemporal modeling. In particular, recent methods [4, 25] employ integral regression [30], which requires large-scale Softmax normalization and becomes a latency bottleneck on mobile GPUs, so that under real-time constraints (e.g., 30 fps) single-image processing already consumes most of the per-frame latency budget, leaving little room for streaming spatiotemporal modeling.

Therefore, a framework that simultaneously achieves causal spatiotemporal modeling and real-time operation on real mobile devices has not yet been well established. We bridge this gap by reducing the cost of single-image estimation to free up the latency budget and reallocating it to streaming spatiotemporal modeling.

3 Methods

3.1 Overview

We propose a two-stage architecture, PocketSTPose, to model inter-joint and temporal relationships on mobile devices (i.e., to learn dependencies over a token sequence of joints×frames). PocketSTPose decouples frame-wise spatial estimation (Stage 1), which projects image information into joint tokens, from spatiotemporal modeling (Stage 2), which causally integrates information on the joint tokens (Fig. 1).

Stage 1 estimates the 3D coordinates, uncertainty, and aggregated features for each joint from the input frame, thereby projecting image evidence into joint tokens with geometric semantics. Stage 2 takes the outputs of Stage 1 (joint estimates, confidences, uncertainty representations, and image features) as input, and performs spatiotemporal integration under causal constraints on joint tokens to correct jitter inherent to single-image estimation and failures under occlusions.

In addition, we introduce absolute root joint depth estimation, which has often been ignored in recent lightweight mobile-oriented models, and output root-relative 3D joint coordinates consistent with the camera coordinate system.

Notation. Throughout this section, t and j denote the frame and joint indices, respectively, where J is the number of joints. The input frame at time t is denoted $\mathbf{I}_t \in \mathbb{R}^{H_{\text{img}} \times W_{\text{img}} \times 3}$, where H_{img} and W_{img} are the input image height and width. The backbone produces a feature map of spatial size $H \times W$ with C channels, where each of the HW spatial locations is referred to as a grid (indexed by g). Stage 1 outputs K per-joint hypotheses to represent prediction uncertainty,

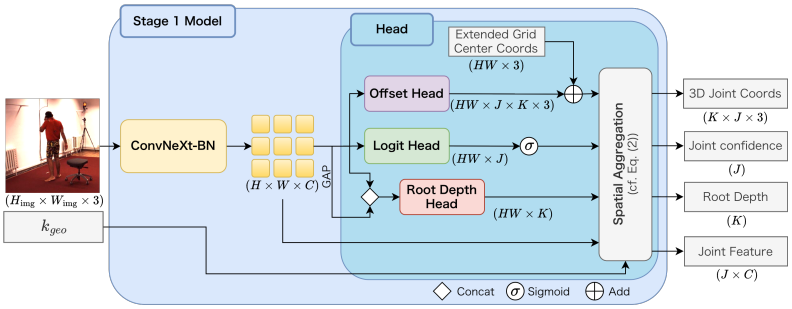


Fig. 2: Architecture of Stage 1. Given a single image, the model extracts backbone features and predicts multi-hypothesis 3D joints, absolute root joint depth, and confidence/features, via confidence-weighted aggregation.

and k_{geo} denotes the camera-dependent normalization factor used for absolute depth (defined in Sec. 3.2). Stage 2 stacks L spatiotemporal blocks. We use bold symbols for vectors/tensors (e.g., $\mathbf{p}$) and plain symbols for scalars (e.g., c).

3.2 Stage 1: Frame-wise Spatial Estimation

In Stage 1, we perform spatial estimation via a multi-hypothesis 3D joint estimation task that preserves uncertainty, using a direct regression design without generating high-resolution heatmaps. Specifically, given an input image $\mathbf{I}_t$, we simultaneously predict K hypotheses of 3D joint coordinates, K hypotheses of absolute root joint depth, and confidence/features used by the subsequent stage. The architecture of Stage 1 is illustrated in Fig. 2.

Backbone and Low-Resolution Design We adopt a BatchNorm variant of ConvNeXt (ConvNeXt-BN) backbone based on ConvNeXtPose [25], and remove upsampling layers to directly use a low-resolution feature map $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$.

Multi-hypothesis Inference with DistPred Since single-image 3D estimation inherently involves uncertainty, Stage 1 preserves predictions as a set of hypotheses rather than compressing them into a point estimate. Concretely, for each joint we output a set of K hypotheses $\{\hat{\mathbf{P}}_j'^{(k)}\}_{k=1}^{K}$, and optimize both an accuracy term and an inter-hypothesis diversity term using a CRPS-approximation loss based on DistPred [18]. Unless otherwise noted, we use the same fixed number of hypotheses K at training and inference.

SAR-based Regression and Confidence-weighted Aggregation Instead of dense heatmap estimation, we adopt direct regression based on SAR [33] operating on low-resolution features. This avoids normalization bottlenecks typical on mobile GPUs, preserving the latency budget for the spatiotemporal modeling in Stage 2. To optimize for mobile deployment, we modify the original SAR formulation in two key ways. First, we replace its discrete max-confidence selection (argmax/gather) with a continuous confidence-weighted aggregation. High-dimensional indexing is often inefficient or unsupported by mobile inference runtimes (e.g., TFLite GPU delegate), whereas continuous aggregation ensures stable, low-latency execution. Second, we bypass voxelization to avoid the overhead of 3D tensor construction. Instead, we regress 3D offsets directly from 2D

grids, deferring the resolution of depth ambiguity to the causal spatiotemporal refinement in Stage 2. Finally, the regression head is implemented as an MLP to support multi-task prediction.

Accordingly, combined with the multi-hypothesis inference above, the estimation of 3D joint candidates and confidences by this head is formulated as follows. From each grid feature $\mathbf{f}_g$ ($g \in \{1, \ldots, HW\}$), we regress a confidence logit $s_{j,g} \in \mathbb{R}$ (a scalar score indicating how reliable grid g is for joint j) and K hypotheses of 3D offsets $\Delta\mathbf{P}'^{(k)}_{j,g} \in \mathbb{R}^3$ for each joint j using an MLP. Let the grid center coordinate be $\mathbf{G}_g$ (with z extended as 0) and $\sigma(\cdot)$ be the sigmoid function. We obtain the grid-derived joint hypotheses $\mathbf{P}'^{(k)}_{j,g}$ and the per-grid confidence $c_{j,g} \in [0, 1]$ as

$$\mathbf{P}'^{(k)}_{j,g} = \mathbf{G}_g + \Delta\mathbf{P}'^{(k)}_{j,g}, \quad c_{j,g} = \sigma(s_{j,g}). \tag{1}$$

At inference time, we aggregate them spatially via a confidence-weighted average:

$$w_{j,g} = \frac{c_{j,g}}{\sum_g c_{j,g}}, \quad \hat{\mathbf{P}}'^{(k)}_j = \Psi\left(\sum_g w_{j,g}\mathbf{P}'^{(k)}_{j,g}\right), \tag{2}$$

where $w_{j,g}$ is the normalized aggregation weight for grid g, and $\hat{\mathbf{P}}'^{(k)}_j$ is the resulting aggregated k-th hypothesis for joint j. Here $\Psi(\cdot)$ denotes an inverse normalization function that maps a normalized prediction back to the metric coordinate space: it linearly restores the normalized image coordinates $(x, y) \in [0, 1]^2$ to the pixel coordinate system, and restores the normalized depth z to a metric root-relative depth based on the depth range D_{range}. Note that the confidence c and the grid features $\mathbf{f}$ are also aggregated in the same manner using $w_{j,g}$, yielding the per-joint aggregated confidence $\hat{c}_j$ and feature $\hat{\mathbf{f}}_j$ that are passed to Stage 2.

Absolute Root Joint Depth Estimation For the absolute root joint depth d_{root}, we also estimate K hypotheses $d'^{(k)}_{\text{root},g}$ from each grid using a dedicated MLP. Since depth estimation tends to depend on global information, we concatenate a global feature $\mathbf{f}_{\text{global}}$ obtained by Global Average Pooling and predict depth with an MLP. Following RootNet [24], we introduce a geometric normalization factor k_{geo}, which is computed from the camera focal length and the area of the person bounding box and encodes the camera-to-person scale prior used to make absolute depth regression camera-agnostic. For inference-time aggregation, we do not design a separate confidence for depth; instead, we reuse the root-joint confidence $s_{\text{root},g}$. Furthermore, to avoid inter-task interference, we stop gradient propagation to the shared trunk for the depth head and its corresponding logit pathway.

3.3 Stage 2: Spatiotemporal Refinement

Stage 2 integrates (i) higher-order inter-joint relationship modeling based on a hypergraph and (ii) causal temporal modeling based on SSMs, in a manner compatible with streaming inference, and estimates camera-coordinate root-relative 3D joint coordinates $\mathbf{p}^{\text{ref}}_t \in \mathbb{R}^{J\times 3}$. The architecture of Stage 2 is shown in Fig. 3.

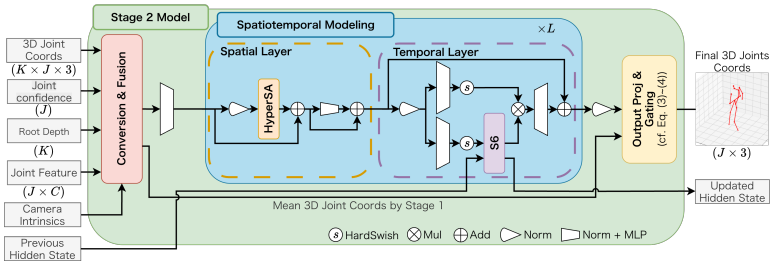


Fig. 3: Architecture of Stage 2. Alternating HyperSA-based spatial modeling and SSM-based causal temporal modeling refine joints with a cached hidden state.

Input Representation To perform spatiotemporal modeling while maintaining geometric consistency, Stage 2 first transforms the estimates from Stage 1 into the camera coordinate system. Specifically, for $\hat{\mathbf{P}}'^{(k)}_{t,j}$ and $\hat{d}^{(k)}_{t,\mathrm{root}}$ obtained from Stage 1, we apply back-projection Π^{-1} based on camera intrinsics to obtain absolute coordinates $\mathbf{P}^{(k)}_{t,j}$. We then subtract the root joint to obtain root-relative coordinates $\mathbf{p}^{(k)}_{t,j}$. This transformation avoids mixing image-plane and depth coordinate systems, and allows geometric constraints such as bone lengths and velocities to be handled naturally in a consistent coordinate system.

Since single-image estimation inherently contains uncertainty, we propagate the K-hypothesis distribution obtained in Stage 1 to the next stage without collapsing it. Concretely, we preserve the distribution shape using the hypothesis mean $\bar{\mathbf{p}}_{t,j}$ and the deviations from the mean $\boldsymbol{\Delta}^{(k)}_{t,j}$. In addition, we append the estimated confidence $\hat{c}_{t,j}$ and aggregated feature $\tilde{\mathbf{f}}_{t,j}$ so that not only distributional shape but also image-derived semantic information is transmitted. We construct the Stage 2 input token $\mathbf{v}_{t,j}$ by concatenating these four components.

Spatiotemporal Blocks Following MixSTE [36], we alternately apply spatial modeling and temporal modeling with L layers. In the spatial layer, we model the within-frame joint tokens $\{\mathbf{v}_{t,j}\}_{j=1}^{J}$ using a Transformer-style attention mechanism with an MLP [32]. As the attention mechanism, we adopt HyperSA [40] to encode higher-order skeletal relationships based on hyperedges. In this work, we use a fixed hypergraph (rather than a learnable one) to prioritize computational efficiency and stability during streaming inference. In the temporal layer, we perform causal temporal modeling on the per-joint temporal tokens $\{\mathbf{v}_{\tau,j}\}_{\tau\le t}$ using an SSM-based state update module with the S6 mechanism following Mamba [11]. Here, S6 refers to the selective state-space mechanism introduced in Mamba, in which the state-space parameters are made input-dependent so that the model can selectively propagate or forget information through a fixed-size recurrent hidden state, enabling causal sequential updates with linear complexity. To reduce cache size during streaming inference, we remove the causal 1D convolution and channel expansion components of Mamba and conduct temporal modeling solely via state updates.

Gated Output From the final features of Stage 2, a linear layer predicts a residual correction $\Delta\mathbf{p}^{\mathrm{res}}_{t,j}$, a reconstructed coordinate $\mathbf{p}^{\mathrm{new}}_{t,j}$, and a gate logit

$g_{t,j}^{\text{logit}}$. We then obtain the gate coefficient $\alpha_{t,j}$ via a sigmoid function and mix the residual and reconstruction pathways per joint. The refined coordinate $\mathbf{p}_{t,j}^{\text{ref}}$ is defined as:

$$\alpha_{t,j} = \sigma(g_{t,j}^{\text{logit}}), \tag{3}$$

$$\mathbf{p}_{t,j}^{\text{ref}} = (1 - \alpha_{t,j}) \cdot (\bar{\mathbf{p}}_{t,j} + \Delta\mathbf{p}_{t,j}^{\text{res}}) + \alpha_{t,j} \cdot \mathbf{p}_{t,j}^{\text{new}}. \tag{4}$$

Using only residual correction may be insufficient when the initial prediction of Stage 1 is severely inaccurate. Therefore, we additionally use the reconstruction pathway so that, when a global repositioning from the initial estimate is needed, $\alpha_{t,j}$ can favor $\mathbf{p}_{t,j}^{\text{new}}$. Here $\bar{\mathbf{p}}_{t,j}$ is the per-joint hypothesis mean from Stage 1 (defined in Sec. 3.3, Input Representation), so the residual pathway corrects this mean while the reconstruction pathway can replace it entirely.

3.4 Training Strategy and Losses

To prevent destabilizing Stage 1 representations via gradients from Stage 2, we train the model in two steps: we first train Stage 1 on image data, and then freeze Stage 1 and train Stage 2 on video data. For Stage 1 training, we additionally use data annotated only with 2D labels by excluding depth-related losses; for Stage 2 training, we use only data with 3D annotations.

Losses for Stage 1. For joint coordinates, we compute a regression loss for the mean estimate and a CRPS-approximation loss from DistPred [18] for each grid, and optimize their confidence-weighted spatial aggregation (aggregation weighted by the normalized $c_{j,g} = \sigma(s_{j,g})$) as the loss aggregated over grids. Note that this loss is computed in the coordinate space where X, Y are represented as normalized image coordinates and Z as normalized relative depth. For absolute root joint depth, we similarly introduce both a mean loss and a distributional loss, aggregate them with confidence weights, and define the total loss as a weighted sum. The final Stage 1 loss is given as a weighted sum of the joint-coordinate loss and the absolute root joint depth loss.

Losses for Stage 2. We train Stage 2 on root-relative 3D joint coordinates in the camera coordinate system. In addition to a regression error (weighted MPJPE) and a temporal smoothness regularization (following KTPFormer [27]), we introduce bone-length matching to the ground truth and bone-length consistency in input sequence, and define the total loss as a weighted sum of these terms.

4 Experiments

4.1 Experimental Setup

Data and Preprocessing We mainly use Human3.6M [14] for training and evaluation, and additionally leverage MPII [1] to improve the generalization of the single-frame model. For all datasets, we generate a person bounding box

from the provided 2D joint coordinates and crop the person region as input. For joint definitions, we use an 18-joint representation (adding a thorax joint) for training and inference; evaluation is performed with 17 joints by excluding the additional joint. For MPII [1] and 3DPW [21], we apply a joint-mapping to match the Human3.6M joint definition following prior work [16, 25].

Evaluation Protocols On Human3.6M, following ConvNeXtPose, we evaluate under Protocol 2 which trains on subjects S1, S5, S6, S7, and S8 and evaluates on S9 and S11. Unless otherwise noted, we apply test-time augmentation via horizontal flipping of the input images, following prior work [23, 25, 38].

Metrics We primarily report Mean Per Joint Position Error (MPJPE). As an auxiliary metric for temporal consistency, we also report bone-length consistency. For the same person in a video, we report the mean variance and coefficient of variation of bone lengths over bones and time.

In addition, we use the Energy Score [8] to assess the validity of multi-hypothesis estimation in Stage 1. To reduce bias with respect to the number of samples K, we adopt an unbiased form that excludes self-distances, and normalize the pairwise distance term by $K(K-1)$.

On-device Latency Measurement We measure inference latency on a Samsung Galaxy S20 (Snapdragon 865) to reflect real deployment conditions. All models are converted to FP32 TensorFlow Lite format and evaluated on CPU and GPU delegate (FP16 allowed). We report the mean latency and standard deviation measured by the LiteRT Benchmark Tool [10]. Importantly, the reported latency corresponds to per-frame streaming inference time on-device and reflects the computational cost of processing a single frame.

Implementation Details In Stage 1, we employ a ConvNeXt-BN backbone derived from the ConvNeXtPose-XS architecture, with an input resolution of 256×256 and the number of hypotheses $K = 100$. Stage 2 is configured with $L = 4$ layers, an embedding dimension of 256, and 8 attention heads. Furthermore, the expansion ratio of the Transformer's MLP is set to 2, while the S6 state size is maintained at 16.

For accuracy evaluation, we use clips of $T = 30$ frames sampled at 10 fps due to dataset constraints, computing metrics on the current frame while using the preceding 29 frames to build the causal state, whereas latency is measured for per-frame causal streaming inference with cached states (i.e., not scaling with T).

Further details on preprocessing, metric definitions, training hyperparameters, and on-device latency measurement are provided in the supplementary material.

4.2 Comparison with Prior Methods

Comparison with Mobile-oriented 3D Human Pose Estimation Models We compare our method with three representative mobile 3D human pose estimation models, MoVNect [13], MobileHumanPose [4] and ConvNeXtPose [25]. Table 1 reports the accuracy, on-device latency, computational cost and model

Table 1: Quantitative comparison and computational cost on Human3.6M (Protocol 2). We report MPJPE (mm), on-device GPU latency (ms), MACs (G), and parameters (M). Latency for MoVNect and MobileHumanPose (†) is cited from [4] and excludes Softmax; ConvNeXtPose and PocketSTPose are measured on a Galaxy S20 (Snapdragon 865) with Softmax included. Cited and measured latencies are therefore not directly comparable.

Model	MPJPE(mm) ↓	CPU(ms)	GPU(ms)	MACs(G)	Params(M)
w/ GT absolute root joint depth					
MoVNect [13]	97.3	22.0 ± 5.0 †	25.2 ± 4.9 †	**0.68**	**1.03**
MobileHumanPose-XS [4]	69.6	**12.2 ± 4.2** †	**13.0 ± 4.0** †	1.96	2.24
MobileHumanPose-L [4]	51.4	108.0 ± 6.9 †	109.0 ± 3.2 †	2.75	4.07
ConvNeXtPose-XS [25]	56.6	**15.84 ± 0.64**	25.99 ± 0.39	0.82	3.53
ConvNeXtPose-S [25]	51.8	33.64 ± 1.44	31.73 ± 0.41	1.76	7.44
ConvNeXtPose-M [25]	51.1	65.77 ± 3.45	2104.88 ± 10.93	2.82	7.59
ConvNeXtPose-L [25]	**49.8**	93.63 ± 4.93	2151.13 ± 8.76	4.30	8.38
PocketSTPose (Ours)	49.9	26.11 ± 0.95	**13.49 ± 0.02**	1.02	10.69
w/o GT absolute root joint depth					
PocketSTPose (Ours)	50.4	28.15 ± 1.40	14.57 ± 0.02	1.09	11.64

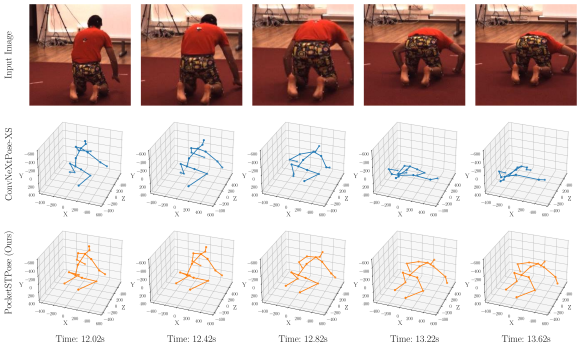


Fig. 4: Qualitative comparison with ConvNeXtPose-XS on Human3.6M.

parameter size. In this evaluation, following ConvNeXtPose, Human3.6M Protocol 2 evaluates frames sampled every 64 frames.

Latencies of MobileHumanPose and MoVNect are cited from [4] and exclude Softmax, whereas ConvNeXtPose is measured with its official implementation including Softmax. Ours is measured per streaming step including cache I/O (protocol in the supplementary material). Since ConvNeXtPose uses GT absolute root joint depth at evaluation, we also report ours under that setting.

As shown in Tab. 1, our method achieves accuracy (MPJPE) close to larger ConvNeXtPose-M/L models with substantially fewer computations. Under the matched setting using GT absolute root joint depth, our method attains comparable MPJPE with only about 24% of the MACs of ConvNeXtPose-L. Moreover, even under the more practical setting without GT absolute root joint depth, our method matches or exceeds ConvNeXtPose-M and MobileHumanPose-L in accuracy while using only about 40% of their MACs. These results indicate that our contribution lies not in a lightweight design alone, but primarily in a favorable accuracy–efficiency balance achieved through computation reallocation.

PocketSTPose has the largest parameter count (11.64M), but this is practically negligible; on mobile GPUs latency is governed by MACs and bandwidth-heavy operations rather than parameter count. Note that except for RootNet-based approaches [24], prior mobile methods are evaluated only with GT absolute root joint depth; PocketSTPose is one of the few that also operates without it (the "w/o" GT row).

Tab. 1 also shows a discontinuous GPU latency degradation for ConvNeXtPose-M/L (exceeding 2000 ms), which is not observed on CPU and thus cannot be attributed to MACs alone. This behavior stems from large-scale Softmax normalization in their higher-resolution heatmaps: as a diagnostic, removing it drops ConvNeXtPose-M to 17.80 ms.

ConvNeXtPose computes sub-pixel expectations via integral regression [30] and thus cannot omit Softmax. For a like-for-like comparison, the relevant baselines are ConvNeXtPose-XS/S, which avoid this delegate pathology; against these, our direct-regression design still reduces GPU latency from 25.99/31.73 ms to 14.57 ms, and stays stable even without GT absolute root joint depth. The over-2000 ms M/L latencies are reported as a diagnosed delegate issue, not as the basis of our latency claim. Note that although MoVNect has the smallest MACs and parameters and MobileHumanPose-XS indicates the lowest latency, their accuracies, 97.3 and 69.6, are much inferior to ours and the other baselines.

Qualitative Comparison Fig. 4 visualizes predictions on multiple frames sampled every 0.4 s from the same Human3.6M video, arranged horizontally. As shown in Fig. 4, ConvNeXtPose-XS, which estimates each frame independently, exhibits inconsistent bone lengths. In contrast, our method maintains consistent joint configurations across frames. In addition, in the rightmost two examples under severe occlusions, ConvNeXtPose-XS shows severe structural failure, whereas such failures are not observed in our results.

Comparison with High-accuracy 3D Human Pose Estimation Models We further compare our method with recent high-accuracy 3D human pose estimation models designed for high-performance GPUs without computational constraints, where we evaluate all frames on Human3.6M Protocol 2 following prior work. Tab. 2 reports the exact accuracy and computational cost (a semi-log compute–accuracy plot is provided in the supplementary material). Unlike most of these methods, which rely on an external 2D pose estimator (e.g., Stacked Hourglass or CPN) followed by a 2D-to-3D lifting network, our method operates directly on the image and requires no separate 2D backbone. We compute computational cost assuming per-frame inference for sequentially arriving frames in real-time deployment. For two-stage methods, we sum the costs of 2D pose estimation and 2D-to-3D lifting to reflect the end-to-end system cost.

As shown in Tab. 2, these high-accuracy methods attain lower MPJPE (typically 37–45 mm) but at a computational cost one to two orders of magnitude larger than ours: even the most efficient lifting-based competitors exceed 14 GMACs once their 2D pose estimator is included, whereas our method requires only 1.09 GMACs. Our method thus sits at the extreme low-compute end of the compute–accuracy frontier (visualized in the supplementary material), trading a

Table 2: Comparison with representative high-accuracy 3D pose estimation models on Human3.6M Protocol 2 (accuracy and computational cost). "2D Pose Model" denotes the external 2D estimator each method depends on; our method requires none, and its MACs include 2D estimation and lifting. The full table covering all model sizes is provided in the supplementary material.

Method	MPJPE (mm) ↓	MACs (G) ↓	2D Pose Model
MotionBERT [41]	**37.5**	200.7	Stacked Hourglass
PoseMamba-L [12]	38.1	53.8	Stacked Hourglass
MotionAGFormer-L [23]	38.4	104.2	Stacked Hourglass
SasMamba [6]	41.5	27.2	Stacked Hourglass
HDFormer [3]	42.6	14.5	CPN
MixSTE [36]	40.9	152.9	CPN
AITEPose-ResNet50 [34]	48.7	5.4	None
Ours	50.5†	**1.09**	None

(† Evaluated on all frames; cf. 50.4mm with every-64-frame sampling in Table 1.)

Table 3: Quantitative results on 3DPW (MPJPE on 14 joints, valid-camera frames only). "Zero": zero-shot, no 3DPW fine-tuning; "FT": fine-tuned on 3DPW. High-accuracy mesh-based models are shown for reference; they are not designed for mobile streaming inference.

Method	3DPW MPJPE (mm) ↓
High-accuracy models	
HMR [15] (Zero)	130.0
SPIN [17] (Zero)	96.9
VIBE [16] (Zero)	93.5
HMR2.0a [9] (Zero)	70.0

Method	3DPW MPJPE (mm) ↓
Mobile-oriented models	
Liu *et al.* [19] (Zero)	126.2
Stage 1 (Ours) (Zero)	299.34
Stage 1 (Ours) (FT)	86.71
PocketSTPose (Ours) (Zero)	134.44
PocketSTPose (Ours) (FT)	**81.28**

moderate increase in MPJPE for a roughly 13×–180× reduction in computation. We emphasize that this comparison is not intended to claim state-of-the-art accuracy; these methods target offline clip processing on high-performance GPUs and rely on an external 2D backbone whose cost alone exceeds our entire pipeline. Rather, it shows that PocketSTPose remains within a practically useful accuracy range under a compute budget compatible with real-time on-device operation.

4.3 Generalization to In-the-wild Data (3DPW)

Because PocketSTPose is motivated by real-world mobile deployment, we additionally evaluate on 3DPW [21], an in-the-wild benchmark captured with a moving camera, to assess generalization beyond the controlled indoor setting of Human3.6M. Following the common protocol used in 3D human mesh reconstruction works such as VIBE [16], we report MPJPE on 14 evaluation joints and evaluate only on frames with valid camera parameters. Joint annotations are converted to our target joint definition with the same name-based matching used for MPII. In this experiment, we do not apply test-time augmentation. "Zero" denotes zero-shot evaluation (trained only on Human3.6M and MPII), while "FT" denotes fine-tuning on the 3DPW training split. To examine the generalization of each stage, we also report Stage 1-only results. Further training details are provided in the supplementary material.

As shown in Tab. 3, although zero-shot accuracy remains limited (134.44 mm), PocketSTPose generalizes reasonably to in-the-wild data despite using

Table 4: Ablations on Stage 2 (Human3.6M Protocol 2).

	Settings				Eval Results		
Model	ST Modeling	Gating	FB	DB	MPJPE (mm) ↓	BL_{var} (mm^2) ↓	BL_{cv} ↓
Ours	HyperSA/S6	✓	✓	✓	**50.39**	25.81	1.87
w/o Gating	HyperSA/S6		✓	✓	51.45	24.91	1.87
w/o FB	HyperSA/S6	✓		✓	50.76	27.59	1.95
w/o DB	HyperSA/S6	✓	✓		52.11	24.95	1.85
w/o HyperSA	SA/S6	✓	✓	✓	51.77	37.94	2.26
w/o S6	HyperSA/SA	✓	✓	✓	51.72	**21.76**	**1.73**

only indoor training data in the zero-shot setting. The effect of the proposed spatiotemporal refinement is most visible here: while the zero-shot Stage 1 estimate is very poor (299.34 mm), Stage 2 refinement reduces the error to 134.44 mm, and the gap persists after fine-tuning (Stage 1: 86.71 mm vs. full model: 81.28 mm), indicating that causal temporal integration helps stabilize predictions under domain shift. The dominant failure mode of the zero-shot model is absolute root-joint depth rather than relative pose: replacing the predicted root depth with the ground-truth value improves the zero-shot Stage 1 MPJPE from 299.34 mm to 120.87 mm, consistent with Human3.6M being a fixed-camera indoor dataset (detailed depth-error analysis is given in the supplementary material). Overall, these results provide cross-dataset evidence, beyond the controlled Human3.6M benchmark, that the proposed temporal modeling is effective under realistic in-the-wild conditions.

4.4 Ablation Studies

In this section, we analyze the contribution of each component in PocketSTPose on Human3.6M Protocol 2, using frames sampled every 64 frames. We ablate design choices in Stage 1/Stage 2 as well as spatiotemporal integration itself.

Ablations on Stage 1 To exclude the influence of downstream modeling, we evaluate using outputs from Stage 1 only. We ablate the head structure (Linear/MLP), gradient stopping (Detach), the loss configuration ($\mathcal{L}_{\text{SAR}}$, $\mathcal{L}_{\text{Dist}}$), and the number of samples K; the full table and the Stage 1-only inference procedure are provided in the supplementary material. In summary, combining the proposed MLP head with gradient stopping achieves the best MPJPE, and the combined use of $\mathcal{L}_{\text{SAR}}$ and $\mathcal{L}_{\text{Dist}}$ is favorable in terms of both MPJPE and Energy Score. Increasing K improves the Energy Score but tends to degrade MPJPE; we therefore set $K = 100$ as a trade-off between the two.

Ablations on Stage 2 Fixing Stage 1 to the proposed configuration, Tab. 4 compares accuracy and temporal consistency by varying the spatial and temporal modeling mechanisms in Stage 2, the gating mechanism, feature bypass (FB), and distribution bypass (DB). When replacing the temporal modeling mechanism with Self-Attention, we introduce MQA [29] and KV-cache to support streaming inference and maintain latency, and use a standard Transformer block structure.

As shown in Tab. 4, all components improve MPJPE. HyperSA and FB also reduce BLvar and BLcv, with HyperSA giving the largest reduction, whereas

Table 5: Latency-matched comparison between an expanded Stage 1 model and our method (Human3.6M Protocol 2).

Model	MPJPE (mm) ↓	BL_{var} (mm^2)↓	BL_{cv} ↓	Mobile latency GPU FP16 (ms) ↓
w/ GT absolute root joint depth				
Expanded Stage 1 model	51.20	196.27	4.49	15.33 ± 0.04
Ours	**49.93**	**41.97**	**2.31**	$\mathbf{13.49 \pm 0.02}$
w/o GT absolute root joint depth				
Expanded Stage 1 model	55.87	209.17	4.77	16.40 ± 0.06
Ours	**50.39**	**25.81**	**1.87**	$\mathbf{14.57 \pm 0.02}$

gating and DB slightly increase BLvar (e.g., 25.81 vs. 24.91 mm^2 w/o gating); we accept this small trade-off for their MPJPE gains. Introducing S6 yields better MPJPE than standard Self-Attention but slightly degrades these bone-length metrics: Self-Attention directly aggregates features across time steps, whereas SSM propagates information through a compressed hidden state, which may weaken geometric regularity. We prioritize final coordinate accuracy and thus adopt S6.

Effectiveness of Spatiotemporal Modeling (Latency-matched Comparison) To verify the effectiveness of spatiotemporal modeling itself, we compare our method with a model that expands Stage 1 to achieve comparable latency. Tab. 5 reports accuracy and temporal consistency.

As shown in Tab. 5, regardless of whether GT absolute root joint depth is used, our method consistently improves both MPJPE and bone-length temporal consistency (BL_{var}, BL_{cv}) over the expanded Stage 1 model within a comparable latency regime. The substantial improvement in temporal consistency indicates that spatiotemporal integration suppresses frame-to-frame jitter and structural failures that cannot be resolved simply by increasing single-frame capacity, demonstrating that adding spatiotemporal integration is beneficial under limited computational resources.

5 Conclusion

We proposed PocketSTPose, a causal framework for real-time streaming 3D human pose estimation on mobile devices. By decoupling single-frame estimation and state-update-based refinement, it integrates spatiotemporal dependencies under strict mobile constraints, achieving 14.57 ms per frame and 50.4 mm MPJPE on Human3.6M Protocol 2. Several evaluations remain future work: multi-seed accuracy variance, comparison against online-restricted lifting baselines under matched causal windows, temporal-consistency metrics for external streaming methods, accuracy under 30 fps input sampling, and multi-device latency. Extending Stage 2 to absolute-coordinate refinement while preserving real-time performance is also left for future work.

Acknowledgements This work was supported by JST CRONOS JPMJCS24K4 and JSPS KAKENHI 26K23875.

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 3686–3693 (2014) 8, 9
2. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., Grundmann, M.: BlazePose: On-device Real-time Body Pose Tracking. In: CVPR Workshop on Computer Vision for Augmented and Virtual Reality (2020) 1, 4
3. Chen, H., He, J.Y., Xiang, W., Cheng, Z.Q., Liu, W., Liu, H., Luo, B., Geng, Y., Xie, X.: HDFormer: High-order Directed Transformer for 3D Human Pose Estimation. In: Proc. of the Thirty-Second International Joint Conference on Artificial Intelligence. pp. 581–589 (2023) 1, 12
4. Choi, S., Choi, S., Kim, C.: MobileHumanPose: Toward real-time 3D human pose estimation in mobile devices. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 2328–2338 (2021) 1, 4, 9, 10
5. Cui, H., Hayama, T.: HGMamba: Enhancing 3D Human Pose Estimation with a HyperGCN-Mamba Network. In: Proc. of International Joint Conference on Neural Networks (2025) 3
6. Cui, H., Hua, W., Huang, R., Jia, S., Hayama, T.: SasMamba: A Lightweight Structure-Aware Stride State Space Model for 3D Human Pose Estimation. In: Proc. of IEEE Winter Conference on Applications of Computer Vision (2026) 1, 3, 12
7. Dao, T., Fu, D.Y., Saab, K.K., Thomas, A.W., Rudra, A., Re, C.: Hungry Hungry Hippos: Towards Language Modeling with State Space Models. In: Proc. of International Conference on Learning Representations (2023) 1, 3
8. Gneiting, T., Raftery, A.E.: Strictly Proper Scoring Rules, Prediction, and Estimation. Journal of the American Statistical Association **102**(477), 359–378 (2007) 9
9. Goel, S., Pavlakos, G., Rajasegaran, J., Kanazawa, A., Malik, J.: Humans in 4D: Reconstructing and Tracking Humans with Transformers. In: Proc. of IEEE International Conference on Computer Vision. pp. 14737–14748 (2023) 12
10. Google: Benchmark Interpreter API. https://ai.google.dev/edge/litert/models/measurement (2026), accessed: 2026-07-06 9
11. Gu, A., Dao, T.: Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 (2023) 1, 3, 7
12. Huang, Y., Liu, J., Xian, K., Qiu, R.C.: PoseMamba: monocular 3d human pose estimation with bidirectional global-local spatio-temporal state space model. In: Proc. of AAAI Conference on Artificial Intelligence. pp. 3842–3850 (2025) 1, 3, 12
13. Hwang, D.H., Kim, S., Monet, N., Koike, H., Bae, S.: Lightweight 3D Human Pose Estimation Network Training Using Teacher-Student Learning. In: Proc. of IEEE Winter Conference on Applications of Computer Vision. pp. 468–477 (2020) 4, 9, 10
14. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. Proc. of IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(7), 1325–1339 (2014) 8
15. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-End Recovery of Human Shape and Pose. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7122–7131 (2018) 12

16. Kocabas, M., Athanasiou, N., Black, M.J.: VIBE: Video Inference for Human Body Pose and Shape Estimation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 5253–5263 (2020) 9, 12
17. Kolotouros, N., Pavlakos, G., Black, M., Daniilidis, K.: Learning to Reconstruct 3D Human Pose and Shape via Model-Fitting in the loop. In: Proc. of IEEE International Conference on Computer Vision. pp. 2252–2261 (2019) 12
18. Liang, D.: DistPred: A Distribution-Free Probabilistic Inference Method for Regression and Forecasting. In: Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 753–764 (2025) 5, 8
19. Liu, Y., An, P., Huang, X., Yang, C.: A lightweight method for 3D human pose and shape regression. In: Fourth International Conference on Sensors and Information Technology. vol. 13107, p. 131073C. SPIE (2024) 12
20. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M., Lee, J., Chang, W.T., Hua, W., Georg, M., Grundmann, M.: MediaPipe: A Framework for Perceiving and Processing Reality. In: CVPR Workshop on Computer Vision for Augmented and Virtual Reality (2019) 1
21. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering Accurate 3D Human Pose in the Wild Using IMUs and a Moving Camera. In: Proc. of European Conference on Computer Vision. pp. 614–631 (2018) 9, 12
22. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A Simple Yet Effective Baseline for 3D Human Pose Estimation. In: Proc. of IEEE International Conference on Computer Vision. pp. 2659–2668 (2017) 3
23. Mehraban, S., Adeli, V., Taati, B.: MotionAGFormer: Enhancing 3D Human Pose Estimation with a Transformer-GCNFormer Network. In: Proc. of IEEE Winter Conference on Applications of Computer Vision. pp. 6905–6915 (2024) 1, 9, 12
24. Moon, G., Chang, J.Y., Lee, K.M.: Camera Distance-Aware Top-Down Approach for 3D Multi-Person Pose Estimation From a Single RGB Image. In: Proc. of IEEE International Conference on Computer Vision. pp. 10132–10141 (2019) 6, 11
25. Nguyen, H.S., Kim, M., Im, C., Han, S., Han, J.: ConvNeXtPose: a fast accurate method for 3d human pose estimation and its AR fitness application in mobile devices. IEEE Access **11**, 117393–117402 (2023) 1, 4, 5, 9, 10
26. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7745–7754 (2019) 3
27. Peng, J., Zhou, Y., Mok, P.: KTPFormer: Kinematics and Trajectory Prior Knowledge-Enhanced Transformer for 3D Human Pose Estimation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 1123–1132 (2024) 1, 3, 8
28. Qiu, Z., Qiansheng, Y., Jian, W., Dongmei, F.: IVT: An End-to-End Instance-guided Video Transformer for 3D Pose Estimation. In: Proc. of ACM International Conference Multimedia. pp. 6174–6182 (2022) 3
29. Shazeer, N.: Fast Transformer Decoding: One Write-Head is All You Need. arXiv:1911.02150 (2019) 13
30. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral Human Pose Regression. In: Proc. of European Conference on Computer Vision. pp. 536–553 (2018) 3, 4, 11
31. Tang, Z., Qiu, Z., Hao, Y., Hong, R., Yao, T.: 3D Human Pose Estimation With Spatio-Temporal Criss-Cross Attention. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 4790–4799 (2023) 1, 3
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proc. of Neural Information Processing Systems. p. 6000–6010 (2017) 1, 7

33. Wang, D., Zhang, S.: Spatial-Aware Regression for Keypoint Localization. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 624–633 (2024) 3, 5
34. Xie, B., Liu, G., Deng, F., Lu, M.: AITEPose: Learning an End-to-End Monocular 3D Human Pose Estimator via Auxiliary-Information-Driven Training Enhancement. IEEE Transactions on Circuits and Systems for Video Technology pp. 9775–9788 (2025) 12
35. Yu, B.X., Zhang, Z., Liu, Y., Zhong, S.H., Liu, Y., Chen, C.W.: GLA-GCN: Global-local Adaptive Graph Convolutional Network for 3D Human Pose Estimation from Monocular Video. In: Proc. of IEEE International Conference on Computer Vision. pp. 8818–8829 (2023) 1
36. Zhang, J., Tu, Z., Yang, J., Chen, Y., Yuan, J.: MixSTE: Seq2seq Mixed Spatio-Temporal Encoder for 3D Human Pose Estimation in Video. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 13232–13242 (2022) 1, 3, 7, 12
37. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic Graph Convolutional Networks for 3D Human Pose Regression. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 3420–3430 (2019) 3
38. Zhao, Q., Zheng, C., Liu, M., Wang, P., Chen, C.: PoseFormerV2: Exploring Frequency Domain for Efficient and Robust 3D Human Pose Estimation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 8877–8886 (2023) 1, 3, 9
39. Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., Ding, Z.: 3D Human Pose Estimation with Spatial and Temporal Transformers. In: Proc. of IEEE International Conference on Computer Vision. pp. 11656–11665 (2021) 1, 3
40. Zhou, Y., Cheng, Z.Q., Li, C., Fang, Y., Geng, Y., Xie, X., Keuper, M.: Hypergraph Transformer for Skeleton-based Action Recognition. arXiv:2211.09590 (2022) 7
41. Zhu, W., Ma, X., Liu, Z., Liu, L., Wu, W., Wang, Y.: MotionBERT: A Unified Perspective on Learning Human Motion Representations. In: Proc. of IEEE International Conference on Computer Vision. pp. 15039–15053 (2023) 1, 3, 12