Fusionformer: Exploiting the joint motion synergy with fusion network based on transformer for 3D human pose estimation

Xinwei Yu
Xidian University, No.266, Xinglong section, Xiliang Road, Xi’an, Shaanxi province, 710126, China
yxw13137069201@163.com

Abstract. This paper proposes a 2D-3D supervised Fusionformer method for current 3D human pose estimation. It introduces self-trajectory module and cross-trajectory module to capture the motion differences and synergy of different joints. In addition, the created Global Local Fusion Block (GLF) combines global spatio-temporal pose features and local joint trajectory features in parallel. Furthermore, to eliminate the impact of poor 2D poses on 3D projection, a pose refinement network is introduced to balance the consistency of the 3D projection. Finally, the proposed method is evaluated on two benchmark datasets: Human3.6M and MPI-INF-3DHP. Compared to Poseformer and MGCN baseline methods, the results show an improvement of 3.0% MPJPE and 2.0% MPJPE on the Human3.6M dataset. By fully exploiting the characteristics of local joint synergy and adaptively fusing them with global pose features, our method demonstrates superior performance in 3D human pose estimation.

Keywords: Fusionformer, 3D human pose estimation, self-trajectory module, cross-trajectory module, local joint synergy, GLF network.

1. Introduction
3D Human pose estimation has played an increasingly important role in the field of computer vision during recent years, such as virtual reality, motion capture, and action recognition. 3D human pose estimation can be defined as predicting the joint positions of the human body from pictures or videos. According to previous work, there are two principal solutions to this task: (1) single-stage: end-to-end model (2) two-stage: 2D lifting to 3D model. Since the first scheme directly regresses to the 3D coordinates, the source of its error cannot be identified, so the related work [1, 2] did not produce the desired effect. 1 Our work follows [3-6] to train a high precision 2D lifting to 3D model, which detects 2D keypoints firstly and then lift them to 3D. Although many excellent work [6-8] has made huge progress on this task, there were still some problems. At first, the projection of 2D key points into 3D poses is a multi-solution problem, so there is a certain depth ambiguity. In addition, some joints are occluded when performing some complex movements.

In order to address the aforementioned issues, numerous methods not only capture the constraint relationship between joints in the spatial domain but also integrate multiple frames of information in the time domain to achieve more robust and stable representations. [9] utilized a convolutional neural network (CNN) to effectively process information in a temporal sequence. However, a drawback of this approach is the limited receptive field. [7] et al. employed a Transformer Encoder-based network [10]
to model the relationship between human pose sequences in time and space simultaneously [3, 11].

A recent study [6] examined the temporal expression of motion in each joint (see Figure 2 (a)). It separated all the joints and models the relationship of each joint in the time domain through a parallel network. This approach effectively simulated the inter-frame correspondence of each joint, enhancing the temporal and spatial expression of local body posture. We could not help but question whether joints are independent or completely unrelated to each other in terms of time.

However, through meticulous examination of the frames selected from Figure 1, which capture the walking action at time instances $t_0$ and $t_1$, we can discern the prominent movement trajectories of the upper body $P_{b\text{up}}^{t_0\rightarrow t_1}$ and the lower body $P_{b\text{down}}^{t_0\rightarrow t_1}$. In Figure 1, it is evident that the wrist joint $p_j^1$ and the elbow joint $p_j^2$ of the upper body exhibit similar directional movements, resulting in closely aligned trajectories $p_j^1_{t_0\rightarrow t_1}$ and $p_j^2_{t_0\rightarrow t_1}$. The same observation holds true for the two joints $p_j^3$ and $p_j^4$ selected from the lower body. By observing the body’s movement trajectory and the joint’s movement trajectory, we found that in movements where there is strong dependence between adjacent joints, the coordinated movement of neighboring joints produces local movement trajectories of body parts. In conclusion, adjacent joints exhibit a certain level of correlation instead of complete isolation in the time domain. Each joint is not only spatially related to each other but also has a certain degree of temporal correlation (as shown in Figure 2 (b)).

Based on the aforementioned conclusion, the temporal correlation of joints is incorporated into the process of modeling local joint features. Initially, we follow the approach of extracting a single joint based on [6] and model the trajectory of each individual joint in parallel processing. However, it brought a large number of network parameters. A simplified approach is developed to address this challenge by aligning all joints to a unified dimension and feeding them into a sub-network. This methodology effectively eliminates redundancy in the network modules, as depicted in the accompanying Figure 4. By adopting this simple approach, we can explore the temporal self-correlation of each joint in a manner similar to how [7] explored spatial dependencies among joints, making it both straightforward and consistent.

Figure 1. The two provided images depict the human body’s motion postures at time $t_0$ and $t_1$ respectively. The red arrow $P_b$ represents the overall direction of body movement, with $P_{b\text{up}}$ and $P_{b\text{down}}$ denoting the upper and lower body parts respectively. The blue arrow $p_{j_i}$ signifies the motion direction of the $i$-th joint.
After the modeling of individual joint self-correlation, the degree of inter-correlation among different joint trajectories is further explored. Initially, all joint trajectories were fed into a self-attention network to uncover their correlations. However, this approach did not capture the cross-trajectory correlations of joints across multiple frames. Instead, it provides feature correlations obtained after compressing all trajectories. We hypothesize that the dynamic changes in joint coupling may be caused by variations in human movements and the distances between different joints, as observed in their respective temporal trajectories. Inspired by the works of many excellent mask models [12], we attempted to mask specific joints in a single joint trajectory while exploring the temporal inter-correlation of different joint trajectories. By masking joints with lower inter-correlation, we aimed to enhance the network’s attention to joints that exhibit strong coordination. Our proposed method effectively addresses the issue of insufficient temporal extraction of local joint features in human posture. Furthermore, it enhances the model’s capability to handle missing or obscured joint data. This results in an overall improvement of the network’s robustness and generalization capabilities.

In conclusion, we propose a novel network architecture called Fusionformer for 3D human pose estimation. Fusionformer consists of two branches, namely: (i) Global Information Interaction Module (GIM): exploiting global features in space and time, (ii) Local Information Interaction Module (LIM): locally capturing differences of a single joint and the synergy of different joints in the temporal domain. In GIM, the temporal-spatial communication encoder from [7] is adopted to capture both the spatial features of 3D human poses and the temporal features between multiple frames. In LIM, we introduce two novel encoders: the joint self-trajectory encoder (STE) and the joint cross-trajectory encoder (CTE). STE effectively models the temporal self-correlation features of joints in the same position across dimensions. In contrast, CTE masks out irrelevant joints within the trajectory and subsequently extracts inter-joint correlation features across trajectories for joints at different positions. Then, the distinct features generated by the shallow network are fed into the Global Local Fusion Block (GLF) to yield the integrated global-local fusion features. Our method achieves excellent results on low input video frames through parallel- fusion. This approach successfully tackles the problem of limited temporal exploration of local joint features, while also enhancing the model’s capacity to manage absent or obscured joint data, thereby bolstering the network’s resilience and generalization potential. This research advancement holds considerable importance for the domain of computer vision, particularly in the context of human pose estimation.

Our contributions are four-fold as follows:

- Proposed method Fusionformer incorporates the modeling of joint correlations, effectively capturing the individual characteristics of each joint and exploring the mutual relationships among different joints in time domain.
Furthermore, the synergy of joint movements is modeled by leveraging their associated characteristics. By determining multiple local synergy directions among various joints, the accuracy and stability of human posture estimation are enhanced.

This paper creates Global Local Fusion Block (GLF) for the adaptive integration of global pose and local joint features.

Fusionformer has achieved the most advanced performance and excellent generalization capabilities on the two challenging data sets of 3D HPE.

2. Related work

3D HPE can be performed from a single view or multiple views. The method [13, 14] based on multi-view estimation has high accuracy by employing the uniqueness of transformation from camera coordinates to world coordinates. However, it is complex to obtain multi-view data in the actual situations, so our proposed method has better robustness and generalization ability by training a single-view model. Early 3D human pose estimation method [15] relied on manual modeling of bones and extracting prior motion features. With the improvement of computing power, data-driven deep learning methods are emerging. These methods can be categorized into end-to-end methods and 2D lifting to 3D methods. The end-to-end approach directly generates 3D joint sequences from input images or videos. Although such an approach omits the intermediate representation of 2D poses, it increases the sources of error and computational consumption. Some end-to-end methods [1, 2] suffer from it. Benefiting from efficient 2D pose detectors OpenPose [16], CPN [17], AlphaPose [18], and HR-Net [19], many recent methods have also adopted 2D lifting to 3D methods, while their performances are far superior to end-to-end methods. [20] stacked linear layers to regress the 3D pose, utilizing only a single frame to predict the 3D coordinates. However, consecutive multiple frames can provide sufficient temporal information to improve the stability and accuracy of the predicted pose. In response, [21] proposes to use recurrent networks to mine the temporal relationships between multiple frames of input through long- and short-term memory units. Some work [22, 23], using the spatial constraints of human posture, constructs the combined losses to improve the performance of the network. [9] employs causal convolution and free convolution to predict intermediate frames by capturing the relationship between multi-frame inputs. Due to the ability of GCN to efficiently mine irregular connections between nodes, [24] models human posture joints in the form of graph structures, thus overcoming the limitations of CNN and RNN in this regard. Fusionformer also uses the 2D pose detector to obtain 2D keypoints as input, and reconstructs more accurate 3D human poses through the main network.

Recently, the emergence of BERT [25] marks the successful application of Transformer in Natural Language Processing (NLP). The self-attention mechanism [26] is the core structure of Transformer, capable of handling contextual relations between long-range dependencies in parallel. [7] et al. applied a single Transformer network for the first time to explore the temporal and spatial relationships of human pose sequences by constructing two sub-encoders. [27] takes into account the redundancy between input multi-frame sequences, and improves the efficiency of the model by aggregating the timing information of multiple frames into the target frame. [28] utilizes VIT encoder network [10] of different depths to obtain three hypothetical poses with diverse semantic information, and then captures accurate pose representations through self-hypothesis learning and cross-hypothesis learning. However, the above method only captures the global features of human body posture along the time chain. Unlike [7, 27, 28], we use the self-attention mechanism to pay more attention to the correlation between local joints across consecutive frames, and use a simple method to reconstruct the input gesture sequence.

The local context-aware methods [6, 29-32] are mainly designed to extract local features of joints in space-time, due to the motion discrepancy of partial joints and root joints (representing global motion trajectories). [29-31] group joints according to certain characteristics (degrees of freedom and hierarchical relationships) in order to obtain the commonness among the joints in the group. However, grouping according to a fixed frame-work limits the randomness of dependent joint distribution under different movements. [32] proposed to combine Convolutional Neural Network (CNN) and Transformer, and use multiple small-scale convolution kernels to capture any relevant joints. [6] mines the trajectory
of a single joint in time by separating the joints. However, separating joints may ignore the temporal relationship of different joints, which limits the model’s mining of local features. In our work, we further model different joints in the time domain. Through the self-attention mechanism, the similarity of joint motion states between different groups is calculated, and the cooperative representation of local adjacent joints is obtained.

In recent years, feature fusion techniques have been widely used in image segmentation and image classification tasks to extract more comprehensive and precise information. For example, DANet [33] utilizes two parallel spatial attention and channel attention mechanisms to enhance the perception of various positions and semantic information in feature maps, thereby further improving the performance of semantic segmentation. It adopts a simple element-wise addition method to fuse different features. Furthermore, DOLG [34] extracts global and local image features through global and local branches, and then fuses them using a weighted feature fusion approach. This method has been widely used in 3D human pose estimation tasks because global pose features and local joint features have different semantic characteristics. Therefore, this paper also adopts a similar approach by using a dual-branch network to separately handle the overall pose and local joints separately. It also designs a reasonable Global-Local Fusion (GLF) block to adaptively fuse features with different semantics.

3. Method
In this section, the fusion architecture of elevating 2D lifting to 3D is explored. The figure 3 describes the basic structure of the proposed method. The high-precision 2D extractor is utilized to extract the 2D pose sequence from images or videos, followed by a deep neural network that predicts the corresponding 3D pose sequence. Inspired by recent methods such as [7] and [6], we have simultaneously introduced self-trajectory encoders and cross-trajectory encoders based on the existing temporal encoder and spatial encoder. This allows us to explore the differences in individual joint movements and the coordination among different joint movements. Our architecture is mainly divided into two modules: Global interaction module and Local interaction module. By shallow fusion of temporal-spatial global pose features and local joint features, accurate 3D human poses can be efficiently obtained. We assume that the 2D pose sequence obtained from the 2D extractor is represented as \( X = \{ x_1, x_2, x_3, \ldots, x_T \} \), where \( T \) represents the number of input video frames, \( x_t \) represents the 2D human pose at each time step.

3.1. Synergy Geometry Model
In this section, synergy geometry model is presented to accurately capture the synergy representation of joint trajectories in 3D human body pose. Based on the observation in Figure 1, it can be inferred that the movement trajectory \( p_{b_{up}}^{t_0 \rightarrow t_1} \) of the upper body region is potentially formed by the collaborative motion of certain joints around the left elbow \( p_{j_1} \) and left wrist \( p_{j_2} \).

\[
p_{b_{up}}^{t_0 \rightarrow t_1} = \alpha \cdot p_{j_1}^{t_0 \rightarrow t_1} + \beta \cdot p_{j_2}^{t_0 \rightarrow t_1} + \cdots,
\]

(1)

\( \alpha \) and \( \beta \) represent the projection factors of \( p_{j_1} \) and \( p_{j_2} \) respectively.

For the local collaborative direction of each part \( \omega \), the above formulas can be summarized as follows:

\[
p_{b_{\omega}}^{t_0 \rightarrow t_1} = \sum_{i=1}^{j} \sigma_{l,\omega} \cdot p_{j_i}^{t_0 \rightarrow t_1},
\]

(2)

Formula 2 describes how to calculate the local collaborative direction through a weighted sum, where each part \( \omega \) has different projection factors \( \sigma_{l,\omega} \) to determine its contribution to the final result.

At certain moments in joint motion trajectories, due to occlusion or the particularity of motion, the coordination level among different joints is diminished. Considering these factors, we attempt to
utilize a learnable random occlusion matrix $\Phi$ to selectively occlude joints at specific moments in different trajectories. The definition of the occlusion matrix $\Phi$ is as follows:

$$
\Phi_{|J \times T|} = 
\begin{bmatrix}
\varepsilon_1^{t_0} & \varepsilon_1^{t_0+1} & \ldots & \varepsilon_1^{t_1} \\
\varepsilon_2^{t_0} & \varepsilon_2^{t_0+1} & \ldots & \varepsilon_2^{t_1} \\
\vdots & \vdots & \ddots & \vdots \\
\varepsilon_J^{t_0} & \varepsilon_J^{t_0+1} & \ldots & \varepsilon_J^{t_1}
\end{bmatrix},
$$

(3)

In Formula 3, $J$ represents the number of joints, $T$ represents the number of video frames, and $\varepsilon_i^{t} \in [0, 1]$ represents the occlusion factor, with a dimension of $\mathbb{R}^D$, where $D$ denotes the dimension of joint embedding.

**Figure 3.** The proposed architecture consists of two main modules: a Global Interaction Module (GIM), which includes temporal and spatial encoders for global information communication, and a Local Interaction Module (LIM), which includes Self-Trajectory Encoder (STE) and Cross-Trajectory Encoder (CTE) for local information exchange.

Then, we obtain the joint trajectories by element-wise multiplication of the occlusion matrix $\Phi$ and the reconstructed pose matrix $\Lambda$. This formula can be represented as follows:

$$
\begin{bmatrix}
    p_{j_1}^{t_0 \rightarrow t_1} \\
    p_{j_2}^{t_0 \rightarrow t_1} \\
    \vdots \\
    p_{j_J}^{t_0 \rightarrow t_1}
\end{bmatrix} = \Phi \odot \Lambda
$$

(4)

In Formula 4, the reconstructed pose matrix $\Lambda$ has a dimension of $\mathbb{R}^{J \times T}$. The occluded joint trajectories are then fed into an attention network to compute the inter-correlation between trajectories. Finally, multiple local collaborative directions are determined based on the above Formula 2.

Fusionformer in this paper extensively explores the collaborative characteristics of human body poses, allowing for the consideration of local collaborative correlations between different joint trajectories during depth regression. This approach enhances the robustness and accuracy of 3D human pose estimation.
3.2. Transformer Block

Due to the Transformer’s effectiveness in handling long-range dependencies, we incorporate a Transformer-based self-attention mechanism [26] into our approach. This mechanism enables us to effectively capture the local dependence features among joints in human body poses. Moreover, we have made enhancements to specific modules to better explore the joint-specific local dependency characteristics in human body poses. In our method, the 2D pose sequence $X$ is fed into a stack of consecutive Transformer Blocks after block embedding. It is crucial to highlight that separate embedding operations are required for the GIM and LIM inputs, due to their distinct semantic meanings. The block embedding operation is mainly accomplished by employing a simple linear layer to linearly transform the 2D coordinates. Specific embedding can be described as follows:

$$X_{\phi} = \phi(X)$$

where $\phi(\cdot)$ represents the embedding function Linear, $X_{\phi}$ represents the high-dimensional features obtained after embedding the 2D pose sequence $X$. Furthermore, we combine a learnable positional matrix $E_{pos} \in \mathbb{R}^{N \times D}$, where positional embedding can be represented as:

$$\tilde{X}_{\phi} = X_{\phi} + E_{pos}$$

where $N$ represents the extended dimension, $D$ represents the embedding dimension, and $\tilde{X}_{\phi}$ represents the high-dimensional features of $X_{\phi}$ after positional embedding. In Fusionformer, since different encoders capture temporal, spatial, global, and local dependencies of the sequence through different dimensions, we set different $N$ and $D$ for positional embedding to enhance the network’s constraint on joint positions and precise pose representation.

Now let’s review the basic Vision Transformer [10]: The attention mechanism can be described as obtaining three matrices, $Q$ (query), $K$ (key), and $V$ (value), through linear transformations and position embedding of the same input sequence, followed by dot-product operations. The specific formula is as follows:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QQ^T}{\sqrt{d_k}}\right)V$$

where $Q$, $K$, and $V$ are essentially matrices obtained from the same input sequence through linear transformation and positional embedding. Self-attention is obtained through dot-product operations. Since a single head can only focus on a single aspect of features, we can feed $Q$, $K$, and $V$ into multiple heads ($n$ heads) to obtain richer features. This is known as Multi-Head Attention (MHA), and the specific formula is as follows:

$$\text{MHA} = f\left(\text{Concat}(\text{Head}_1, \text{Head}_2, \ldots, \text{Head}_n)\right),$$

where $\text{Head}_i = \text{Attention}(Q_i, K_i, V_i), i \in [1, \ldots, n]$ (8)

where $f$ represents the linear projection function. Subsequent to the self-attention layer, we incorporate normalization layers and feed-forward networks. After acquiring the embedded features as described in Equations 5 and 6, our approach follows the subsequent steps:

$$\tilde{X}_{\phi}^{l-1} = \text{MHA}\left(\text{LN}\left(\tilde{X}_{\phi}^{l-1}\right)\right) + \tilde{X}_{\phi}^{l-1}$$

$$\tilde{X}_{\phi}^{l-1} = \text{FFN}\left(\text{LN}\left(\tilde{X}_{\phi}^{l-1}\right)\right)$$

where $l = 1,2,\ldots,L$

In Formula 9, LN is the normalization layer, FFN is the feed-forward network, and $L$ represents the number of Transformer blocks. In our method, the spatial encoder and temporal encoder have the same structure and are consistent with the Transformer block. However, the self-trajectory encoder and cross-trajectory encoder have undergone some improvements.
3.3. Global Interaction Module

The Transformer-based Global Interaction Module (GIM) learns global spatio-temporal interaction information between joints. Inspired by [7], the network structure of the GIM is illustrated in Figure 3, which consists of two encoder structures: Spatial Encoder (SE) and Temporal Encoder (TE). The main purpose of the Spatial Encoder is to explore the spatial dependencies among all joints at each time step. To obtain high-dimensional feature vectors \( X_\phi \in \mathbb{R}^{T \times J \times D} \), we apply block embedding to each joint in the pose sequence \( X \in \mathbb{R}^{T \times J \times 2} \). Here, \( D \) represents the dimension of the joint embedding. Detailed implementation is presented in Formula 5 as described above. Next, the spatial positional information is incorporated by embedding the learnable position parameters \( E_{pos,s} \in \mathbb{R}^{N \times J \times D} \) to produce \( \hat{X}_\phi \), where the expanded dimension \( N = T \) and the embedded dimension \( D' = J \times D \). The output obtained from the Spatial Encoder is the interaction feature of different joints, with a dimension of \( \mathbb{R}^{T \times J \times D} \). Similarly, the Temporal Encoder is designed to explore the temporal dependencies among the poses at each time step. We flatten the input dimension from \( \mathbb{R}^{T \times J \times D} \) to \( \mathbb{R}^{T \times (J \times D)} \), and the time position parameters \( E_{pos,t} \in \mathbb{R}^{N \times T \times (J \times D)} \) are embedded into it, where the extended dimension \( N = 1 \) and the embedding dimension \( D' = T \times (J \times D) \). Finally, the output obtained after the temporal encoder is the temporal interaction features of poses at different time steps, which have the same dimension as the input.

The structures of TE and SE are the same, as shown in Formula 9 above. By inputting different pose sequence dimensions, we obtain global attention between joints and poses. To better integrate our global features, we combine spatial and temporal features for output. Specific implementation of GIM is described as follows:

\[
\begin{align*}
X_{\phi}^{se} &= SE(X), \\
X_{\phi}^{se\rightarrow te} &= TE(X_{\phi}^{se}), \\
X_{\phi}^{G} &= Concat(X_{\phi}^{se}, X_{\phi}^{se\rightarrow te})
\end{align*}
\]

where SE (\( \cdot \)) and TE (\( \cdot \)) represent the spatial encoder and temporal encoder, respectively. \( X_{\phi}^{se} \) and \( X_{\phi}^{se\rightarrow te} \) represent the outputs after passing through the spatial encoder and temporal encoder, respectively. \( X_{\phi}^{G} \) represents the global features of joint and pose obtained by combining the spatial and temporal outputs, with a size of \( \mathbb{R}^{T \times J \times D} \).

3.4. Local Interaction Module

![Figure 4. The image above shows the detailed structure of the Local Information Interaction Module (LIM), which consists of two encoders: (a) Self-Trajectory Encoder (STE) and (b) Cross-Trajectory Encoder (CTE). Pose Reshape is step for reconstructing pose trajectories. CTA is Cross-Trajectory Attention.](image-url)
The representation of local features is not required to be constrained by all other joints, indicative of local constraints. In contrast to GIM, fine-grained local joint features are captured through the design of the Local Interaction Module (LIM). As shown in Figure 3 and 4, LIM mainly consists of two parts: Self-Trajectory Encoder (STE) and Cross-Trajectory Encoder (CTE).

3.4.1. Self-Trajectory Encoder (STE). Inspired by the [6, 7] approach, we aim to extract self-trajectory features of a single joint across different time steps using a non-splitting strategy. As shown in the above Figure 4(a), each 2D patch represents a joint, where the column dimension represents the number of joints J, and the row dimension represents the number of video frames T. During the spatial encoding of human poses, all joints within each frame are embedded. Then, attention computations are performed. To embed the same joint at different time steps, we need to transpose the input human pose sequence \( \mathbf{X} \in \mathbb{R}^{T \times J \times 2} \) along the joint count dimension J and the video frame dimension T. By reconstructing the 3D pose, the trajectory of a single joint at different time steps can be obtained, denoted as \( \hat{\mathbf{X}} \in \mathbb{R}^{T \times J \times 2} \). Specific implementation is described as follows:

\[
\hat{\mathbf{X}} = \mathbf{X} \text{transposedim}_t, (\text{dim}_j,)
\]

where \text{transposedim} function represents the dimension permutation method, \text{dim}_t represents the joint dimension T, and \text{dim}_j represents the video frame dimension J.

As shown in Figure 3, after embedding the reconstructed trajectory using Formula 5, we obtain \( \hat{\mathbf{X}}_\phi \in \mathbb{R}^{J \times T \times D} \). Then, we expand the embedded position information 6 of the single joint in the temporal dimension, denoted as \( \mathbf{E}_\text{pos, st} \in \mathbb{R}^{N \times T \times D} \), where the expanded dimension \( N = J \) and the embedded dimension \( D' = T \times D \). The embedded features are then fed into a self-attention network to obtain the attention weights of the single joint at different times, thus exploring the self-trajectory features. The subsequent implementation is the same as in Formula 9. Finally, the output dimension of the STE network is \( \mathbb{R}^{J \times T \times D} \).

3.4.2. Cross-Trajectory Encoder (CTE). STE extracts local features of a single joint at different time steps but overlooks the interactions between different joints. To solve this problem, we construct a Cross-Trajectory Encoder (CTE) to learn the interrelated features among different joints. As shown in Figure 4, we flatten the output dimension of STE from \( \mathbb{R}^{J \times T \times D} \) to \( \mathbb{R}^{J \times (T \times D)} \). Then, we embed the position parameters of different joint trajectories using \( \mathbf{E}_\text{pos, ct} \in \mathbb{R}^{J \times (T \times D)} \), where the extended dimension \( N = 1 \) and the embedding dimension \( D' = J \times (T \times D) \).

If the flattened feature is directly fed into a Transformer Block based on self-attention mechanism, it may overlook the cross-trajectory communication between local joints. Therefore, we need to introduce the Cross-Trajectory Attention (CTA) to capture inter-joint interactions as shown in Figure 4 (b).

CTA masks the query vector matrix (Q) and key vector matrix (K) separately to hide certain joints within the trajectories. To achieve this, we establish two matrices of learnable parameters, \( q\_mask \) and \( k\_mask \), used to mask Q and K, respectively. The masking operation is performed through element-wise multiplication (\( \odot \)). Furthermore, the masked Q and K are used to calculate similarity, which generates attention weights for capturing inter-joint interactions. Finally, these attention weights are multiplied by the value matrix (V) to obtain the joint features. This approach ensures that both collaborative features (interactions) and non-collaborative features between joints are effectively captured.

The specific implementation of CTA can be summarized as follows:

\[
\begin{align*}
Q\_\text{masked} & = Q \odot q\_mask \\
K\_\text{masked} & = K \odot k\_mask \\
\text{Attn} & = \text{softmax} (Q\_\text{masked} \otimes K\_\text{masked}) \\
\text{Output} & = \text{Attn} \otimes V
\end{align*}
\]
where \( Q \), \( K \), and \( V \) are matrices obtained through linear transformations of the same input sequence. \( q \_\text{mask} \) and \( k \_\text{mask} \) represent the learnable parameter matrices corresponding to \( Q \) and \( K \). \( \odot \) denotes element-wise multiplication, and \( \otimes \) represents matrix multiplication.

The overall formula of the Cross-Trajectory Encoder (CTE) can be defined as:

\[
X^m_{\phi} = \text{CTA} \left( \text{LN} \left( X^{m-1}_{\phi} \right) \right) + X^{m-1}_{\phi} \\
X_{\phi}^{m} = \text{FFN} \left( \text{LN} \left( X_{\phi}^{m-1} \right) + X_{\phi}^{m-1} \right) \\
X_{\phi}^{m} = \text{LN} \left( X_{\phi}^{m} \right)
\]

(13)

where \( M \) represents the total number of layers in the Cross-Trajectory Encoder. \( X_{\phi}^{m} \) denotes the output of the \( m \)-th layer of the Cross-Trajectory Encoder, with a dimension of \( \mathbb{R}^{T \times J \times D} \).

Similar to global features, the local features derived from STE and CTE are combined for output representation. The Local Integration Module (LIM) is implemented as follows:

\[
X_{\phi}^{\text{ste}} = \text{STE}(X) \\
X_{\phi}^{\text{ste} \oplus \text{cte}} = \text{TE}(X_{\phi}^{\text{ste}}) \\
X_{\phi}^{L} = \text{Concat}(X_{\phi}^{\text{ste}}, X_{\phi}^{\text{ste} \oplus \text{cte}})
\]

(14)

here \( \text{STE} (\cdot) \) and \( \text{CTE} (\cdot) \) represent model representations of the Self-Trajectory Encoder and Cross-Trajectory Encoder, respectively. \( X_{\phi}^{\text{ste}} \) and \( X_{\phi}^{\text{ste} \oplus \text{cte}} \) denote the outputs after passing through the Self-Trajectory Encoder and Cross-Trajectory Encoder, respectively. \( X_{\phi}^{L} \) represents the combined local features that incorporate both self-trajectory and cross-trajectory information, with a dimension of \( \mathbb{R}^{2T \times J \times D} \).

3.5. Global Local Fusion Block

The 3D pose features with global pose characteristics and local joint characteristics are obtained by mining the 2D human body pose sequence through GIM and LIM. If we directly perform concatenation or addition operations, it cannot effectively correct the global features with local features. As shown in Figure 3, the network structure of GLF is presented clearly. In GLF, the global features \( X_{\phi}^{G} \) and local features \( X_{\phi}^{L} \) are first reshaped into dimensions \( R^{2T \times J \times D} \), and then concatenated along the T dimension to obtain \( X_{\phi}^{\text{concat}} \in \mathbb{R}^{4T \times J \times D} \), as follows:

\[
X_{\phi}^{\text{concat}} = X_{\phi}^{G} \oplus X_{\phi}^{L}
\]

(15)

Where \( X_{\phi}^{\text{concat}} \) represents the result obtained by concatenating the global feature \( X_{\phi}^{G} \) and the local feature \( X_{\phi}^{L} \). The symbol \( \oplus \) denotes the concatenation operation. Ultimately, \( X_{\phi}^{\text{concat}} \) contains the information of both global and local features.

To better fuse the global features and the local features, two-dimensional average pooling is performed along the T dimension on the concatenated features \( X_{\phi}^{\text{concat}} \in \mathbb{R}^{4T \times J \times D} \), which are then flattened to obtain the compressed features \( X_{\phi}^{\text{squeeze}} \in \mathbb{R}^{4T} \), as follows:

\[
X_{\phi}^{\text{squeeze}} = \frac{1}{J \times D} \sum_{j=1}^{J} \sum_{d=1}^{D} x_{j,d}(X_{\phi}^{\text{concat}})
\]

(16)

Where \( X_{\phi}^{\text{concat}} \) represents the feature map obtained after concatenation operation. It contains features from J positions and D channels. \( x_{j,d} \) represents the element value at the \( j \)-th position and \( d \)-th channel in the feature map.

We input the compressed features into a fully connected layer to extract the relationships between channels. Then, we use SoftMax activation function to obtain the fusion coefficients \( \partial^{G} \) and \( \partial^{L} \).
between the global features and the local features, both of which have dimensions $\mathbb{R}^{2T}$. The specific formulas are as follows:

$$\mathbf{Y}^{[1,2]} = \omega \cdot \mathbf{X}^{\text{squeeze}} + \beta$$

$$[\partial^G, \partial^L] = \frac{e^{\gamma_i}}{\sum_{j=1}^{4T} e^{\gamma_j}}$$

(17)

Where $\mathbf{Y}^{[1]}$ represents linearly transforming a high-dimensional input vector $\mathbf{X} \in \mathbb{R}^d$ into a low-dimensional input vector $\mathbf{Y}^{[1]} \in \mathbb{R}^k$, and $\mathbf{Y}^{[2]}$ represents the reverse process of dimensionality increase, where $d < k$.

The fusion coefficients $\partial^G$ and $\partial^L$ are expanded to $\mathbb{R}^{2T \times JD}$ and element-wise multiplication is performed with the global features and local features, resulting in outputs with dimensions $\mathbb{R}^{4T \times JD}$. The specific formulas are as follows:

$$\mathbf{Y} = [\partial^G, \partial^L] \odot [\mathbf{X}_\phi^G, \mathbf{X}_\phi^L]$$

(18)

Where $\odot$ denotes element-wise multiplication.

Finally, a normalization layer and a one-dimensional convolutional layer are applied to reduce the dimensionality of the global features and local features, resulting in more robust three-dimensional human pose features $\mathbb{R}^{T \times JD}$, as follows:

$$\mathbf{Y} = \text{Conv}(\text{LN} (\mathbf{Y}))$$

(19)

The resulting output feature tensor $\mathbf{Y}$ will have reduced dimensionality while preserving the important information from the global and local features.

3.6. Regression Head & Pose Refinement

Finally, a regression head is utilized for dimensionality reduction on the fused features of the output. This includes a normalization layer and a linear layer, resulting in $\mathbf{Y} \in \mathbb{R}^{T \times (JD \times 3)}$. Due to the poor feature extraction capability of the 2D pose extractor for some complex actions, there is also a problem of poor consistency in the 3D projection. To address this issue, we use a Pose Refinement (PR) network, which balances the influence of the 2D pose using two linear layers. Two sequences of 3D poses are inputted into the network, with the first one being the predicted output of the network and the second one representing the connection between the true 2D pose and the depth predicted by the first one. Confidence scores are then obtained for both sets of results. Finally, the weighted average of the results is calculated based on the confidence scores to obtain the final prediction.

3.7. Loss Function

Our method builds a loss on all frames of the output and finally predicts intermediate frames. Our model employs a L2 loss to minimize the error between the predicted value and the true label:

$$\mathcal{L}_m = \frac{1}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} \left| \mathbf{y}_{i,j} - \tilde{\mathbf{y}}_{i,j} \right|_2$$

(20)

where $\mathbf{y}_{i,j}$ and $\tilde{\mathbf{y}}_{i,j}$ represent the true label and predicted value of the $j$ joint on the $i$ frame, respectively.

Meanwhile, in order to improve the consistency of 3D projections, we introduce a pose fine-tuning loss $\mathcal{L}_r$ to fine-tune the predicted all frames:

$$\mathcal{L}_r = \frac{1}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} \left| \mathbf{y}_{i,j} - \tilde{\mathbf{y}}_{i,j} \right|_2$$

(21)
where $\tilde{y}_{i,j}$ represents the concatenation of the ground truth 2D labels and the predicted depth for joint $j$ in sample $i$.

In our experiments, the total combined loss can be expressed as:

$$\mathcal{L} = \lambda_m \mathcal{L}_m + \lambda_r \mathcal{L}_r$$

(22)

Where $\lambda_m$ and $\lambda_r$ correspond to the balance weights of $\mathcal{L}_m$ and fine-tune loss $\mathcal{L}_r$, respectively. In order to achieve a balance between different losses, we take inspiration from the coefficient settings used in many fine-tuning methods. After careful experimentation, we have found that setting $\lambda_m = \lambda_r$ leads to the best performance.

4. Experiments

During the experiment, we evaluated the three common 3D HPE datasets, namely Human3.6m, MPI-INF-3DHP. In addition, different compositions of Fusionformer are conducted ablation experiments to verify the performance of each module. Finally, the advantages of our method are intuitively presented through visual test results and joint trajectory attention.

4.1. Implementation Details

In our experiments, we used two 2080Ti NVIDIA GPUs and used the current mainstream pytorch framework. When training the model, we used the Adam optimizer. We set the initial learning rate to 0.001 and multiply it by a jitter factor of 0.95 after every fifth iteration, for a total of 40 iterations. Following the previous method [3, 9, 20, 51], we also adopt a flip strategy for the input to augment the data. We employed refinement modules similar to [3, 24].

4.2. Comparison with State-of-the-Art Methods

4.2.1. Results on Human3.6M. Our proposed Fusionformer is compared with previous mainstream methods on the Human3.6M dataset. Table 1 shows the results of our model with an input of 9 frames. To our surprise, our method outperforms the previous state-of-the-art methods under two metrics, Protocol #1 (49.0mm) and Protocol #2 (38.2mm). Some methods [7, 42] utilize refinement module to further enhance the performance of the model. Compared to them, our method achieved lower MPJPE without utilizing refinement module. Furthermore, through the incorporation of refinement module, we achieved 48.4mm under MPJPE, resulting in a 1mm reduction in error (a relative improvement of 2%) compared to MGCN [24]. Remarkably, our model surpasses the state-of-the-art (SOTA) methods across all action categories, as demonstrated in Figure 5. In particular, our method Fusionformer (†) exhibits a significant improvement of approximately 7% compared to Poseformer [7], such as Eat, Pose, and Walk actions. For actions with strong depth ambiguity, such as Greet, Photo, and Wait, our method also shows a slight improvement compared to MGCN [24]. Figure 5 indicates that our method can accurately predict 3D poses across various complex action categories.

4.2.2. Results on MPI-INF-3DHP. To further examine the generalizability of the model, we also evaluated our method on the MPI-INF-3DHP dataset. In order to test the performance of the model in the low input frame scene, we also use the 9-frame 2D pose sequence as the input of the model. Compared to previous state-of-the-art methods, our method even outperforms some methods in multi-input frame scenarios. Table II demonstrates that our method leads the way on all metrics (PCK, AUC, MPJPE). The improvement in PCK and AUC metrics indicate that our method predicts an increased number of keypoints correctly. Additionally, the MPJPE shows a 14.6mm improvement compared to the second-best method [50]. Table 2 reflects the adaptability of our model to indoor and outdoor and other scenarios.
Figure 5. Comparison of our proposed method with SOTA methods [7,24] on the test set of the Human3.6M dataset, and evaluated the percentage improvement in human pose estimation outcomes for various actions, utilizing Poseformer [7] as a baseline. Using detected 2D poses from Cascaded Pyramid Network (CPN) [52]. † indicates the use of refinement module.

Table 1. Quantitative comparisons of Mean Per Joint Position Error (MPJPE) in millimeter between the estimated pose and the ground-truth on Human3.6M under Protocol #1 and Protocol #2. The inputs are the CPN detection 2D pose. (†) adopts the same refinement module as [3, 24]. Red: best; Blue: second best.

| Protocol | Dir | Disc. | Eat | Greet | Phone | Photo | Pose | Pur. | Sit | SitD | Smoke | Wait | WalkD | Walk | WalkT | Avg |
|----------|-----|-------|-----|-------|-------|-------|------|------|-----|------|--------|------|-------|------|-------|-----|
| Tekin et al [35] (ICCV'2017) | 54.2 | 61.4 | 60.2 | 61.2 | 79.4 | 78.3 | 80.1 | 70.1 | 107.3 | 69.3 | 70.3 | 74.3 | 51.8 | 63.2 | 69.7 |
| Martinez et al [36] (ICCV'2017) | 51.8 | 56.2 | 58.1 | 59.0 | 69.5 | 78.4 | 55.2 | 58.1 | 74.0 | 94.6 | 62.3 | 59.1 | 65.1 | 49.5 | 52.4 | 62.9 |
| Fang et al [37] (AAAI'2018) | 50.1 | 54.3 | 57.0 | 57.1 | 66.6 | 73.3 | 53.4 | 55.7 | 72.8 | 88.6 | 60.3 | 57.7 | 62.7 | 47.5 | 50.6 | 60.4 |
| Hossain et al [38] (ECCV'2018) | 48.4 | 50.7 | 57.2 | 55.2 | 63.1 | 72.6 | 53.0 | 51.7 | 66.1 | 80.9 | 59.0 | 57.3 | 62.4 | 46.6 | 49.6 | 58.3 |
| Pavlakos et al [39] (CVPR'2018) | 48.5 | 54.4 | 54.4 | 52.0 | 59.4 | 65.3 | 49.9 | 52.9 | 65.8 | 71.1 | 56.6 | 52.9 | 60.9 | 44.7 | 47.8 | 56.2 |
| Lee et al [40] (ECCV'2018) | 40.2 | 49.2 | 47.8 | 52.6 | 50.1 | 75.0 | 50.2 | 43.0 | 55.8 | 73.9 | 54.1 | 55.6 | 58.2 | 43.3 | 43.3 | 52.8 |
| Pavllo et al [49] (ICCV'2019) | 47.1 | 50.6 | 49.0 | 51.8 | 53.6 | 61.4 | 49.4 | 47.4 | 59.3 | 67.4 | 52.4 | 49.5 | 55.3 | 39.5 | 42.7 | 51.8 |
| Xu et al. [41] (CVPR'2021) | 45.2 | 49.9 | 47.5 | 50.9 | 54.9 | 66.1 | 48.5 | 46.3 | 59.7 | 71.5 | 51.4 | 48.6 | 53.9 | 39.9 | 44.1 | 51.9 |
| Cai et al. [13]† (ICCV'2019) | 46.5 | 48.8 | 47.6 | 50.9 | 52.9 | 61.3 | 48.3 | 45.8 | 59.2 | 64.4 | 51.2 | 48.4 | 53.5 | 39.2 | 41.2 | 50.6 |
| Zeng et al. [7] (ICCV'2021) | 44.5 | 48.2 | 47.1 | 47.8 | 51.2 | 56.8 | 50.1 | 45.6 | 59.9 | 66.4 | 52.1 | 45.3 | 54.2 | 39.1 | 40.3 | 49.9 |
| Zou et al. [42]† (ICCV'2021) | 45.4 | 49.2 | 45.7 | 49.4 | 50.4 | 58.2 | 47.9 | 46.0 | 57.5 | 63.0 | 49.7 | 46.6 | 52.2 | 38.9 | 40.8 | 49.4 |
| Zhang et al. [43]† (CVPR'2023) | 44.8 | 49.1 | 45.6 | 49.3 | 51.0 | 57.5 | 46.7 | 45.3 | 55.8 | 62.5 | 51.0 | 46.7 | 52.8 | 38.1 | 40.6 | 49.1 |

Fusionformer(ours) | 45.4 | 48.5 | 45.4 | 48.2 | 50.6 | 57.0 | 47.6 | 44.9 | 59.3 | 66.3 | 50.7 | 46.8 | 50.6 | 37.2 | 38.6 | 49.0 |

Fusionformer(ours)† | 45.1 | 48.9 | 43.8 | 48.1 | 50.1 | 56.7 | 46.5 | 44.7 | 57.5 | 64.6 | 49.9 | 46.1 | 50.4 | 36.6 | 38.5 | 48.4 |
different joints. However, the temporal relationships between joints but also enhances the synergistic movement coordination among achieving 49.7mm and 49.4mm. It is evident that our designed LIM not only effectively captures the dependencies of human pose separately by employing a double branch may lead to the loss of global or local information. Therefore, we constrain the global and local information. Therefore, we constrain the global and local information. Therefore, we constrain the global and local information. Therefore, we constrain the global and local information. Therefore, we constrain the global and local information. Therefore, we constrain the global and local information. Therefore, we constrain the global and local information.

In this section, we assess the effectiveness of each component in our model by analyzing their impact and performance. To conduct this evaluation, we utilize the Human3.6M dataset and employ the CPN [49] detector to obtain 2D keypoints for accurate evaluation.

| Protocol #2 | Dir. | Desc. | Eat | Greet | Phone | Photo | Pose | Sit | SitD | Smoke | Wait | WalkD | Walk | WalkT | Avg. |
|-------------|------|-------|-----|-------|-------|-------|------|-----|------|-------|------|-------|------|-------|------|
| Martinez et al. [36] (ICCV'2017) | 39.5 | 43.2 | 46.4 | 47.0 | 51.0 | 56.0 | 41.4 | 40.6 | 56.5 | 69.4 | 49.2 | 45.0 | 49.5 | 38.0 | 43.1 | 47.7 |
| Sun et al. [43] (ICCV'2017) | 41.1 | 44.3 | 45.0 | 45.4 | 51.5 | 53.0 | 43.2 | 41.3 | 59.3 | 73.3 | 51.0 | 44.0 | 48.0 | 38.3 | 44.8 | 48.3 |
| Fang et al. [44] (AAAI'2018) | 38.2 | 41.7 | 43.7 | 44.9 | 48.5 | 55.3 | 40.2 | 38.2 | 54.5 | 64.4 | 47.2 | 44.3 | 47.3 | 36.7 | 41.7 | 45.7 |
| Hossain et al. [38] (ECCV'2018) | 35.7 | 39.3 | 44.6 | 43.0 | 47.2 | 54.0 | 38.3 | 37.5 | 51.6 | 61.3 | 46.5 | 41.4 | 47.3 | 34.2 | 39.4 | 44.1 |
| Lee et al. [49] (ECCV'2018) | 34.9 | 35.2 | 43.2 | 42.6 | 46.2 | 55.0 | 37.6 | 38.8 | 50.9 | 67.3 | 48.9 | 35.2 | 50.7 | 31.0 | 34.6 | 43.4 |
| Cai et al. [3] (ECCV'2019) | 36.8 | 38.7 | 38.2 | 41.7 | 40.7 | 46.8 | 37.9 | 35.6 | 47.6 | 51.7 | 41.3 | 36.8 | 42.7 | 31.0 | 34.7 | 40.2 |
| Pavllo et al. [9] (CVPR'2019) | 36.0 | 38.7 | 38.0 | 41.7 | 40.1 | 45.9 | 37.1 | 35.4 | 46.8 | 53.4 | 41.4 | 36.9 | 43.1 | 30.3 | 34.8 | 40.0 |
| Zeng et al. [45] (ECCV'2020) | 35.8 | 39.2 | 36.6 | 36.9 | 39.8 | 45.1 | 38.4 | 36.9 | 47.7 | 54.4 | 38.6 | 36.3 | 39.4 | 30.3 | 35.4 | 39.4 |
| Zou et al. [24] (ICCV'2021) | 35.7 | 38.6 | 36.3 | 40.5 | 39.2 | 44.5 | 37.5 | 35.4 | 46.4 | 51.2 | 40.5 | 35.6 | 41.7 | 30.7 | 33.9 | 39.1 |
| Cai et al. [49] (ICASSP'2023) | 35.1 | 38.6 | 36.6 | 39.4 | 39.8 | 43.8 | 36.7 | 34.7 | 45.3 | 52.8 | 39.9 | 35.4 | 41.8 | 31.2 | 33.5 | 39.0 |

| Fusionformer(ours) | 34.9 | 37.5 | 35.8 | 38.2 | 39.3 | 43.0 | 36.0 | 34.0 | 47.5 | 52.4 | 40.1 | 34.9 | 39.3 | 28.5 | 31.6 | 38.2 |
| Fusionformer(ours)† | 34.7 | 37.8 | 35.5 | 38.1 | 38.6 | 42.3 | 35.4 | 34.4 | 46.2 | 51.1 | 40.1 | 35.1 | 39.3 | 28.5 | 32.7 | 37.9 |

Table 2. Quantitative comparison with the state-of-the-art methods on MPI-INF-3DHP. The best result is marked in bold.

| Method | PCK↑ | AUC↑ | MPJPE↓ |
|--------|------|------|--------|
| Mehta et al. [47] (3DV'2017) | 75.7 | 39.3 | 117.6 |
| Pavllo et al. [9] (CVPR’2019) | 86.0 | 1.9 | 84.0 |
| Lin et al. [48] (BMVC’2019) | 83.6 | 51.4 | 79.8 |
| wang et al. [49] (ECCV’2020) | 86.9 | 62.1 | 68.1 |
| zheng et al. [7] (ICCV’2021) | 88.6 | 56.4 | 77.1 |
| zhang et al. [6] (CVPR’2022) | 94.4 | 66.5 | 54.9 |
| Hu et al. [50] (MM’ 2021) | 97.9 | 69.5 | 42.5 |
| Fusionformer(ours) | 98.0 | 70.1 | 27.9 |

4.3. Ablation Study
In this section, we assess the effectiveness of each component in our model by analyzing their impact and performance. To conduct this evaluation, we utilize the Human3.6M dataset and employ the CPN [52] detector to obtain 2D keypoints for accurate evaluation.

4.3.1. Effect of Model Components. In Table 3, we analyze the impact of each module on network performance through experiments. In the single-branch design, we achieved results of 49.9mm and 50.2mm when evaluating GIM and LIM independently, respectively. However, the constraint of a single branch may lead to the loss of global or local information. Therefore, we constrain the global and local dependencies of human pose separately by employing a double-branch architecture. With the addition of STE and CTE to the GIM, we observed respective improvements in the experimental results, achieving 49.7mm and 49.4mm. It is evident that our designed LIM not only effectively captures the temporal relationships between joints but also enhances the synergistic movement coordination among different joints. However, the presence of different semantic pose features in the two branches makes...
the integration challenging, resulting in an MPJPE of 49.4mm. To address this issue, we introduced the GLF network to fuse global and local features, reducing the discrepancy between each branch. This resulted in an improved MPJPE of 49.0mm. After incorporating the refinement module, the MPJPE was ultimately improved to 48.4mm. The above experiments demonstrate that by introducing a dual-branch network that captures global and local features separately, we have achieved remarkable results. Although this approach has led to a certain increase in parameter count and FLOPs, the performance improvement is substantial.

Table 3. Ablation study on different components of our Fusionformer. GIM donates Global Interaction Module, LIM donates Local Interaction Module. The four different encoders (SE, TE, STE, CTE) are shown in the table below. GLF is Global Local Fusion Block, and PR represents the Pose Refinement Module.

| GIM | LIM | GLF | PR | Params (M) | FLOPs (M) | MPJPE (mm) |
|-----|-----|-----|-----|------------|----------|-----------|
| SE  | TE  | STE | CTE |            |          |           |
| Single-branch | ✔ | ✔ | ✔ | 9.58M | 150 | 49.9 |
| | | | | 9.63M | 153 | 50.2 |
| Double-branch | ✔ | ✔ | ✔ | 10.14M | 178 | 49.7 |
| | ✔ | ✔ | ✔ | 10.87M | 185 | 49.4 |
| | ✔ | ✔ | ✔ | 11.21M | 210 | 49.0 |
| | ✔ | ✔ | ✔ | 11.30M | 212 | 48.4 |

4.3.2. Impact of Parameters in STE and CTE. Table 4 shows how different parameter settings of STE and CTE affect the performance and computational complexity of the model. Based on the outstanding performance of Poseformer, we initially set the embedding dimension \( D = 32 \), the number of layers in the spatial encoder \( L_{se} = 4 \), and the number of layers in the temporal encoder \( L_{te} = 4 \), to achieve optimal performance for GIM. The results indicate that we obtained poor performance when setting lower numbers of layers for STE and CTE. However, by continuously stacking more layers, we consistently gained improvements in performance. After setting \( L_{ste} = 3 \) and \( L_{cte} = 3 \) separately, we observed that no further improvement was achieved, with the value remaining at 48.4mm. Continuing with the increase in embedding dimension \( D \), we observed a continuous decline in experimental results, indicating that the model’s performance had already reached saturation. Due to the shallow network architecture and low input video frames, it can be observed from the table that our model has a relatively low parameter count.

Table 4. Ablation study on different parameters of STE and CTE. Here, \( L_{ste} \) and \( L_{cte} \) indicate the number of STE and CTE layers, respectively. \( D \) represents the embedding dimension.

| \( L_{ste} \) | \( L_{cte} \) | \( D \) | Params (M) | MPJPE (mm) |
|-------------|-------------|-----|----------|-----------|
| 2           | 2           | 32  | 171      | 49.5      |
| 2           | 3           | 32  | 189      | 49.1      |
| 3           | 2           | 32  | 193      | 48.7      |
| 3           | 3           | 32  | 212      | 48.4      |
| 4           | 4           | 32  | 256      | 48.5      |
| 5           | 5           | 32  | 317      | 49.0      |
| 3           | 3           | 48  | 651      | 48.7      |
| 3           | 3           | 64  | 1063     | 48.9      |
Figure 6. Visualization of self-attentions among different joint trajectories. The x-axis and y-axis correspond to the queries and the predicted outputs, respectively. Each row shows the attention weight \( w_{i,j} \) of the j-th query for the i-th output.

4.4. Qualitative Results
From Figure 6, in order to show the synergy between joints, we further visualize the trajectory attention of different joints. We selected a continuous sequence of 9 frames for attention visualization in a specific action. From the trajectory attention maps, we can observe the temporal dependencies between different joints. In Figure 6, it is specifically demonstrated that adjacent or distant joints exhibit high attention weights over time, indicating the synergistic movement coordination among these joints. Furthermore, Figure 7 presents a visual comparison of our method with Poseformer under the scenario of low input frames. We selected three different actions captured from multiple viewpoints for this analysis. Finally, we conducted tests on outdoor actions in various scenarios, and achieved accurate reconstruction of 3D human poses. The visual results mentioned above demonstrate that our model, by capturing the joint-level local coordination features of human poses, can accurately project the 3D pose of the human body in complex actions.

Figure 7. Qualitative comparison among the proposed method (Fusionformer), PoseFormer on Human3.6M dataset.

5. Conclusion
In this paper, we propose Fusionformer, a Transformer-based fusion network. Similar to the two-stream network, we introduce two branches GIM and LIM. The GIM module handles the global spatio-temporal
correlation of human poses. LIM not only models the temporal trend of a single joint, but also deeply mines the temporal synergy of different joints. Moreover, we create GLF to fuse the global and local features. The GLF module learns the fusion coefficient of global features and local features to balance the fused generation of 3D poses. Finally, we also introduce a fine-tuning network to eliminate the influence of poor 2D pose on 3D projection consistency to further improve the stability of the model. The results of two benchmark datasets (Human3.6M, MPI-INF-3DHP) demonstrate the excellent performance of our model on the task of 3D human pose estimation. However, we also acknowledge that our current method still has some limitations. Firstly, dealing with high input frames can lead to a large number of parameters, resulting in increased computational burden. Therefore, future work can explore model compression and acceleration techniques to improve efficiency and adapt to larger-scale datasets. Secondly, in addition to pose estimation, future research can focus on action prediction and generation to achieve higher-level analysis and synthesis of human behavior. Furthermore, exploring directions such as cross-domain and cross-modal pose estimation, as well as incorporating prior knowledge and contextual information, are also worth investigating. In conclusion, while our method has achieved significant advancements in 3D human pose estimation tasks, there are still several aspects that require further improvement and exploration. By optimizing the model, expanding the application domains, and incorporating advanced techniques, we can propel the development of 3D human pose estimation, enabling more accurate, robust, and efficient pose estimation methods to be applied in practical scenarios.

References
[1] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[2] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In European Conference on Computer Vision, pages 752–768. Springer, 2020.
[3] 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, pages 2272–2281, 2019.
[4] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5064–5073, 2020.
[5] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In Proceedings of the IEEE international conference on computer vision, pages 2640–2649, 2017.
[6] 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, pages 13232–13242, 2022.
[7] 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, pages 11656–11665, 2021.
[8] 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, pages 13147–13156, 2022.
[9] 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, pages 7753–7762, 2019.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv, abs/2010.11929, 2020.

[11] Wenbo Hu, Changgong Zhang, Fangneng Zhan, Lei Zhang, and Tien-Tsin Wong. Conditional directed graph convolution for 3d human pose estimation. Proceedings of the 29th ACM International Conference on Multimedia, 2021.

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 16000–16009, 2022.

[13] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 4341–4350, 2019.

[14] Size Wu, Sheng Jin, Wentao Liu, Lei Bai, Chen Qian, Dong Liu, and Wanli Ouyang. Graph-based 3d multi-person pose estimation using multi-view images. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 11128–11137, 2021.

[15] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T Tan. Occlusion-aware networks for 3d human pose estimation in video. In Proceedings of the IEEE/CVF international conference on computer vision, pages 723–732, 2019.

[16] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1302–1310, 2017.

[17] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[18] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[19] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[20] 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, 2021.

[21] Xiaoxiao Du, Ram Vasudevan, and Matthew Johnson-Roberson. Bio-Istm: A biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction. IEEE Robotics and Automation Letters, 4(2):1501–1508, 2019.

[22] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. 2017 IEEE International Conference on Computer Vision (ICCV), pages 2621–2630, 2017.

[23] 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 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 2272–2281, 2019.

[24] 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, pages 11477–11487, 2021.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In North American Chapter of the Association for Computational Linguistics, 2019.
[26] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Neural Information Processing Systems, 2017.

[27] W. Li, H. Liu, R. Ding, M. Liu, P. Wang, and W. Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. 2021.

[28] Wenhai Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13137–13146, 2022.

[29] Jue Wang, Shaoli Huang, Xinchao Wang, and Dacheng Tao. Not all parts are created equal: 3d pose estimation by modeling bi-directional dependencies of body parts. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.

[30] Xiangtao Zheng, Xiumei Chen, and Xiaoliang Lu. A joint relationship aware neural network for single-image 3d human pose estimation. IEEE Transactions on Image Processing, 29:4747–4758, 2020.

[31] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Snnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. ArXiv, abs/2007.09389, 2020.

[32] M. Hassanin, A. Khamiss, M. Bennamoun, F. Boussaid, and I. Radwan. Crossformer: Cross spatio-temporal transformer for 3d human pose estimation. 2022.

[33] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3141–3149, 2019.

[34] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 11752–11761, 2021.

[35] Bugra Tekin, Pablo M rquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, pages 3941–3950, 2017.

[36] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[37] Hao-Shu Fang, Yuanlu Xu, Wenguang 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, 2018.

[38] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 68–84, 2018.

[39] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7307–7316, 2018.

[40] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In Proceedings of the European conference on computer vision (ECCV), pages 119–135, 2018.

[41] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3d human pose estimation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 16105–16114, 2021.

[42] Shengping Zhang, Chenyang Wang, Liqiang Nie, Hongxun Yao, Qingming Huang, and Qi Tian. Learning enriched hop-aware correlation for robust 3d human pose estimation. International Journal of Computer Vision, 131(6):1566–1583, 2023.

[43] Shuang Liang, Xiao Sun, and Yichen Wei. Compositional human pose regression. Computer Vision and Image Understanding, 176:1–8, 2018.
[44] Haoshu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In AAAI, 2018.

[45] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, pages 507–523. Springer, 2020.

[46] Jialun Cai, Hong Liu, Runwei Ding, Wenhao Li, Jianbing Wu, and Miaoju Ban. Htnet: Human topology aware network for 3d human pose estimation. arXiv preprint arXiv:2302.09790, 2023.

[47] Dushyant Mehta, Helge Rhodin, Dan Casas, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation using transfer learning and improved CNN supervision. CoRR, abs/1611.09813, 2016.

[48] Jiahao Lin and Gim Hee Lee. Trajectory space factorization for deep video-based 3d human pose estimation. ArXiv, abs/1908.08289, 2019.

[49] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. CoRR, abs/2004.13985, 2020.

[50] W. Hu, C. Zhang, F. Zhan, L. Zhang, and T. T. Wong. Conditional directed graph convolution for 3d human pose estimation. 2021.

[51] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In Proceedings of the European conference on computer vision (ECCV), pages 529–545, 2018.

[52] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7103–7112, 2018.