UNSPAT: Uncertainty-Guided SpatioTemporal Transformer for 3D Human Pose and Shape Estimation on Videos

Minsoo Lee*, Hyunmin Lee*, Bumsoo Kim, Seunghwan Kim
LG AI Research
{minsoo.lee, hyunmin, bumsoo.kim, sh.kim}@lgresearch.ai

Abstract

We propose an efficient framework for 3D human pose and shape estimation from a video, named Uncertainty-Guided SpatioTemporal Transformer (UNSPAT). Unlike previous video-based methods that consider temporal relationships with global average pooled features, our approach incorporates both spatial and temporal dimensions without compromising spatial information. We address the excessive complexity of spatiotemporal attention through two modules: Spatial Alignment Module (SAM) and Space2Batch. The modules align input features and compute temporal attention at every spatial position in a batch-wise manner. Furthermore, our uncertainty-guided attention re-weighting module improves performance by diminishing the impact of artifacts. We demonstrate the effectiveness of the UNSPAT on widely used benchmark datasets and achieve state-of-the-art performance. Our method is robust to challenging scenes, such as occlusion, and cluttered backgrounds, showing its potential for real-world applications.

1. Introduction

3D human pose and shape estimation is a task to reconstruct a human mesh from an input image or video using parametric models [2, 27, 32]. The geometric and motion information from humans provide huge potential across a wide range of applications, such as computer graphics, motion analysis, healthcare, AR/VR. Recent advances in deep learning have made significant progress in single-frame estimation [16, 21, 22, 24, 25, 31, 33, 35, 41] by predicting the skinned multi-person linear model (SMPL) [27] parameters in a data-driven way, simplifying the complicated reconstruction of meshes with a single linear function.

However, frame-based estimations are limited in that they are prone to suffer from challenging factors such as motion blur and occlusion which are present in the scene, resulting in temporally unstable prediction. Thus, an important step towards a robust human pose and shape estimator is to consider the spatiotemporal relationships in an input sequence, enabling the model to cope with these challenges.

To this end, video-based approaches have been previously proposed [7, 20, 28, 39, 40]. Current video-based studies in literature exploit a rather straightforward extension to the temporal axis, where they first extract static features from each frame individually, and then aggregate static features to compose temporal-aware features. The aggregated feature from a specific window of frames is used to predict the SMPL parameter for a given timestamp. Although this straightforward extension has been shown to improve temporal errors (*e.g.* acceleration error), it results in a significant increase in reconstruction errors (*e.g.* MPJPE, and PA-MPJPE). This is mainly because previous works spatially pooled the individual frame into a single feature vector to model temporal relationships with a reasonable level of complexity.

In this paper, the main challenge boils down to a simple question: "Is it possible to build a 3D human pose and shape estimation framework that takes into account both the spatial and temporal dimensions?". At a glance, a spatiotemporal-aware framework may seem simple, especially since recent studies have demonstrated the impressive performance of transformer self-attention [10, 11, 14, 34, 37] in modeling relationships between every input token across multiple dimensions [3, 5, 13, 26, 43]. However, as the complexity of transformer attention grows quadratically with respect to the input size, extending the input to both the spatial and temporal axis without pooling not only causes a huge burden in complexity but also results in slow convergence and degraded performance owing to the excessive amount of input tokens [6, 9, 12, 44]. Recent video-based methods [39, 40] using transformer architecture were unable to address this issue and therefore limited to modeling the temporal consistency between input frames without considering spatial information.

To address this issue, we propose a simple yet efficient framework UNcertainty-guided SPAtioTemporal Transformer (UNSPAT) that efficiently incorporates spatial information in the temporal axis. For computational efficiency,

*Equal contribution.

we propose two modules, named *spatial alignment module (SAM)* and *Space2Batch*. SAM spatially aligns adjacent features to a current timestamp feature with a predicted affine transformation matrix. In Space2Batch, input features are spatially aligned by SAM, thus enabling a decomposition of spatial relationships with temporal relationships. By treating the spatial axis in a batch-wise manner, the transformer attention can be computed on the temporal axis within the same spatial position. This approach significantly reduces the complexity of full spatiotemporal attention while still computing temporal attention at every spatial position.

For the accuracy gain, we propose an *uncertainty-guided attention re-weighting* method and utilize the inverse kinematics [24] procedure by retaining spatial information. Our *uncertainty-based attention re-weighting* method learns to discriminate spatiotemporal positions that contain artifacts and diminishes the impact of such positions on the final attention weight. The predicted uncertainty map helps to filter out misleading spatial information from adjacent frames. Powered by the uncertainty-guided attention re-weighting, UNSPAT effectively handles challenging scenes, showing its potential for real-world applications.

With its efficiently designed framework, UNSPAT successfully encodes spatiotemporal information without a high complexity. Furthermore, explicitly modeling uncertainty enhances the robustness against misleading information. We evaluate the proposed method on a widely used benchmark dataset and show that the UNSPAT outperforms previous state-of-the-art methods.

The contributions of this paper are:

- We propose a transformer-based framework UNSPAT for 3D human pose and shape estimation in a video that incorporates spatial information in the temporal axis.
- We propose two modules that effectively relieve the complexity of calculating spatiotemporal attention, namely the *spatial alignment module* and *Space2Batch*.
- We propose *uncertainty-guided attention re-weighting* method that enhances the robustness of UNSPAT for challenging scenes by selectively filtering out misleading information.
- Our UNSPAT achieves state-of-the-art performance in both mesh reconstruction and temporal accuracy on various benchmarks.

2. Related Work

Image-based 3D human pose and shape estimation. Pioneer studies on estimating 3D human pose and shape from a monocular image primarily predicted the parameters of the 3D human body model [2, 27, 32]. In particular, SMPL [27] is the most widely used parametric model as it is a statistical model that encodes human subjects with pose and shape parameters. With the advances in deep learning, an increasing number of studies began to employ a deep network to directly regress the pose and shape parameters from an RGB image. However, while 2D key points can be easily annotated in a variety of scenes with different people and backgrounds, accurately annotating the 3D ground truth from in-the-wild scenes is challenging. Therefore, several methods leverage additional cues to estimate accurate SMPL parameters. Some of them leverage 2D information that includes 2D joint heatmap and silhouettes [33], 2D body/part segmentation [31,41], 2D keypoint re-projection loss [16,36]. On the contrary, HybrIK [24] combines the advantages of a 3D skeleton and a parametric model. Specifically, the approach involves transforming 3D joints into relative rotations of the skeletons, thus obtaining a more accurate and realistic 3D skeleton derived from the reconstructed 3D mesh. This helps to close the loop between the 3D skeleton and the parametric body model.

Despite the promising achievements in image-based 3D human pose estimation methods, there still exist several problems. As the networks estimate 3D pose from an image, they are vulnerable to occlusion, which often results in temporally jittering outputs.

Video-based 3D human pose and shape estimation. Compared to image-based methods, video-based methods encounter more challenges, such as processing temporal information and finding correspondence between spatial and temporal information. HMMR [17] proposed a 1D convolution temporal encoder that learns to capture 3D human dynamics by estimating SMPL parameters of past and future frames. VIBE [20] used a bi-directional gated recurrent unit (GRU) to encode static features from the input frames into a temporal feature. They also proposed an adversarial learning framework to produce feasible poses from a motion generator. MEVA [28] decomposed a human motion into a coarse motion estimated by motion compression autoencoder and a residual motion learned through motion refinement. TCMR [7] used GRU to leverage temporal information from the past and future frames, reducing the strong dependency on the current static feature. The methods incorporate temporal information from adjacent frames and successfully produce temporally plausible outputs.

Recently, transformer architecture is adopted for video-based methods to aggregate sequence features for its powerful performance [39, 40]. However, because the complexity of transformer attention increases quadratically with respect to its input dimension, they spatially global average pool their spatiotemporal features when computing temporal self-attention. MAED [39] completely decomposes the temporal and spatial axes and computes self-attention for each axis, while MPS-Net [40] uses the global average pooled feature as input to the transformer. However, none of these

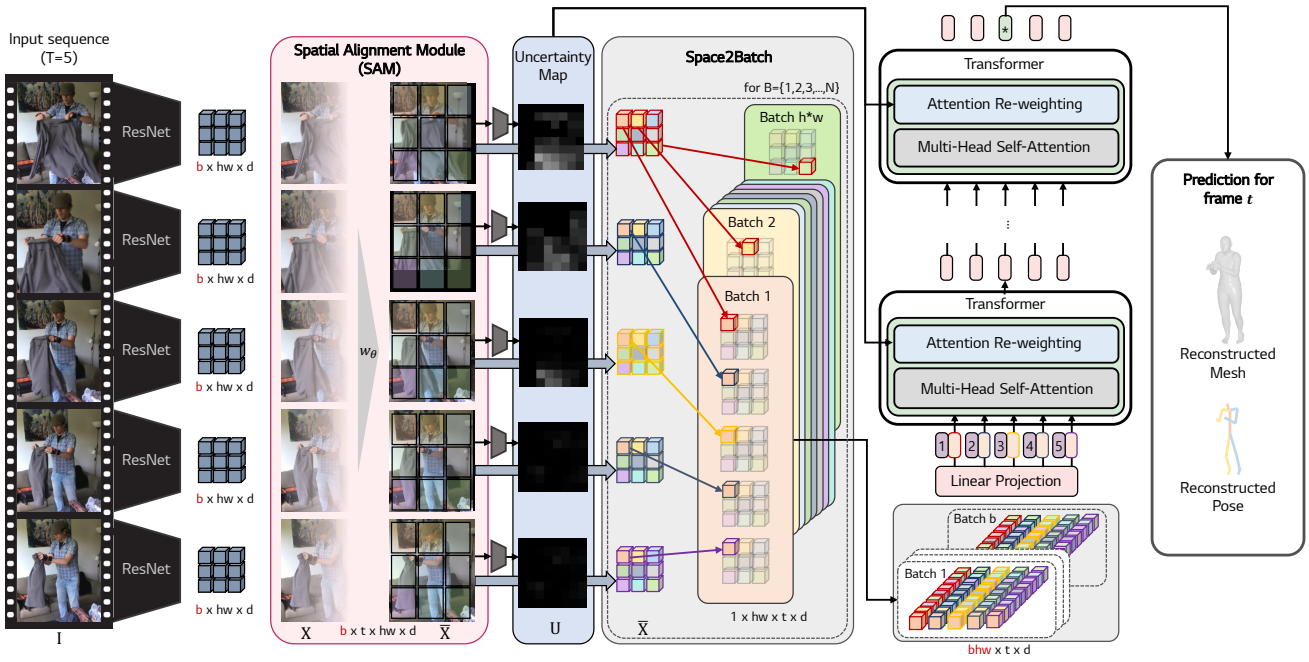

Figure 1. Overview of UNSPAT. First, we extract a spatial feature $\mathbf{x}_t$ from the input $\mathbf{I}_t$. Then, our Spatial Alignment Module (SAM) aligns the adjacent features to the current feature by applying affine transformations. Next, the aligned features $\bar{\mathbf{x}}_t$ are used as input to the estimator, which predicts the uncertainty map $\mathbf{u}_t$. To prepare $\bar{\mathbf{x}}_t$ for the transformer, it is reshaped from (b, t, hw, d) to (bhw, t, d) using Space2Batch. Within the transformer, our uncertainty-guided re-weighting adjusts the attention weights by utilizing the predicted uncertainty map $\mathbf{u}_t$. Finally, the 3D keypoints and 3D human mesh are predicted by several layers and inverse kinematics.

methods fully consider spatiotemporal self-attention, while it is the straightforward extension that could potentially improve performance. In this paper, to overcome this limitation, we propose well-designed transformer architecture that can compute spatiotemporal self-attention without a significant increase in complexity.

3. Proposed Method

Given an input image sequence $\{\mathbf{I}_t\}_{t=1}^{T}$, our task is to estimate 3D human pose and shape for the current frame, *i.e*. middle frame, by leveraging information from adjacent frames. To represent the human pose and shape, we employ the parametric human body model, SMPL [27]. Specifically, a human body is represented by its shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ and pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{24\times3}$, where the pose parameter signifies the axis-angle rotation of each joint and the shape parameter represents the linear coefficient for the principal component of the parametric human shape. By estimating a set of $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, a 3D human mesh can be recovered by the predefined SMPL vertex regressor.

3.1. SpatioTemporal Transformer

Spatial feature extraction and alignment We first extract the bounding box of a person in an image $\mathbf{I}_t$, and then extract a feature $\mathbf{x}_t$ upon the bounding box. In contrast to the previous video-based methods [7, 20, 28, 40] that used average pooled vectors as features, we sustain spatial information in the feature $\mathbf{x}_t \in \mathbb{R}^{h\times w\times d}$ to obtain accurate mesh. The temporal features are used to exploit useful information from adjacent frames and strengthen temporal consistency. Previous studies relied on a temporal encoder (*e.g*., GRU, transformer) to aggregate static feature vectors into temporal features. In contrast, because our features include a spatial dimension, a more elaborate procedure is needed. Specifically, bounding boxes of human regions estimated by the tracker are not well-aligned. Therefore, directly aggregating features among the temporal domains would lead to unwanted results, especially when the camera or the person in the scene moves fast. To solve this problem, we propose a *spatial alignment module (SAM)* that learns affine transformation parameters to transform each of the adjacent features to the current feature.

SAM takes a pair of a current feature and an adjacent feature as input and predicts the affine transformation matrix. We denote the module as g_θ and the affine transformation parameters as Θ, $g_\theta(\mathbf{x}_t, \mathbf{x}_{t+\delta}) \rightarrow \Theta \in \mathbb{R}^3$. The elements of Θ correspond to scale and translation along the x and y axis, respectively. Specifically, g_θ first computes the visual similarity of features by dot product operation. Then, two fully connected layers are applied to obtain the transformation parameters. Note that we only predict scale and translation parameters because rotation and shear rarely happen in pre-

dicted bounding boxes in a real-world scenario.

$$(s, t_x, t_y) = g_\theta(\mathbf{x}_t, \mathbf{x}_{t+\delta}) \quad A = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix}. \quad (1)$$

Given affine transformation matrix A, an adjacent feature $\mathbf{x}_{t+\delta}$ can be warped to align with current feature. The warping operation W can be expressed as $W(\mathbf{x}_{t+\delta}, A) \rightarrow \bar{\mathbf{x}}_{t+\delta}$.

Space2Batch. Let the sequence of features that are spatially aligned by *SAM* be denoted as $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_t\}_{t=1}^T$, where $\bar{\mathbf{x}}_t \in \mathbb{R}^{b\times w\times h\times d}$. Given query, key, and value as $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{b\times w\times h\times d}$, we encode $\bar{\mathbf{X}}$ with N layers of transformer encoders to model the spatiotemporal relationship between frames within a certain temporal window. Because the complexity of the transformer attention grows quadratically with respect to its input dimension (*i.e.*, softmax$(\frac{\mathbf{QK}^T}{\sqrt{d}})V$), computing the attention for $\bar{\mathbf{X}}$ results in a complexity of $O(dw^2h^2T^2)$. To deal with the excessive complexity caused by calculating attention for each and every point in the spatiotemporal axis, we propose Space2Batch. As the spatial positions for the features in a temporal window have been aligned by SAM, we can decompose the temporal correlations from spatial positions by treating the spatial dimensions ($h\times w$) as a batch, thus calculating the attention *only* between identical spatial positions. This results in a significantly reduced complexity from $O(dw^2h^2T^2)$ to $O(dwhT^2)$ while achieving better performance than that when considering the full spatiotemporal attention (see Experiment for details).

Uncertainty-guided attention. When considering spatiotemporal correlations with transformers, it is crucial to prevent the propagation of errors in specific frames to the overall sequence prediction. This is especially the case for videos, where challenging factors (*e.g.*, occlusion, dilution) in certain frames lead to the erroneous output. However, previous studies have naively aggregated the temporal relationships with transformers and thus suffer from propagated errors. To this end, we propose a novel uncertainty-guided attention re-weighting featuring an uncertainty estimator.

The key to our attention re-weighting is the prediction of the spatiotemporal positions where the model is highly vulnerable to *uncertain* information. During training, we intentionally create *synthetic* artifacts by replacing random spatiotemporal positions with those of other videos in the batch. Then, we train a small network to discriminate the artifacts that usually occur in challenging regions. During test time, the *uncertainty map* predicted by the estimator is used to re-weight the attention for the spatiotemporal positions where artifacts might have occurred.

Our spatiotemporal transformer calculates the spatial attention weight $\mathbf{a}_{i,j} \in \mathbb{R}^{h\times w}$ for $(i, j) \in \{0, ..., T\}$, where i and j denote query and key time sequence, respectively. Then, we penalize $\mathbf{a}_{i,j}$ with the predicted uncertainty map $\mathbf{u}_j$ for the key time sequence j. Given $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ and the predicted uncertainty maps $\mathbf{U} = \{\mathbf{u}_t\}_{t=1}^T$, the re-weighted transformer attention is written as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\big(\frac{\mathbf{QK}^T}{\sqrt{d}} - \gamma\mathbf{U}\big)\mathbf{V}, \quad (2)$$

where γ is a scale that balances the original attention with the uncertainty prediction. This uncertainty-based re-weighting helps identify the areas that require assistance from adjacent frames when making predictions for the current frame. Moreover, it prevents erroneous information to be propagated to the adjacent frames.

3.2. Pose and Shape Estimation

To obtain a 3D mesh from the encoded spatiotemporal feature, we follow the architecture of Li *et al*. [24] which utilizes a 3D keypoints estimation and inverse kinematics.

3D keypoints estimation. The encoded feature $\mathbf{m}_t$ goes through three deconvolution layers followed by a 1 × 1 convolution layer to obtain a 3D body keypoints heatmap. Then, the soft-argmax operation is utilized to estimate the 3D keypoints $\mathbf{P} = \{\mathbf{p}_k\}_{k=1}^K$.

Pose and shape estimation via inverse kinematics. Similarly to Li *et al*. [24], we utilize inverse kinematics to estimate a relative rotation matrix $\mathbf{R}_k \in \mathbb{SO}(3)$ for each joint k from the estimated 3D keypoints $\mathbf{P} = \{\mathbf{p}_k\}_{k=1}^{24}$. According to twist-and-swing decomposition [4], $\mathbf{R}_k$ can be decomposed as:

$$\mathbf{R}_k = \mathbf{R}_k^{sw}\mathbf{R}_k^{tw}, \quad (3)$$

where $\mathbf{R}_k^{sw}$ and $\mathbf{R}_k^{tw}$ denote swing rotation and twist rotation, respectively. Through the inverse kinematics algorithm, $\mathbf{R}_k^{sw}$ can be determined from $\vec{\mathbf{p}}_k$ and $\vec{\mathbf{t}}_k$, where $\mathbf{t}_k$ denotes the k^{th} joint position of the template skeleton, and $\vec{\mathbf{p}}_k$ and $\vec{\mathbf{t}}_k$ denote the relative location from its parent joint. The swing rotation can be expressed as the rotation axis $\vec{\mathbf{n}}_k$ and the angle α_k. Thus, $\mathbf{R}^{sw}$ can be determined by the Rodrigues formula as:

$$\begin{aligned} \vec{\mathbf{n}}_k &= \frac{\vec{\mathbf{t}}_k \times \vec{\mathbf{p}}_k}{\|\vec{\mathbf{t}}_k \times \vec{\mathbf{p}}_k\|}, \\ \cos\alpha_k = \frac{\vec{\mathbf{t}}_k \cdot \vec{\mathbf{p}}_k}{\|\vec{\mathbf{t}}_k\|\|\vec{\mathbf{p}}_k\|}, &\quad \sin\alpha_k = \frac{\vec{\mathbf{t}}_k \times \vec{\mathbf{p}}_k}{\|\vec{\mathbf{t}}_k\|\|\vec{\mathbf{p}}_k\|}, \end{aligned} \quad (4)$$

$$\mathbf{R}_k^{sw} = I + \sin\alpha_k[\vec{\mathbf{n}}_k]_\times + (1 - \cos\alpha_k)[\vec{\mathbf{n}}_k]_\times^2, \quad (5)$$

where I represents the 3 × 3 identity matrix, and $[\vec{\mathbf{n}}_k]_\times$ denotes the skew-symmetric matrix of $\vec{\mathbf{n}}_k$.

Also, the twist rotation $\mathbf{R}_k^{tw}$ can be expressed with the axis $\vec{\mathbf{t}}_k$ and the twist angle ϕ_k as:

$$\mathbf{R}_k^{\text{tw}} = \mathbf{I} + \frac{\sin\phi_k}{\|\vec{\mathbf{t}}_k\|}[\vec{\mathbf{t}}_k]_\times + \frac{(1 - \cos\phi_k)}{\|\vec{\mathbf{t}}_k\|^2}[\vec{\mathbf{t}}_k]_\times^2, \quad (6)$$

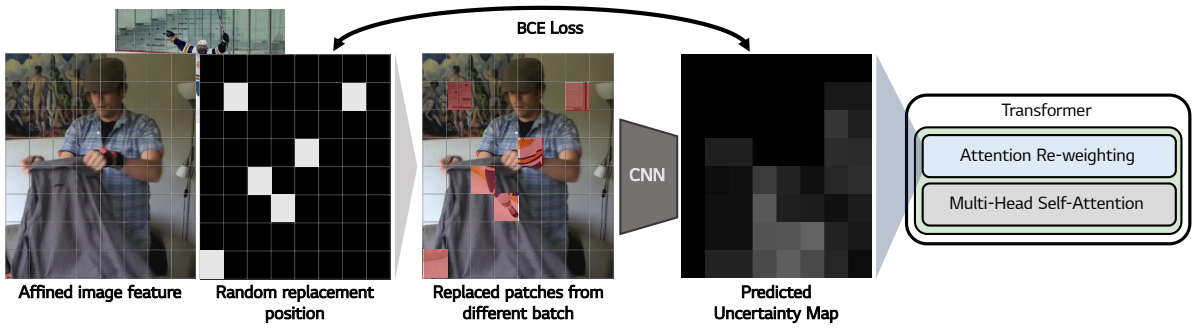

Figure 2. Training scheme of the uncertainty estimator. We intentionally create *synthetic* artifacts by replacing random spatiotemporal positions with other videos in the batch. Then, we train a small network to discriminate the artifacts that usually occur in challenging regions. The uncertainty value of randomly replaced patches is trained to be 1 and otherwise as 0 via BCE loss.

where $[\vec{\mathbf{t}}_k]_\times$ denotes the skew-symmetric matrix of $\vec{\mathbf{t}}_k$. However, unlike $\mathbf{R}_k^{sw}$, the twist rotation $\mathbf{R}_k^{tw}$ cannot be determined since the twist angle ϕ_k cannot be calculated from $\mathbf{p}_k$. Thus, we estimate $\boldsymbol{\phi} = \{\phi_k\}_{k=1}^{24}$ and other parameters such as shape parameters $\boldsymbol{\beta}$, camera parameters $\mathbf{c}$, and joint prediction confidence score $\boldsymbol{\sigma} = \{\sigma_k\}_{k=1}^{24}$. To do this, we apply several linear layers that take encoded feature $\mathbf{m}_t$ and predict the aforementioned parameters after the global average pooling operation. Finally, the relative rotation matrix $\mathbf{R}_t$ can be determined from Eq. 3, Eq. 4, and Eq. 6. Then, we convert it to axis-angle rotation representation, that is the SMPL pose parameters θ.

3.3. Training Loss

Keypoints loss. To stabilize the keypoints estimation training, we incorporate the joint prediction confidence score $\boldsymbol{\sigma}$ into the Laplacian Loss [18] when we supervise the estimated 3D keypoints $\{\mathbf{p}_k\}_{k=1}^K$ as:

$$\mathcal{L}_{\text{Lap.}}(\mathbf{P}, \hat{\mathbf{P}}, \boldsymbol{\sigma}) = -\frac{1}{K}\sum_{k=1}^{K} \ln \frac{1}{\sqrt{2}\sigma_k} \exp -\frac{\sqrt{2}(\hat{\mathbf{p}}_k - \mathbf{p}_k)}{\sigma_k}, \tag{7}$$

where $\{\hat{\mathbf{p}}_k\}_{k=1}^K$ is the ground-truth 3D keypoints. To additionally utilize ground-truth 2D keypoints $\hat{\mathbf{P}}^{2d} = \{\hat{\mathbf{p}}_k^{2d}\}_{k=1}^K$, we project $\mathbf{p}_k$ to image coordinate space with the estimated camera parameter $\mathbf{c}$ as $\mathbf{p}_k^{2d} = \Pi(\mathbf{p}_k, \mathbf{c})$, where Π denotes the semi-perspective projection operation. Thus, our keypoints loss is formulated as:

$$\mathcal{L}_{\text{pose}} = \mathcal{L}_{\text{Lap.}}(\mathbf{P}, \hat{\mathbf{P}}, \boldsymbol{\sigma}) + \mathcal{L}_{\text{Lap.}}(\Pi(\mathbf{P}, \mathbf{c}), \hat{\mathbf{P}}^{2d}, \boldsymbol{\sigma}). \tag{8}$$

SMPL parameter loss. Using the SMPL, we can obtain the skeleton in its rest pose, along with the additive offsets that correspond to the predicted body shape parameters $\boldsymbol{\beta}$. Once we have this rest pose skeleton, we can then calculate the pose parameters $\boldsymbol{\theta}$ using inverse kinematics [24], which allows us to determine the joint angles. We apply the MSE Loss during training to the shape and pose parameters as:

$$\mathcal{L}_{\boldsymbol{beta}} = \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|, \quad \mathcal{L}_{theta} = \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|, \tag{9}$$

where $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\theta}}$ denote the ground-truth SMPL parameters.

Twist angle loss. To avoid discontinuity, we regress a 2-dimensional vector (c_{ϕ_k}, s_{ϕ_k}) that corresponds to the cosine and sine value of ϕ_k, rather than directly predicting ϕ_k.

$$\mathcal{L}_{tw} = \frac{1}{K}\sum_{k=1}^{K} \left\| (c_{\phi_k}, s_{\phi_k}) - \left(\cos\hat{\phi}_k, \sin\hat{\phi}_k\right) \right\|_2, \tag{10}$$

where $\hat{\phi}_k$ denotes the ground-truth joint angle of the k^{th} joint.

Finally, our overall training objective is written as:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{poss}}\mathcal{L}_{\text{pose}} + \lambda_{\text{beta}}\mathcal{L}_{\text{beta}} + \lambda_{\text{theta}}\mathcal{L}_{\text{theta}} + \lambda_{\text{tw}}\mathcal{L}_{\text{tw}}, \tag{11}$$

where λ denotes the relative importance between losses. We set λ_{pose}, λ_{beta}, λ_{theta}, and λ_{tw} as 1, 1, 1, and 1, respectively.

3.4. Implementation Details

We set the input sequence length T to 16 and set the input video frame rate to 25-30 frames per second as done in previous studies [7, 20]. Similarly to previous work, we use pre-trained ResNet-34 model [24] to extract features and adopt weights to initialize the regressor. To sustain spatial information in the feature $\mathbf{X}_t \in \mathbb{R}^{h\times w\times d}$ we omit the global pooling layer at the end of the backbone. Here the size of h, w, d are 8, 8 and 512 respectively. Our transformer consists of three layers and each layer has eight multi-heads. Also, it utilizes learnable positional embeddings. We use the Adam optimizer [19] and train for 90 epochs with a mini-batch size of 8. With our efficiently designed transformer pipeline, we use a single V100 GPU for training and evaluation. In line with previous studies [7, 17, 22], we use the ground-truth bounding box to crop a human in the image. Then the cropped images are resized to 256×256. For data augmentation, we follow TCMR [7] to occlude the cropped image with various objects. Details of the model architecture and hyperparameters are described in the supplementary.

Methods	3DPW				MPI-INF-3DHP			Human3.6M		
	PA-MPJPE ↓	MPJPE ↓	MPVPE ↓	Accel ↓	PA-MPJPE ↓	MPJPE ↓	Accel ↓	PA-MPJPE ↓	MPJPE ↓	Accel ↓
VIBE [20]	57.6	91.9	-	25.4	68.9	103.9	27.3	53.3	78.0	27.3
MEVA [28]	54.7	86.9	-	11.6	65.4	96.4	11.1	53.2	76.0	15.3
TCMR [7]	52.7	86.5	102.9	7.1	63.5	97.3	8.5	52.0	73.6	3.9
MPS-Net [40]	52.1	84.3	99.7	7.4	62.8	96.7	9.6	47.4	69.4	3.6
Ours	45.5	75.0	90.2	7.1	60.4	94.4	9.2	41.3	58.3	3.8

Table 1. Comparison with state-of-the-art video-based models on 3DPW, MPI-INF-3DHP and Human3.6M datasets. All methods are trained with 2D and 3D video datasets, including 3DPW. The colored cells indicate the best accuracy(□), and the second-best accuracy(□) on each evaluation.

Methods	3DPW			
	PA-MPJPE ↓	MPJPE ↓	MPVPE ↓	Accel ↓
HMR [16]	76.7	130.0	-	37.4
GraphCMR [23]	70.2	-	-	-
SPIN [22]	59.2	96.9	116.4	29.8
I2L-MeshNet [30]	57.7	93.2	110.1	30.9
Pose2Mesh [8]	58.3	88.9	106.3	22.6
HybrIK [24]	48.8	80.0	94.5	25.1
VIBE [20]	56.5	93.5	113.4	27.1
TCMR [7]	55.8	95.0	111.5	7.0
MAED [39]	50.7	88.8	104.5	18.0
MPS-Net [40]	54.0	91.6	109.6	7.5
Ours	48.2	77.8	93.8	7.2

Table 2. Comparison with state-of-the-art image-based (top rows) and video-based (bottom rows) methods on 3DPW dataset. All methods are trained without 3DPW datasets.

4. Experiments

4.1. Experiments setup

Datasets. Following TCMR [7], we use a mixture of 2D and 3D datasets for training. We use 3D video datasets of Human3.6M [15], MPI-INF-3DHP [29], 3DPW [38]. For 2D datasets we use, Penn Action [42], InstaVariety [17], and PoseTrack [1] datasets. Among these datasets, only 3DPW contains accurate ground-truth SMPL parameters and in-the-wild scenes. For evaluation, we use Human3.6M [15], MPI-INF-3DHP [29], 3DPW [38] datasets. The supplementary material contains more detailed information.

Evaluation metrics. We report widely used evaluation metrics for 3D human pose and shape estimation in video. For mesh reconstruction, we consider the mean per joint position error (MPJPE), Procrustes-aligned MPJPE (PA-MPJPE), and mean per vertex position error (MPVPE) in mm. MPJPE is the mean per joint position error after aligning the root joint. This is calculated based on the Euclidean distance between ground-truth and estimated joint positions. The PA-MPJPE is calculated after rigidly aligning the estimated joints to ground-truth joints. For temporal accuracy, we report acceleration error [17] which computes an average acceleration of each joint in mm/s^2.

Training and evaluation protocols. We compare our UNSPAT with previous state-of-the-art methods following their certain training protocols. In Table 1, we compare video-based methods that utilize image features extracted from a pre-trained network without fine-tuning the feature extractor. All methods are trained with a mixture of 2D and 3D video datasets. In Table 2, we compare UNSPAT with image-based and video-based methods using another training protocol. All methods are trained without the 3DPW dataset, but there are no restrictions on using other datasets. We further compare recent video methods based on transformer architecture like our UNSPAT in Table 3.

4.2. Comparison with state-of-the-art methods

Video-based methods. We compare the reconstruction and temporal performance of our UNSPAT with those of previous video-based 3D human pose and shape estimation methods [7,20,28,40]. Table 1 shows that our method outperforms previous state-of-the-art methods by a large margin on all three datasets. Our approach outperforms MPS-Net [40] despite the fact that both methods use transformer architecture to leverage temporal information. A more detailed comparison with MPS-Net is handled in the following subsection. With the result, we demonstrate the effectiveness of our UNSPAT that aggregates spatiotemporal information and prevents error propagation with the proposed uncertainty-guided transformer. More comparative experiments are included in the supplementary material.

Image-based and video-based methods. We further compare UNSPAT with image-based and video-based methods with the 3DPW test set, which is composed of a challenging in-the-wild scene. As shown in Table 2, our UNSPAT outperforms all image- and video-based methods with respect to the reconstruction error. We noticed that previous video-based methods successfully reduced temporal error; however, despite possessing more information by using sequence features, the reconstruction performance is inferior to the image-based methods. We think neglecting spatial information is the main bottleneck of the accuracy gain. On the other hand, our UNSPAT shows notable performance in terms of reconstruction and temporal accuracy. We again

Methods	FLOPs (G) ↓	# Parameters (M) ↓	3DPW (training w/ 3DPW) PA-MPJPE ↓	MPJPE ↓	MPVPE ↓	Accel ↓	3DPW (training w/o 3DPW) PA-MPJPE ↓	MPJPE ↓	MPVPE ↓	Accel ↓
MAED [39]	136.7G	60.1M	45.7	79.1	92.6	17.6	50.7	88.8	104.5	18.0
MPS-Net [40]	4.5G	39.6M	52.1	84.3	99.7	7.4	54.0	91.6	109.6	7.5
Ours	12.9G	11.8M	45.5	75.0	90.2	7.1	48.1	78.3	94.7	7.4

Table 3. Comparison of transformer-based architectures. We compare FLOPs, the number of network parameters, and the model performance using the 3DPW dataset.

	Module S2B	SAM	Unc.	3DPW PA-MPJPE ↓	MPJPE ↓	MPVPE ↓	ACCEL ↓
(1)	X	X	X	62.5	95.8	114.8	8.5
(2)	O	X	X	50.0	80.4	97.5	7.4
(3)	O	X	O	48.7	80.2	95.5	8.5
(4)	O	O	X	48.7	79.4	95.5	7.3
(5)	O	O	O	48.0	78.2	94.5	7.4

Table 4. Ablation study of proposed methods: Space2Batch (S2B), Spatial Alignment Module (SAM) and Uncertainty-guided re-weighting (Unc.).

demonstrate the effectiveness of our UNSPAT in aggregating spatiotemporal information by presenting a significant improvement in reconstruction accuracy without compromising temporal accuracy.

Comparison with transformer-based methods. In this paragraph, we analyze our model and the recent works that use transformer architecture, such as MAED [39] and MPS-Net [40]. We show how each method handles spatiotemporal features and also evaluate with the benchmark dataset in Table 3. We further show qualitative results in Figure 3

MPS-Net [40] applies global average pooling to the spatiotemporal feature in the spatial axis and then computes self-attention along the temporal axis, resulting in the complexity of $O(dT^2)$. This approach encodes temporal information with low computational complexity, showing low FLOPs and acceleration error. However, it sacrifices spatial information, leading to poor reconstruction performance.

MAED [39] completely decomposes temporal and spatial axes and computes attention weights for each axis. It applies global average pooling on the spatial axis when computing attention for the temporal axis, and computes full attention for the spatial axis. However, this architecture greatly increases the complexity of the model when computing attention for the spatial axis, which is $O(dh^2w^2T)$. In addition, the model gets biased towards spatial information over temporal information. Therefore, MAED shows reasonable reconstruction performance but suffers from a high computational cost and acceleration error.

By applying SAM and Space2Batch, our model computes temporal attention for each spatial position in the complexity of $O(dhwT^2)$. This approach is novel and thus results in high reconstruction performance and low acceleration error with less than 10% of the computational cost of MEAD as shown in Table 3.

(a) Cropped image (b) Uncertainty (ours) (c) Ours (d) MPS-Net (e) MAED

Figure 3. Qualitative results of methods using transformer architecture. (a) Cropped image with the human-centered bounding box. (b) Predicted uncertainty map, higher values indicate higher uncertainty. (c-e) Reconstructed meshes by methods using transformer architecture. Among methods, only ours reconstructed accurate mesh. As shown in (b), uncertainty values at truncated human body parts are high. In other words, our model will aggregate information from adjacent features. With the result, we demonstrate that UNSPAT correctly incorporates spatiotemporal information.

4.3. Ablation study

Here we conduct experiments in Table 4 to show the effectiveness of each of the modules in the UNSPAT, that is Space2Batch, spatial alignment, and uncertainty-guided attention re-weighting.

Effectiveness of Space2Batch. In this ablation, we compare our UNSPAT against variants without Space2Batch, which corresponds to Table 4 (1). The most straightforward implementation of spatiotemporal attention will be the full attention on all the h, w, t axis. However, despite its exorbitant complexity, this variant shows serious degradation in performance compared to (2). We conjecture that this is mainly due to the excessive amount of tokens causing the initial attention value to start at a very small uniform value. As introduced in previous work in literature [6,9,12,44], this leads to slow convergence and lower performance. Thus, our Space2Batch which decomposes the spatial axis with temporal attention by aligning the spatial positions using SAM shows a significant reduction in complexity while achieving a noticeable improvement in performance.

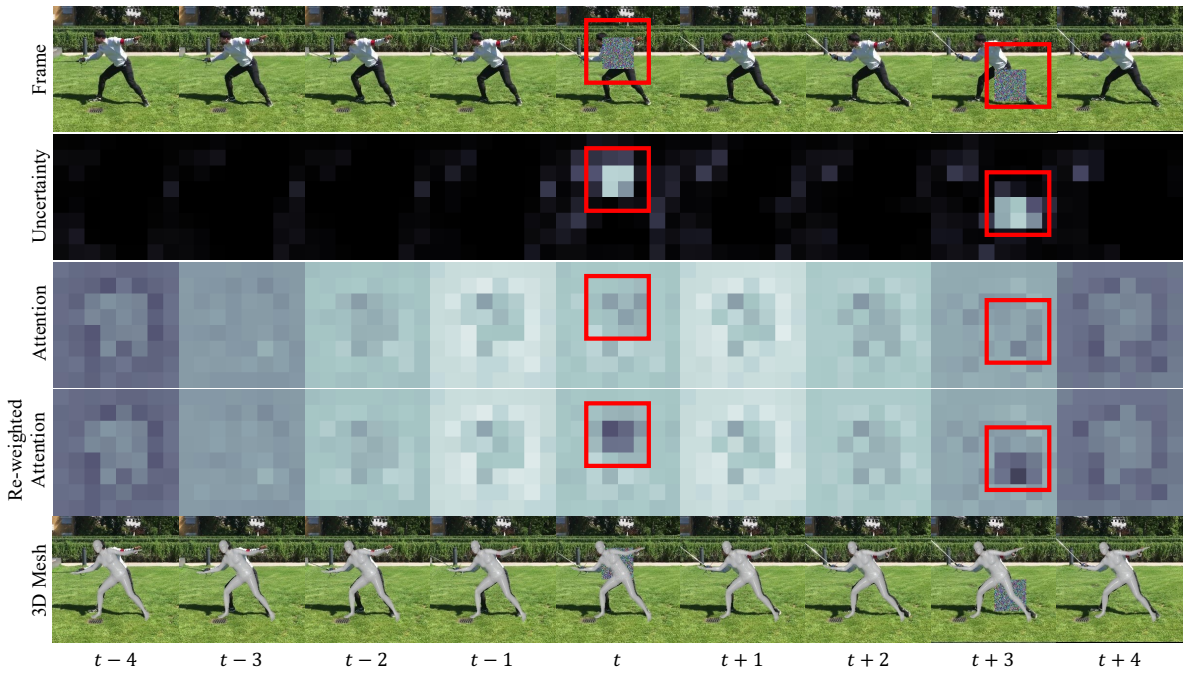

Figure 4. The visualization of uncertainty-guided attention re-weighting. The figure presents how our uncertainty-guided attention re-weighting method works in the presence of occlusion by adding synthetic noise patches to the input frame sequence (the first row). To demonstrate the effectiveness of our method, we show the predicted uncertainty map (the second row), the attention map (the third row), and the re-weighted attention map (the fourth row). In the visualization, brighter colors indicate higher values, whereas darker colors indicate lower values. Lastly, we present the predicted 3D mesh (the fifth row) obtained when each time sequence is the current time sequence.

Effectiveness of spatial alignment module. We compare variants (2), (3), and (5) to analyze the effectiveness of SAM and found that applying SAM improves both the reconstruction performance and acceleration performance in all cases. We speculate that this is because our SAM aligns the features, allowing more accurate information to be retrieved compared to when this alignment is not present in the surrounding frames.

Uncertainty-guided attention re-weighting. We further examine the effects of uncertainty-guided attention re-weighting by comparing variants (2), (4), and (5). While the uncertainty-guided attention re-weighting improves the reconstruction performance in all cases, it decreases the acceleration performance when not used with SAM. This is because our transformer architecture makes a weak assumption about the alignment between frames. Therefore, in our final model, where we employ uncertainty-guided attention re-weighting and SAM simultaneously, we observed an increase in reconstruction performance with only a minimal decrease in acceleration. As a result, our final model achieved the best overall performance.

4.4. Attention Visualization

In Figure 4, we present a visualization of the attention weights and uncertainty map to demonstrate how our uncertainty-guided attention re-weighting method works when a synthetic noise patch is added to the frames. The second row shows the predicted uncertainty map $\mathbf{u}_j$. We denote spatial attention as $\mathbf{a}_{i,j} \in \mathbb{R}^{h \times w}$ for $(i, j) \in \{0, ..., T\}$, where i and j indicate query and key time sequence, respectively. Specifically, the third row displays the attention map $\mathbf{a}_{t,j}$ when the query is the current time sequence t and the key is each other time sequence j.

As shown in the figure, our uncertainty estimator accurately predicts the high uncertainty values for the occluded regions. Furthermore, the predicted uncertainty map $\mathbf{u}_j$ is utilized to re-weight the attention map $\mathbf{a}_{t,j}$ from the third row to the fourth row in Figure 4 using Eq. 2. The last row shows the predicted 3D mesh when each time sequence is the current time sequence. The results demonstrate the robustness of our model to occlusion.

5. Conclusions

We present an efficient framework, the Uncertainty-guided Spatiotemporal transformer (UNSPAT), for 3D human pose and shape estimation from a video. Our approach addresses the limitations of previous video-based methods by incorporating both spatial and temporal information, efficiently aligning input features, and reducing the impact of artifacts through an uncertainty-guided attention re-weighting module. Our experiments demonstrate that UNSPAT achieves state-of-the-art performance on widely used benchmark datasets. We believe that our framework provides a promising approach for real-world applications that require accurate and robust 3D human pose and shape estimation.

Correspond to Bumsoo Kim (bumsoo.kim@lgresearch.ai) or Seung Hwan Kim (sh.kim@lgresearch.ai).

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. PoseTrack: A Benchmark for Human Pose Estimation and Tracking. Proc. of Computer Vision and Pattern Recognition (CVPR), 2018.

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of People. In *Proc. of SIGGRAPH*, 2005.

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[4] Paolo Baerlocher and Ronan Boulic. Parametrization and range of motion of the ball-and-socket joint. In *Deformable Avatars: IFIP TC5/WG5. 10 DEFORM'2000 Workshop November 29–30, 2000 Geneva, Switzerland and AVATARS'2000 Workshop November 30–December 1, 2000 Lausanne, Switzerland*, pages 180–190. Springer, 2001.

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[7] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond Static Features for Temporally Consistent 3D Human Pose and Shape from a Video. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2021.

[8] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, pages 769–787. Springer, 2020.

[9] Xiyang Dai, Yinpeng Chen, Jianwei Yang, Pengchuan Zhang, Lu Yuan, and Lei Zhang. Dynamic detr: End-to-end object detection with dynamic attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2988–2997, 2021.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3621–3630, 2021.

[13] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019.

[14] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. 2014.

[16] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end Recovery of Human Shape and Pose. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3D Human Dynamics from Video. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? volume 30, 2017.

[19] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. Proc. of International Conference on Learning Representations (ICLR), 2014.

[20] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. PARE: Part Attention Regressor for 3D Human Body Estimation. Proc. of Int'l Conf. on Computer Vision (ICCV), 2021.

[22] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. Proc. of Int'l Conf. on Computer Vision (ICCV), 2019.

[23] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 4501–4510, 2019.

[24] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. HybrIK: A Hybrid Analytical-Neural Inverse Kinematics Solution for 3D Human Pose and Shape Estimation. Proc. of Computer Vision and Pattern Recognition (CVPR), 2021.

[25] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022.

[26] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[27] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. In *Proc. of SIGGRAPH ASIA (ACM Trans. on Graph.)*, 2015.

[28] Zhengyi Luo, S Alireza Golestaneh, and Kris M Kitani. 3D Human Motion Estimation via Motion Compression and Refinement. In *Proc. of Asian Conf. on Computer Vision (ACCV)*, 2020.

[29] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. International Conference on 3D Vision (3DV), 2017.

[30] 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 *ECCV*, pages 752–768. Springer, 2020.

[31] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural Body Fitting: Unifying Deep Learning and Model-Based Human Pose and Shape Estimation. In *International Conference on 3D Vision (3DV)*, 2018.

[32] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2019.

[33] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to Estimate 3D Human Pose and Shape from a Single Color Image. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Skeleton-based action recognition via spatial and temporal transformer networks. *Computer Vision and Image Understanding*, 208:103219, 2021.

[35] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Synthetic Training for Accurate 3D Human Pose and Shape Estimation in the Wild. In *Proc. of British Machine Vision Conf. (BMVC)*, 2020.

[36] Hsiao-Yu Fish Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-Supervised Learning of Motion Capture. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. volume 30, 2017.

[38] Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering Accurate 3D Human Pose in the Wild Using IMUs and a Moving Camera. Proc. of European Conf. on Computer Vision (ECCV), 2018.

[39] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with Multi-level Attention for 3D Human Shape and Pose Estimation. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2021.

[40] Wen-Li Wei, Jen-Chun Lin, Tyng-Luh Liu, and Hong-Yuan Mark Liao. Capturing Humans in Motion: Temporal-Attentive 3D Human Pose and Shape Estimation from Monocular Video. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2022.

[41] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, Bill Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly Supervised 3D Human Pose and Shape Reconstruction with Normalizing Flows. In *Proc. of European Conf. on Computer Vision (ECCV)*, 2020.

[42] Weiyu Zhang, Menglong Zhu, and Konstantinos G. Derpanis. From Actemes to Action: A Strongly-Supervised Representation for Detailed Action Understanding. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2013.

[43] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020.

[44] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. 2020.