Uncertainty-Aware Testing-Time Optimization for 3D Human Pose Estimation

Ti Wang, Mengyuan Liu†, Hong Liu, Bin Ren, Yingxuan You, Wenhao Li, Nicu Sebe, Xia Li

***Abstract*—Although data-driven methods have achieved success in 3D human pose estimation, they often suffer from domain gaps and exhibit limited generalization. In contrast, optimization-based methods excel in fine-tuning for specific cases but are generally inferior to data-driven methods in overall performance. We observe that previous optimization-based methods commonly rely on projection constraint, which only ensures alignment in 2D space, potentially leading to the overfitting problem. To address this, we propose an Uncertainty-Aware testing-time Optimization (UAO) framework, which keeps the prior information of pre-trained model and alleviates the overfitting problem using the uncertainty of joints. Specifically, during the training phase, we design an effective 2D-to-3D network for estimating the corresponding 3D pose while quantifying the uncertainty of each 3D joint. For optimization during testing, the proposed optimization framework freezes the pre-trained model and optimizes only a latent state. Projection loss is then employed to ensure the generated poses are well aligned in 2D space for high-quality optimization. Furthermore, we utilize the uncertainty of each joint to determine how much each joint is allowed for optimization. The effectiveness and superiority of the proposed framework are validated through extensive experiments on challenging datasets: Human3.6M, MPI-INF-3DHP, and 3DPW. Notably, our approach outperforms the previous best result by a large margin of 5.5% on Human3.6M.**

***Index Terms*—3D Human Pose Estimation, Testing-time Optimization, Uncertainty Estimation.**

I. Introduction

Monocular 3D human pose estimation aims to estimate 3D body joints from a single image. This task plays an important role in various applications, such as computer animation [1], human-computer interaction [2], and action recognition [3]. However, due to the depth ambiguity, it remains a challenging task that has drawn extensive attention in recent years. With the development of deep learning, data-driven methods [4]–[8] have dominated the field of 3D human pose estimation. During the training process, these methods implicitly acquire knowledge of camera intrinsic parameters and 3D human pose distributions within specific domains. However, these data-driven methods struggle with cross-domain generalization and in-the-wild scenarios [9], [10].

† Corresponding author.

T. Wang, M. Liu, H. Liu, Y. You, and W. Li are with National Key Laboratory of General Artificial Intelligence, Peking University, Shenzhen Graduate School, China. Email: tiwang.cs@gmail.com, youyx@stu.pku.edu.cn, {liumengyuan, hongliu, wenhaoli}@pku.edu.cn. X. Xia is with the Department of Computer Science, ETH Zurich, Zurich 8092, Switzerland. E-mail: ethlixia@gmail.com. B. Ren, is with both the Department of Information, University of Pisa, Italy, and the Department of Information Engineering and Computer Science, University of Trento, Italy. Email: bin.ren@unitn.it. N. Sebe is with the Department of Information Engineering and Computer Science, University of Trento, Italy. E-mail: niculae.sebe@unitn.it.

Historically, traditional solutions usually formulate the 3D human pose estimation task as a pure optimization-based problem [11]–[13]. These optimization-based methods offer two significant benefits. First, they prevent the model from generating unrealistic or out-of-distribution poses, which ultimately boosts the model's robustness. Second, they provide the flexibility of adaptive adjustments tailored for specific samples, thereby enhancing the model's generalization capabilities. Although traditional methods can alleviate domain gaps by estimating 3D poses on a case-by-case basis, their performance is often compromised due to the limitations of manually crafted feature extractors [14]. In contrast, data-driven methods leverage neural networks to learn feature representations in an automated manner, enabling them to effectively capture the intricacy and variability inherent in the data. Recently, some human mesh recovery methods [15]–[19] have also adopted optimization strategies to refine network parameters based on certain constraints. We observe that the previous optimization-based methods [18]–[20] typically rely only on the projection constraint. However, due to the inherent depth ambiguity [6], several potential 3D poses could be projected to the same 2D pose. While alignment in 2D space is assured, this imperfect constraint may lead to overfitting during optimization. Additionally, previous optimization-based methods typically employ pre-trained networks to obtain a good initialization, subsequently optimizing the network parameters. Nevertheless, this approach involves a vast number of parameters to be optimized, which may disrupt the prior information embedded in the pre-trained network.

Based on the above observations, we propose an Uncertainty-Aware testing-time Optimization (UAO) framework for 3D human pose estimation. This optimization framework retains the prior knowledge embedded in the pre-trained model and alleviates the overfitting problem using the uncertainty of joints. Specifically, we design an effective network for 2D-to-3D pose lifting, and employ a separate decoder to obtain the uncertainty associated with each joint. In this way, our network can simultaneously generate the 3D results and the uncertainty of each joint based on input 2D pose during the training phase. For the testing-time optimization, our target is to make the optimized poses more reliable. Different from previous optimization-based methods [16], [17], [19]–[22] that update the network parameters based on specific constraints during test time, our designed optimization strategy freezes the network parameters and optimizes a latent state. This design has two benefits: a) fixing the network parameters can keep the learned pose prior during the optimization process; b) setting the optimization target as a hidden state instead of

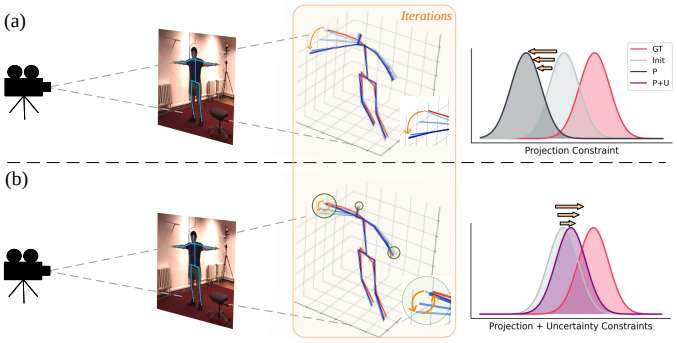

Fig. 1. Schematic diagram of the iterative optimization process. The red pose represents the ground truth, and the blue pose represents the prediction; darker colors indicate later iterations. 'GT' refers to the ground truth 3D pose distribution, 'Init' denotes the initial result, 'P' represents the projection constraint, and 'P+U' combines projection and uncertainty constraints. (a) Using only the projection constraint during test-time optimization can lead to significant deviations from the ground truth. (b) Incorporating the uncertainty constraint effectively mitigates this issue, yielding final poses that more closely match the ground truth.

the network parameter only requires the storage of a small number of parameters, making it more suitable for online inference scenarios. Furthermore, our optimization process incorporates a set of well-designed constraints to guide the optimization direction. For 2D pose alignment, we adopt the camera intrinsic parameters to project the generated 3D pose onto the 2D space and minimize the projection loss. Since we do not know the depth information, achieving precise 3D alignment remains challenging. We observe that relying solely on projection constraint can lead to over-optimization problems, causing the optimized 3D pose to deviate further from the ground truth. To alleviate this, we introduce an uncertainty constraint that allows 3D joint with higher uncertainty to have a larger divergence radius relative to its initial 3D position, while keeping joint with smaller uncertainty closer to its initial 3D position. As shown in Figure 1, relying solely on projection constraint may inadvertently steer optimization in the wrong direction, leading to poorer results. This issue is effectively mitigated by further incorporating uncertainty constraints. With such a unique design, the optimization process is prevented from generating physically implausible poses, leading to more realistic and accurate results.

Our contributions can be summarized as follows:

- We propose a novel Uncertainty-Aware testing-time Optimization (UAO) framework for 3D human pose estimation. First, an effective 2D-to-3D lifting network is designed to generate 3D outputs while estimating the joint uncertainties. During testing, the pre-trained model is frozen to preserve pose priors, and the uncertainty of each joint is utilized to alleviate the overfitting problem.
- For test time optimization, a projection constraint is used to ensure that the generated results are aligned in the 2D space. Additionally, we design the uncertainty constraint to determine the degree to which each joint can be optimized, which alleviates the overfitting problem.
- Extensive experiments demonstrate the effectiveness of the UAO framework and its potential applicability for real-world scenarios.

II. Related Work

Existing single-view 3D human pose estimation methods fall into two primary categories: a) direct estimation [23]–[25]: infer a 3D human pose from 2D images without the need for intermediate 2D pose representation, b) 2D-to-3D lifting [8], [26]–[28]: infer a 3D human pose from an intermediately estimated 2D pose. Benefiting from the success of 2D pose detectors [29], [30], the 2D-to-3D lifting approaches generally outperform the direct estimation approaches in both efficiency and effectiveness, becoming the mainstream approach. In this work, our approach follows the 2D-to-3D lifting pipeline, which can be further categorized into data-driven and optimization-based methods.

A. Data-driven Methods

Data-driven approaches involve training a deep neural network to directly regress 3D human poses from provided 2D poses. Early attempts simply use fully-connected networks (FCNs) to lift the 2D keypoint to 3D space [4]. Since the human skeleton can be naturally represented as a graph, various graph convolutional networks (GCNs) based methods [7], [8], [31]–[33] have been proposed. Recently, transformer-based methods [5], [6], [34]–[38] have been widely applied to 3D human pose estimation, achieving new breakthroughs. Although these methods can effectively learn pose patterns from datasets, they often exhibit poor generalization ability when faced with unseen data.

B. Optimization-based Methods

During the early stages, researchers employed pure optimization-based methods [11]–[13] for 3D human pose estimation. In the era of deep learning, there are also some methods that integrate optimization strategies. We categorize the existing optimization-based methods into two classes. 1) The first class incorporates a regression-based network with optimization strategies, training the entire structure end-to-end, such as SPIN [18] and PyMAF [39]. 2) The second class follows a post-optimization manner, which is more similar to us, using a pre-trained network to obtain a good initialization, including ISO [20], BOA [17], DynaBOA [16], EFT [19], ESCAPE [21] and CycleAdapt [22]. However, all these previous post-optimization methods typically optimize the network parameters based on specific constraints, which may disrupt the deep prior knowledge obtained from datasets. Some approaches also directly optimize the pose and shape parameters of parametric human models (*e.g.,* SMPL and SMPL-X), such as SMPLify [40], SMPLify-X [41], KBody [42], and HuMoR [43]. But they are constrained by inaccurate initial values or limited representational space. As a comparison, we set the optimization variable as the predicted 2D pose in latent state, and keep the model parameters frozen during the iterative optimization process. Besides, pose-related constraints are imperative to ensure the appropriate optimization direction. Previous optimization-based methods simply rely on the

projection constraint [18]–[20]. But due to the depth ambiguity, this imperfect constraint may lead to the wrong optimization direction when over-optimized. To alleviate this problem, we further utilize the uncertainty constraint to ensure that the generated pose lies within the manifold of 3D poses.

C. Uncertainty-based Methods

Uncertainty learning has attracted considerable attention and has been applied to numerous tasks, such as object detection [44], person re-identification [45], and face recognition [46], [47]. The key idea is that by knowing the reliability of its outputs, a model can optimize its predictions for improved downstream application performance. In the domain of 2D human pose estimation, confidence scores derived from 2D heatmaps are widely used to assess the reliability of each keypoint [40], [48]. For 3D human pose estimation, several studies have explored the use of uncertainty to improve model performance. Han et al. [49] introduced an uncertainty-based framework for 3D human pose estimation to mitigate the negative impacts of joints with high ambiguous. Zhang et al. [50] modeled the depth and the 2D distribution uncertainties separately, optimizing these parameters to achieve accurate 3D poses. In our work, we define keypoint uncertainty as the degree of a keypoint's deviation from its mean position in 3D space, with a higher uncertainty value indicating a larger allowable divergence radius. Moreover, our approach goes beyond simple uncertainty estimation by incorporating uncertainty constraints into the optimization phase, thereby overcoming the limitations of relying solely on the projection constraint.

III. Preliminaries

A. Theory of Uncertainty Estimation

Assuming the 3D pose follows Gaussian distributions, we reformulate the 2D-to-3D network as a Bayesian Neural Network (BNN) [51], which estimates a Gaussian distribution of the target rather than only predicting the absolute coordinates of points. Specifically, for the 3D joints $\boldsymbol{J}_i$ of person i in the dataset, our goal is to estimate the distribution of 3D joints for that individual. During training, the objective is to maximize the likelihood of all observation pairs. Let θ denote the network's learnable parameters. The likelihood function $L(\theta)$ can be formulated as follows:

$$L(\theta) = \prod_{i,k} \mathcal{N}(\boldsymbol{J}_{i,k}|\boldsymbol{\mu}_{i,k}, \Sigma_{i,k}), \tag{1}$$

where k is the index for the k-th joint of a person, $\boldsymbol{\mu}_{i,k} = f_\mu(\theta)$ and $\Sigma_{i,k} = f_\Sigma(\theta)$ are the mean and covariance matrix of the Gaussian for the k-th joint of person x_i. f_μ and f_Σ are two network heads predicting joint's Gaussian mean and covariance. Each joint is considered as a random variable following a normal Gaussian distribution. To further elaborate, the mean $\boldsymbol{\mu}_{i,k}$ captures the most probable location of the k-th joint, while the covariance matrix $\Sigma_{i,k}$ characterizes the uncertainty of the distribution around this mean. The likelihood term models the probability of observing the actual 3D joint positions given the estimated means and covariances. In pursuit of the most accurate estimation of the 3D joint positions, we seek the maximum likelihood estimate θ^*, which aligns the estimated Gaussian distribution with the observed 3D joint positions, capturing both the most probable locations $\boldsymbol{\mu}_{i,k}$ and the uncertainty $\Sigma_{i,k}$ associated with the k-th joint. The formulation of the maximum likelihood function can be achieved by:

$$\begin{aligned}\theta^* &= \arg\max_\theta \sum_{i,k} \log \mathcal{N}(\boldsymbol{J}_{i,k}|\boldsymbol{\mu}_{i,k}, \Sigma_{i,k}) \\ &= \arg\min_\theta \frac{1}{2K} \sum_{i,k} \Big((\boldsymbol{J}_{i,k} - \boldsymbol{\mu}_{i,k})^T \Sigma_{i,k}^{-1} (\boldsymbol{J}_{i,k} - \boldsymbol{\mu}_{i,k}) \\ &\quad + \ln |\Sigma_{i,k}| \Big), \end{aligned} \tag{2}$$

where K is the total number of joints per pose. Here we simplify the covariance matrix $\Sigma_{i,k}$ as a diagonal matrix with diagonal elements $\boldsymbol{\sigma}_{i,k}^2$, then we get:

$$\theta^* = \arg\min_\theta \frac{1}{2K} \sum_{i,k} \left(\frac{||\boldsymbol{J}_{i,k} - \boldsymbol{\mu}_{i,k}||^2}{\boldsymbol{\sigma}_{i,k}^2} + \ln \boldsymbol{\sigma}_{i,k}^2 \right). \tag{3}$$

We set $\boldsymbol{s}_{i,k} = \ln \boldsymbol{\sigma}_{i,k}^2$, our target can be reformulated as:

$$\theta^* = \arg\min_\theta \frac{1}{2K} \sum_{i,k} \left(\frac{||\boldsymbol{J}_{i,k} - \boldsymbol{\mu}_{i,k}||^2}{\exp(\boldsymbol{s}_{i,k})} + \boldsymbol{s}_{i,k} \right). \tag{4}$$

Through this formulation, we can minimize the discrepancy between the predicted mean $\boldsymbol{\mu}_i$ and the ground truth $\boldsymbol{J}_i$ while accounting for the associated uncertainty $\boldsymbol{s}_i$. This approach ensures that the estimated Gaussian distribution aligns optimally with the observed 3D joints, providing a robust and reliable characterization of the underlying 3D pose distribution.

B. Graph Convolutional Network

Graph Convolutional Network (GCN) [52] is capable to capture intricate relationships and structures within graph-structured data. Consider an undirected graph as $G{=}\{V, E\}$, where V is the set of nodes, and E is the set of edges. The edges can be encoded in an adjacency matrix $A{\in}\{0,1\}^{N\times N}$. For the input X_l of the l^{th} layer, the vanilla graph convolution aggregates the features of the neighboring nodes. The output X_{l+1} of the l^{th} GCN layer can be formulated as:

$$X_{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X_l W\right), \tag{5}$$

where σ is the ReLU activation function [53], $W_l{\in}\mathbb{R}^{d_1\times d}$ is the layer-specific trainable weight matrix. $\tilde{A}{=}A{+}I_N$ is the adjacency matrix of the graph with added self-connections, where I_N is the identity matrix. Additionally, $\tilde{D}$ is the diagonal node degree matrix. By stacking multiple GCN layers, it iteratively transforms and aggregates neighboring nodes, thereby obtaining enhanced feature representations. For the task of 3D human pose estimation, GCN it can effectively capture of the semantic information of the human skeleton.

IV. Uncertainty Estimation

For 2D-to-3D lifting, data-driven methods regress the 3D human pose $\hat{\mathbf{J}}^{3D} \in \mathbb{R}^{K\times 3}$ from given 2D pose $\mathbf{J}^{2D} \in \mathbb{R}^{K\times 2}$, where K is the number of skeleton joints. They typically train

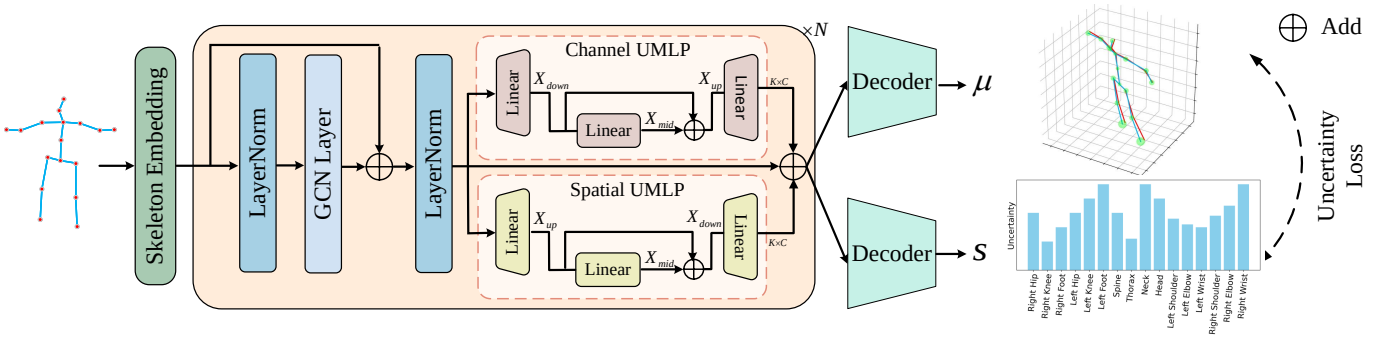

Fig. 2. Structure of the GUMLP model. Initially, the 2D input is first transformed into high-dimensional features through skeleton embedding. Subsequently, the GCN layer captures the graph information from the high-dimensional features. Then the Channel UMLP and Spatial UMLP are utilized to capture the multi-level and multi-scale information of the skeleton. Finally, two decoders are employed to predict the mean pose $\boldsymbol{\mu}$ and the uncertainty value $\boldsymbol{s}$ of the corresponding 3D pose distribution. The model is trained using uncertainty loss, which is calculated based on $\boldsymbol{\mu}$ and $\boldsymbol{s}$.

the model to learn the transformational relationship between 2D pose $\mathbf{J}^{2D}$ and ground truth 3D pose $\mathbf{J}^{3D}$ by minimizing:

$$L_{2D} = ||\mathbf{J}^{3D} - f(\mathbf{J}^{2D})||_2, \tag{6}$$

where f denotes the 2D-to-3D lifting network for 3D human pose estimation. Although this loss function allows the pre-trained model to implicitly encodes inherent patterns for 2D-to-3D lifting and acquire distribution features from the training set, it can not be aware of the uncertainty of each estimated joint. In order to enable the model to predict the uncertainty of each predicted joint, we reformulate the 2D-to-3D network as a Bayesian Neural Network (BNN) [51], which estimates a Gaussian distribution of the target rather than only predicting the absolute coordinates of points.

Specifically, for the k-th joint of the i-th person, we let the network predict the expected position $\boldsymbol{\mu}_{i,k} \in \mathbb{R}^3$ and a simplified scalar covariance $\boldsymbol{\sigma}_{i,k}$ from the input 2D pose $\mathbf{J}^{2D}$. By maximizing the likelihood of the training pairs, the loss function to train the BNN over the i-th person can be formulated as:

$$\begin{aligned} L_i^{train} &= \frac{1}{2K}\sum_{k=1}^{K}\left(\frac{\|\mathbf{J}_{i,k}^{3D} - \boldsymbol{\mu}_{i,k}\|^2}{\boldsymbol{\sigma}_{i,k}^2} + \ln \boldsymbol{\sigma}_{i,k}^2\right) \\ &= \frac{1}{2K}\sum_{k}^{K}\left(\|\mathbf{J}_{i,k}^{3D} - \boldsymbol{\mu}_{i,k}\|^2 \exp(-\boldsymbol{s}_{i,k}) + \boldsymbol{s}_{i,k}\right), \end{aligned} \tag{7}$$

where we substitute $\ln \boldsymbol{\sigma}_{i,k}^2$ with $\boldsymbol{s}_{i,k}$ for numerical stability during the training process.

To achieve this, we propose GUMLP, primarily consisting of two parts: a GCN layer and a parallel U-shaped multi-layer perception (UMLP) module, as shown in Figure 2. In contrast to the conventional data-driven model [4], [7], which typically employs only one decoder to make direct predictions, we utilize two distinct decoders to predict $\boldsymbol{\mu}$ and $\boldsymbol{s}$, respectively. The GCN layer focuses on local joint relations and introduces the topological priors of the human skeleton. For the second component, the UMLP module employs a bottleneck structure [8], [54] instead of the original MLP in the transformer [55]. The UMLP module operates in two parallel pathways: Channel UMLP and Spatial UMLP, to capture multi-scale and multi-level features effectively. In the Channel UMLP pathway, the input $\mathbf{X}$ is first processed by a down-projection layer ($\mathbf{X}_{down}$), followed by a middle layer ($\mathbf{X}_{mid}$) that maintains the same dimension, and ultimately, is fed into an up-projection layer ($\mathbf{X}_{up}$). The simplified formulation can be defined as:

$$\begin{aligned} \mathbf{X}_{down} &= \text{MLP}_{down}(\text{LN}(\mathbf{X})), \\ \mathbf{X}_{mid} &= \text{MLP}_{mid}(\mathbf{X}_{down}) + \mathbf{X}_{down}, \\ \mathbf{X}_{up} &= \text{MLP}_{up}(\mathbf{X}_{mid}) + \mathbf{X}, \end{aligned} \tag{8}$$

where $\text{MLP}(\cdot)$ consists of a linear layer and a GELU activation [56], LN denotes the Layer Normalization [57].

In the Spatial UMLP pathway, given that the number of joints ($K = 17$) is already sufficiently sparse, we implement the operations in reverse order along the spatial dimension, starting with $\mathbf{X}_{up}$, followed by $\mathbf{X}_{mid}$, and finally $\mathbf{X}_{up}$. This parallel design along both channel and spatial dimensions allows the UMLP module to extract rich feature representations across different scales and levels.

V. Testing-Time Optimization

A. Optimization Strategy

The proposed Uncertainty-Aware testing-time Optimization (UAO) framework is illustrated in Figure 3. It is essential to highlight that we only optimize the latent state $\mathbf{z} \in \mathbb{R}^{K\times 2}$ initialized by 2D input $\mathbf{J}^{2D}$. The parameters of the pre-trained model are fixed throughout the inference, which ensures the learned implicit prior from the training dataset not disrupted. Besides, we incorporate constraints into the optimization process. Specifically, projection constraint is applied to guarantee alignment of the optimized 3D pose in 2D space, and uncertainty constraint is introduced to determine the degree to which each joint can be optimized. After several iterations, the latent state $\mathbf{z}$ is refined progressively towards its optimal state. Ultimately, by feeding the optimized $\mathbf{z}^*$ into the pre-trained network, we obtain a high-quality 3D pose. The pseudo-code of the testing-time optimization is summarized in Algorithm 1.

B. Projection Constraint

Projection constraint is commonly used in optimization-based methods, such as SPIN [18] and EFT [19]. For the j-th 3D pose

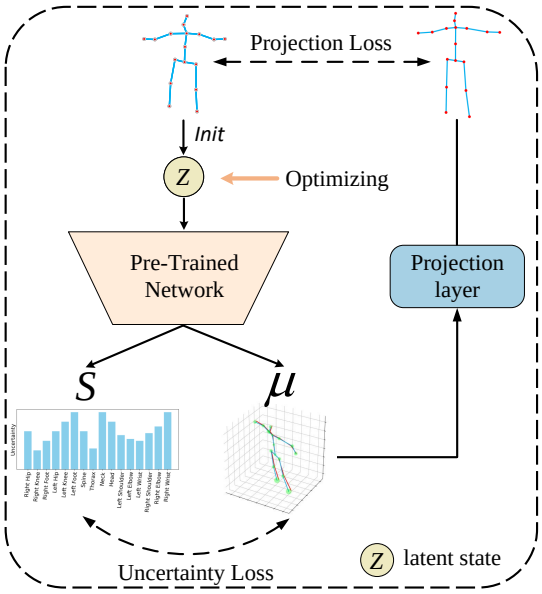

Fig. 3. Structure of the UAO framework. During the testing-time optimization, the parameters of the pre-trained model (GUMLP) are freezed. The variable being optimized is the latent state $\mathbf{z}$, which is initialized with the original 2D pose input $\mathbf{J}^{2D}$. We use the pre-trained GUMLP to obtain the mean pose $\boldsymbol{\mu}$ and uncertainty value $\boldsymbol{s}$ of the corresponding 3D pose distribution. The predicted mean pose is projected back to 2D space and compared with the input pose $\mathbf{J}^{2D}$ to calculate the projection loss. The uncertainty loss is then calculated using $\boldsymbol{\mu}$ and $\boldsymbol{s}$. Under the constraints of projection and uncertainty, the latent state $\mathbf{z}$ is optimized during the iterations.

$\mathbf{J}_j^{3D}$, its 2D projection $\mathbf{J}_j^{2D}$ should conform to the principles of perspective projection [13], which can be expressed as follow:

$$\mathbf{J}_j^{3D} \cdot \mathbf{P} = \mathbf{J}_j^{2D}, \tag{9}$$

where the projection matrix $\mathbf{P} \in \mathbb{R}^{3\times 2}$ contains the intrinsic parameters of the camera. In real scenarios, the intrinsic parameters of the camera can be obtained from camera specifications or can be inferred solely from the input images [58]. During the testing-time optimization, the 2D projection $\hat{\mathbf{J}}_j^{2D} = \hat{\mathbf{J}}_j^{3D} \cdot \mathbf{P}$ of the predicted 3D pose $\hat{\mathbf{J}}_j^{3D}$ is expected to closely align with the given 2D pose $\mathbf{J}_j^{2D}$. A large error between them indicates that the estimated 3D pose $\hat{\mathbf{J}}_j^{3D}$ may be problematic and needs to be corrected. The formulation of projection constraint $L_{P,j}^{opt}$ can be expressed as:

$$L_{P,j}^{opt} = ||\hat{\mathbf{J}}_j^{2D} - \mathbf{J}_j^{2D}||_2. \tag{10}$$

The purpose of the projection constraint is to enforce the relationship between the estimated 3D pose $\hat{\mathbf{J}}_j^{3D}$ and its corresponding 2D projection $\hat{\mathbf{J}}_j^{2D}$ to follow the geometric principles of perspective projection, thereby improving the accuracy of the pose estimation.

C. Uncertainty Constraint

Although the projection constraint can ensure the alignment of the generated pose in 2D space, it often brings over-fitting problems. This is due to the fact that multiple 3D poses might correspond to the same 2D pose after projection, which is an inherent challenge in the 3D human pose estimation task. When we solely depend on the projection constraint for the optimization process, we observe two phenomena: 1) some joints that were well-estimated deviate away from the ground truth; 2) some poorly estimated joints are initially close to the ground truth but then move away from it. To alleviate this, we introduce uncertainty constraint, which allows the well-estimated joints to stay in their original positions and mainly optimize the joints with high uncertainties.

During the testing process, for the j-th 2D input, the pre-trained model provides the initial 3D result $\hat{\boldsymbol{\mu}}_{j,k}$ and uncertainty $\hat{\boldsymbol{s}}_{j,k}$ associated with the k-th estimated 3D joint. We can explicitly formulate the uncertainty constraint $L_{U,j}^{opt}$ as:

$$L_{U,j}^{opt} = \frac{1}{2K}\sum_{k=1}^{K} \left\|\hat{\boldsymbol{\mu}}_{j,k} - f_{\boldsymbol{\mu}}^*(\mathbf{z})_{j,k}\right\|^2 \exp(-\hat{\boldsymbol{s}}_{j,k}), \tag{11}$$

where f^* denotes the pre-traind model, $f_{\boldsymbol{\mu}}^*$ denotes the predicted 3D result. This constraint enables joints with higher uncertainty less constrained by the results of the initial estimation, allowing more room for optimization. On the other hand, the joint with lower uncertainty means that the network is confident about the result, so limits it from further drifting.

Algorithm 1: Optimization Strategy

Input: 2D pose $\mathbf{J}_j^{2D}$, pre-trained model f^* with fixed parameters, projection matrix $\mathbf{P}$, optimization steps T

Output: 3D pose $\hat{\mathbf{J}}_j^{3D}$

Initialization: $\mathbf{z}_j = \mathbf{J}_j^{2D}$, $iter = 0$
$\hat{\boldsymbol{\mu}}_j = f_{\boldsymbol{\mu}}^*(\mathbf{J}_j^{2D}), \hat{\boldsymbol{s}}_j = f_{\boldsymbol{s}}^*(\mathbf{J}_j^{2D})$
while $iter < T$ **do**
  $\hat{\mathbf{J}}_j^{3D} = f_{\boldsymbol{\mu}}^*(\mathbf{z}_j)$
  $L_{P,j}^{opt} = \|\hat{\mathbf{J}}_j^{3D} \cdot \mathbf{P} - \mathbf{J}_j^{2D}\|_2$
  $L_{U,j}^{opt} = \left\|\hat{\boldsymbol{\mu}}_j - \hat{\mathbf{J}}_j^{3D}\right\|^2 / 2\exp(\hat{\boldsymbol{s}}_j)$
  $L_{total,j}^{opt} = \lambda_P \cdot L_{P,j}^{opt} + \lambda_U \cdot L_{U,j}^{opt}$
  $\mathbf{z}_j \leftarrow Adam(\mathbf{z}_j, \nabla L_{total,j}^{opt})$;
  $iter = iter + 1$
$\hat{\mathbf{J}}_j^{3D} = f_{\boldsymbol{\mu}}^*(\mathbf{z}_j)$
return $\hat{\mathbf{J}}_j^{3D}$

D. Combined Optimization Loss Function

The overall optimization loss $L_{total,j}^{opt}$ for the j-th case is defined as the weighted sum of the projection loss and the uncertainty loss:

$$L_{total,j}^{opt} = \lambda_P L_{P,j}^{opt} + \lambda_U L_{U,j}^{opt}, \tag{12}$$

where λ_P and λ_U are hyperparameters for each loss term.

VI. Experiments

A. Datasets and Evaluation Metrics

Here, we provide a more detailed description of the datasets and evaluation metrics.

TABLE I
QUANTITATIVE COMPARISON WITH THE STATE-OF-THE-ART METHODS ON HUMAN3.6M UNDER PROTOCOL 1. TOP-TABLE: 2D POSE DETECTED BY CASCADED PYRAMID NETWORK (CPN) [29] IS USED AS INPUT. BOTTOM-TABLE: GROUND TRUTH 2D POSE IS USED AS INPUT. THE TOP TWO BEST RESULTS FOR EACH ACTION ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
SimpleBaseline [4]	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
VideoPose3D [26]	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
LCN [59]	46.8	52.3	44.7	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
SRNet [60]	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
GraphSH [32]	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
MGCN [7]	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
GraFormer [61]	45.2	50.8	48.0	50.0	54.9	65.0	48.2	47.1	60.2	70.0	51.6	48.7	54.1	39.7	43.1	51.8
UGRN [62]	47.9	50.0	47.1	51.3	51.2	59.5	48.7	46.9	56.0	61.9	51.1	48.9	54.3	40.0	42.9	50.5
MLP-JCG [63]	43.8	**46.7**	46.9	48.9	50.3	60.1	**45.7**	43.9	56.0	73.7	48.9	48.2	50.9	39.9	41.5	49.7
DiffPose [64]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	49.7
ZEDO [65]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	51.4
GUMLP (Ours)	44.3	49.8	44.7	49.3	51.4	58.1	47.3	45.6	58.2	63.7	51.0	47.6	53.4	38.1	40.4	49.4
GUMLP + UAO (Ours)	**42.3**	47.5	**42.8**	**47.8**	**48.7**	**55.8**	45.8	**42.4**	**55.2**	**60.9**	**48.6**	45.9	**50.2**	**36.0**	**38.2**	**46.7**
Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
SimpleBaseline [4]	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
LCN [59]	36.3	38.8	29.7	37.8	34.6	42.5	39.8	32.5	36.2	39.5	34.4	38.4	38.2	31.3	34.2	36.3
MGCN [7]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37.4
GraFormer [61]	32.0	38.0	30.4	34.4	34.7	43.3	35.2	31.4	38.0	46.2	34.2	35.7	36.1	27.4	30.6	35.2
MLP-JCG [63]	**29.1**	36.0	30.4	33.8	35.5	46.5	35.3	31.2	39.2	48.8	33.9	35.2	35.8	**26.9**	29.4	35.1
DAF-DG [66]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	44.4
GUMLP (Ours)	30.2	36.2	30.3	32.9	33.6	41.6	35.1	31.8	38.7	42.6	34.8	36.8	37.2	28.7	30.0	34.7
GUMLP + UAO (Ours)	29.5	**34.5**	**27.5**	**31.1**	**31.8**	**38.4**	**34.0**	**29.6**	**35.5**	**38.7**	**32.9**	**35.0**	**34.9**	28.1	**29.2**	**32.7**

Human3.6M [67] is the most representative benchmark for the estimation of 3D human poses. It contains 3.6 million video frames captured from four synchronized cameras at 50Hz in an indoor environment. There are 11 professional actors performing 17 actions, such as greeting, phoning, and sitting. Following previous works [8], [26], we train our model on five subjects (S1, S5, S6, S7, S8) and test it on two subjects (S9 and S11). The performance is evaluated by two common metrics: MPJPE (Mean Per-Joint Position Error), termed Protocol 1, is the mean Euclidean distance between the predicted joints and the ground truth in millimeters. PA-MPJPE, termed Protocol 2, measures the MPJPE after Procrustes Analysis (PA) [68].

MPI-INF-3DHP [69] is a more challenging 3D pose dataset that contains both indoor and complex outdoor scenes. There are 8 actors performing 8 actions from 14 camera views, which cover a greater diversity of poses. Its test set consists of three different scenes: studio with green screen (GS), studio without green screen (noGS), and outdoor scene (Outdoor). Following [7], [26], [70], we use Percentage of Correct Keypoints (PCK) with a threshold of 150mm and the Area Under Curve (AUC) for a range of PCK thresholds for evaluation.

3DPW [71] is a complex in-the-wild dataset that provides 3D pose and mesh annotations, obtained using hand-held cameras and IMUs. It comprises 60 videos recorded at 30 frames per second. To ensure a fair comparison, we train our model on Human3.6M and test it on the 3DPW training set.

B. Implementation Details

We implement our approach with PyTorch, training and testing it on one NVIDIA RTX 3090 GPU. Following [7], [8], we use 2D pose detected by CPN [29] for Human3.6M, and ground truth 2D pose for MPI-INF-3DHP. During the training phase, we train a deep neural network for deep prior. The proposed GUMLP is designed by stacking GCN layers and UMLP modules for $N = 3$ times. For the testing time optimization, the hyperparameters for projection loss and uncertainty loss are chosen as $\lambda_P = 1$ and $\lambda_U = 0.005$, respectively.

C. Comparison with State-of-the-art Methods

Human3.6M. The comparison results with other state-of-the-art methods under Protocol 1 are shown in Table I. Following [7], [8], [32], the 2D pose detected by CPN [29] is used as input for training and testing. We have observed that our baseline model, GUMLP, produces competitive results. It is worth noting that none of the compared methods employed a post-optimization approach. To further enhance the accuracy of our model predictions, we integrate the Uncertainty-Aware Optimization (UAO) framework, which leverages both projection and uncertainty constraints. This integration significantly improved GUMLP's performance, reducing the error from 49.4mm to 46.7mm in MPJPE, **a relative improvement of 5.5%**. For Protocol 2, we also obtain the best overall results, as shown in Table II. These results validate the effectiveness of the designed 2D-to-3D lifting network (GUMLP) and the UAO framework. To further explore the lower bounds of our approach, we compare it with previous state-of-the-art methods with ground truth 2D pose as input. As shown in Table I (bottom), we achieve state-of-the-art performance, outperforming all other methods. Additionally, we observe that the performance of GUMLP can be further enhanced with the UAO framework, even when the latent state $\mathbf{z}$ is initialized with the ground truth 2D pose. This improvement is attributed to the limited capacity of the pre-trained GUMLP and the effective enhancement of 3D pose quality through the application of projection and uncertainty constraints during the optimization process. This indicates hat optimizing the latent state $\mathbf{z}$ aims to refine the 3D pose, rather than simply making $\mathbf{z}$ closer to the ground truth 2D pose.

TABLE II
QUANTITATIVE COMPARISON WITH THE STATE-OF-THE-ART METHODS ON HUMAN3.6M UNDER PROTOCOL 2. SMPLIFY-X [41] USES 2D POSE DETECTED BY OPENPOSE [72] AS INPUT, WHEREAS OTHER METHODS USE 2D POSES DETECTED BY CASCADED PYRAMID NETWORK (CPN) [29]. § DENOTES THE METHODS THAT USE REFINEMENT MODULE [7], [8]. THE TOP TWO BEST RESULTS FOR EACH ACTION ARE HIGHLIGHTED IN BOLD AND UNDERLINED, RESPECTIVELY.

Protocol 2	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
SMPLify-X [41]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	75.9
SimpleBaseline [4]	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
VideoPose3D [26]	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
LCN [59]	36.9	41.6	38.0	41.0	41.9	51.1	38.2	37.6	49.1	62.1	43.1	39.9	43.5	32.2	37.0	42.2
STGCN [8]§	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
SRNet [60]	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
MGCN [7]§	35.7	38.6	36.3	40.5	**39.2**	44.5	37.0	35.4	46.4	51.2	40.5	**35.6**	41.7	30.7	33.9	39.1
MLP-JCG [63]	**33.7**	**37.4**	37.3	39.6	39.8	47.1	**33.7**	**33.8**	**45.7**	60.5	39.7	37.7	40.1	30.1	33.8	39.3
GKONet [73]	35.4	38.8	35.9	40.4	39.6	**44.0**	36.7	35.4	46.8	53.7	40.9	36.6	42.0	30.6	33.9	39.4
ZEDO [65]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	42.1
GUMLP (Ours)	35.3	38.5	35.9	40.2	40.3	44.9	36.3	35.0	46.2	51.8	41.2	35.8	41.7	30.3	33.9	39.2
GUMLP + UAO (Ours)	34.9	38.3	**35.3**	40.2	40.1	44.9	36.5	34.8	46.0	51.9	40.8	36.0	42.3	**30.0**	**33.4**	**39.0**

TABLE III
QUANTITATIVE COMPARISONS WITH STATE-OF-THE-ART METHODS ON THE MPI-INF-3DHP DATASET.

Method	GS ↑	noGS ↑	Outdoor ↑	All PCK ↑	All AUC ↑
SimpleBaseline [4]	49.8	42.5	31.2	42.5	17.0
GraphSH [32]	81.5	81.7	75.2	80.1	45.8
MGCN [7]	86.4	86.0	85.7	86.1	53.7
GraFormer [61]	80.1	77.9	74.1	79.0	43.8
UGRN [62]	86.2	84.7	81.9	84.1	53.7
GUMLP	87.6	85.9	89.3	87.4	54.8
GUMLP + UAO	**88.0**	**86.9**	**89.8**	**88.0**	**55.1**

TABLE IV
CROSS-DOMAIN EVALUATION ON 3DPW. SOURCE DATASET: HUMAN3.6M, TARGET DATASET: 3DPW.

Method	PA-MPJPE ↓	MPJPE ↓
ISO [20]	70.8	-
Wang *et al.* [74]	68.3	109.5
PoseAug [9]	58.5	94.1
PoseDA [75]	55.3	87.7
PMCE [76]	52.3	81.6
AdaptPose [10]	46.5	81.2
ZEDO [65]	42.6	80.9
BOA [17]	49.5	77.2
DynaBOA [16]	40.4	65.5
DAPA [77]	46.5	75.0
CycleAdapt [22]	**39.9**	64.7
ESCAPE [21]	47.9	79.0
GUMLP (Ours)	42.0	67.5
GUMLP + UAO (Ours)	41.1	**60.8**

MPI-INF-3DHP. To evaluate the generalization ability of the proposed UAO framework, we further compare our approach against previous state-of-the-art methods on cross-dataset scenarios. As shown in Table III, GUMLP obtains the best results in all scenes and all metrics, consistently surpassing other methods. After utilizing the UAO framework, the performance can be further improved in both indoor and outdoor scenarios. This suggests that the designed testing-time optimization process can effectively enhance the model's ability to generalize to unknown actions and datasets.

3DPW. We further evaluated our model's performance on the 3DPW dataset, which features complex outdoor scenes. As shown in Table IV, our baseline model, GUMLP, outperforms previous methods in both PA-MPJPE and MPJPE metrics. Furthermore, the integration of the UAO framework improves performance by reducing MPJPE by 6.7mm. This result highlights the effectiveness of our proposed optimization strategy in outdoor environments.

D. Ablation Study

Impact of Model Design. The GUMLP model primarily comprises two components: a GCN layer and a parallel U-shaped multi-layer perception (UMLP) module, as illustrated in Figure 2. To investigate the design choices of GUMLP, experiments were conducted on Human3.6M using the 2D poses extracted by CPN [29] as input. The results are reported in Table V. Our findings indicate that using only the spatial UMLP, which captures multi-scale features, or only the channel UMLP, which captures multi-level features, does not yield the best performance. The optimal results are achieved when both modules are used together, demonstrating that a combination of multi-level and multi-scale features is necessary for optimal performance.

Impact of Model Configurations. We conducted ablation studies by exploring the performance of GUMLP with different hyperparameters. The results are presented in Table VI. We observe that with a fixed number of layers L, the model consistently achieves optimal performance with a hidden size C of 512. Doubling or halving the hidden size results in performance degradation.This is because too few parameters lead to underfitting, while too many cause overfitting and hinder convergence. With a hidden size C of 512, we determined that the optimal number of layers is $L = 3$ for the best performance. In Table VII, we further explore the performance of Spatial UMLP with different spatial dimension S_{mid} after the up-projection layer. The results show that when $S_{mid} = 17 * 6$, our model achieves the best performance.

Impact of the Initialization Prior. In Table VIII, we use a template 2D pose to initialize the latent state $\mathbf{z}$, treating this as a condition lacking an effective initialization prior. Rows 1 and 4 correspond to general data-driven approaches, where the 2D pose is directly fed into a fixed network to generate an estimated 3D pose without any further optimization. Comparing

TABLE V
ABLATION STUDY ON DIFFERENT DESIGNS OF OUR MODEL.

GCN	Channel UMLP	Spatial UMLP	MPJPE ↓
✓	✓		50.5
✓		✓	56.2
✓	✓	✓	49.4

TABLE VI
ABLATION STUDY ON VARIOUS CONFIGURATIONS OF GUMLP. L DENOTES THE NUMBER OF LAYERS, C DENOTES THE HIDDEN SIZE.

L	C	Params (M)	FLOPs (M)	MPJPE ↓
2	256	1.55	25.27	52.1
2	512	5.91	76.44	50.8
2	1024	23.33	256.49	54.0
3	256	1.92	37.25	50.8
3	512	7.32	112.1	**49.4**
3	1024	28.87	374.61	50.6
4	256	2.29	49.23	50.5
4	512	8.72	147.76	49.6
4	1024	34.41	492.72	51.4

rows 1–3 with rows 4–6, it is evident that a good initialization is crucial for achieving favorable convergence. Even the worst-case scenario of using the input 2D pose for initialization performs better than cases without an initialization prior. These results emphasize the significance of an effective initialization strategy in improving overall performance.

Impact of Constraints during Optimization. Setting appropriate targets for test-time optimization is essential for achieving high-quality 3D poses. To this end, we evaluate the effectiveness of two proposed constraints in the optimization process. Table VIII presents the ablation results using the UAO framework on the Human3.6M test set with different constraints. Comparing rows 2–3 with rows 5–6, the results show that the projection constraint significantly enhances the performance of the purely data-driven method. Furthermore, the introduction of the uncertainty constraint further improves the model's generalization ability, leading to superior performance. These results highlight the effectiveness of both constraints in the optimization process. To further explore the optimization mechanism, we illustrate the performance with different iterations on the test set of Human3.6M, as shown in Figure 4. We observe that using only the projection constraint leads to a rapid decrease in MPJPE, followed by a gradual increase. In contrast, the error can be stabilized at a lower level with the further introduction of the uncertainty constraint, even with more iterations. The main reason is that the projection

TABLE VII
ABLATION STUDY ON SPATIAL UMLP. L DENOTES THE NUMBER OF LAYERS, C DENOTES THE HIDDEN SIZE. S_{mid} DENOTES THE SPATIAL DIMENSIONS AFTER THE UP-PROJECTION LAYER.

L	C	S_{mid}	Params (M)	FLOPs (M)	MPJPE ↓
3	512	17×1	7.21	78.8	50.0
3	512	17×3	7.23	86.7	50.1
3	512	17×6	7.32	112.1	**49.4**
3	512	17×9	7.45	153.1	50.0

TABLE VIII
ABLATION STUDY ON INITIALIZATION PRIOR AND CONSTRAINTS.

Row Index	Initialization Manner	Initialization Prior	Projection Constraint	Uncertainty Constraint	MPJPE ↓
1	2D Template Pose				351.8
2			✓		93.1
3			✓	✓	69.5
4	Input 2D Pose	✓			49.4
5		✓	✓		47.1
6		✓	✓	✓	46.7

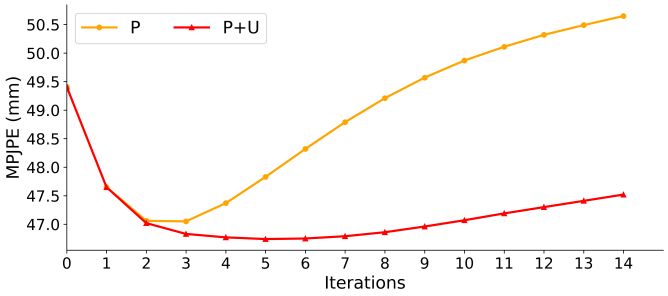

Fig. 4. Performance with different iterations on Human3.6M test set. Here we adopt the pre-trained GUMLP as baseline and use the 2D pose detected by CPN [29] to initialize the latent state $\mathbf{z}$. $\mathbf{P}$ and $\mathbf{U}$ represent projection constraint and uncertainty constraint, respectively.

constraint only enforces consistency in the 2D plane, which is easy to overfit, yielding unrealistic poses in the 3D space. To complement this, the uncertainty constraint helps maintain the output within a plausible 3D manifold. It restricts joints with low uncertainty from deviating too far from the ground truth while allowing joints with high uncertainty to have a larger divergence radius. Experimental results demonstrate the robust adaptability of our method, especially in real-world scenarios where the optimal number of iterations is uncertain.

Inference speed of different Optimization Strategies. As is well known, optimization-based methods are more time-consuming. We evaluated the inference speed of various optimization strategies on a single frame using one GeForce RTX 3090, with results shown in Table IX. The baseline models

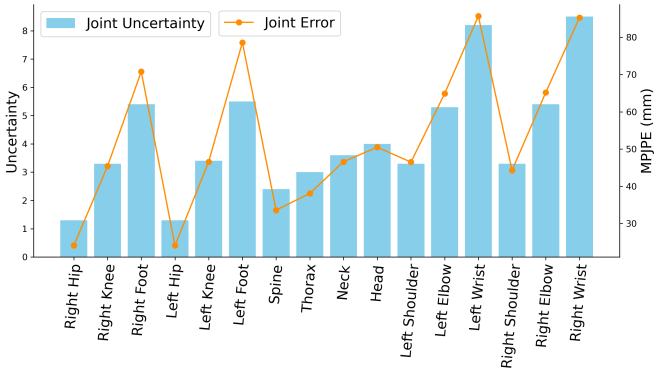

Fig. 5. Combination of a histogram representing the uncertainty for each joint and a line chart depicting the mean error (MPJPE) for each joint. The left y-axis denotes the magnitude of uncertainty, while the right y-axis signifies the error value in MPJPE.

TABLE IX
INFERENCE SPEED OF OPTIMIZATION METHODS. UAO* SIGNIFIES EMPLOYING THE NETWORK'S LATENT STATE AS THE VARIABLE FOR OPTIMIZATION.

Method	FPS
GUMLP (Ours)	310.0
SPIN [18]	60.2
SPIN + BOA [17]	1.2
SPIN + DynBOA [16]	0.9
SPIN + DAPA [77]	2.3
SPIN + CycleAdapt [22]	13.5
SPIN + ESCAPE [21]	43.5
GUMLP + UAO (Ours)	48.0
GUMLP + UAO* (Ours)	**125.0**

include our GUMLP and the human mesh reconstruction method SPIN [18]. Although these baselines differ, their inference times are relatively fast and have minimal impact on overall runtime. Since the primary computational cost lies in the optimization stage, we can fairly compare the runtime of different methods. Our baseline model, GUMLP, reaches **310** fps. By setting the number of iterations to 5, our UAO framework reaches **48** fps, surpassing all the other optimization methods. This improvement is attributed to our UAO framework optimizing the latent state $\mathbf{z}$ rather than the entire model parameters, thereby accelerating the inference process. It is worth noting that the output of any layer within the network can be selected as the latent state for optimization. For faster processing while maintaining satisfactory performance, we select the latent state before the final fully connected layer as the optimized variable, denoted as UAO*. This choice significantly enhances the inference speed, achieving up to **125** fps.

E. Uncertainty and Error for each Joint

Figure 5 shows the average uncertainty and average error for each joint estimation across the entire test set, using the GUMLP model pre-trained on the human3.6M dataset. For uncertainty estimation, we observe that the joints at the end of limbs often exhibit significant predicted uncertainty, such as the left wrist and the left foot. This phenomenon is attributed to the fact that joints at the ends of the limbs typically experience higher movement velocities, presenting challenges in accurate prediction. Moreover, the line chart illustrating joint errors follows a trend similar to the histogram, indicating that joints with larger uncertainties generally exhibit higher prediction errors. This further validates the effectiveness of GUMLP in predicting joint uncertainties.

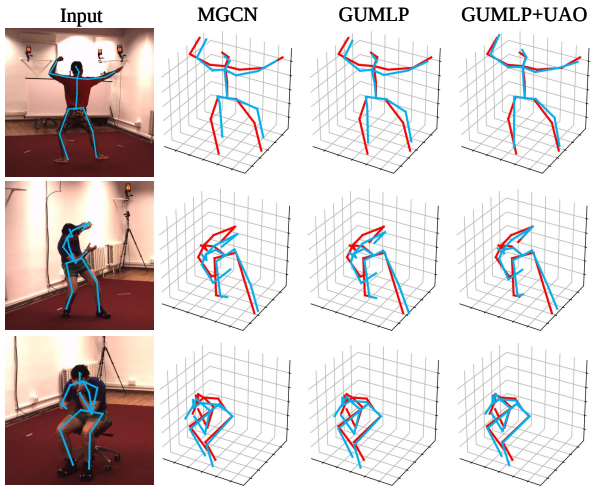

Fig. 6. Quantitative comparison of MGCN [7], GUMLP, and GUMLP with UAO on Human3.6M. The blue pose represents the predicted results, while the red pose represents the ground truth.

TABLE X
PERFORMANCE ACROSS DIFFERENT VIEWS ON HUMAN3.6M IN MPJPE. IN-DOMAIN REFERS TO TESTING UNDER THE SAME VIEW AS THE TRAINING, WHILE OUT-OF-DOMAIN REFERS TO TESTING UNDER VIEWS OTHER THAN THE TRAINING VIEW. P_i DENOTES THE iTH CAMERA VIEW.

Training views	In Domain			Out-of-Domain		
	GUMLP	+ UAO	Δ	GUMLP	+ UAO	Δ
P_1	55.3	53.2	2.1	87.0	82.5	4.5
P_1, P_2	49.9	47.4	2.5	62.5	57.3	5.2
P_1, P_2, P_3	50.3	47.6	2.7	59.6	54.0	5.6
P_1, P_2, P_3, P_4	49.4	46.7	2.7	-	-	-

F. Generalization to Unseen Camera Views

Table X presents the relative gains of the proposed approach over in- and out-domain cases on Human3.6M, where four camera views are available. It can be inferred that the UAO framework can not only help the in-domain cases but also bring more significant improvements to unseen camera views. This verifies its ability to break the domain gap and meet our design expectations. Besides, from the last column of the table, a large training set can bring more optimization gains, which may be explained by better pre-training leading to a better depiction of the 3D pose manifold.

G. Qualitative Results

The visualization results on the Human3.6M dataset are shown in Figure 6. We can see that our original GUMLP is marginally superior to the previous state-of-the-art method, MGCN [7]. Moreover, employing the UAO framework brings better generation with more precise poses. For example, the 3D pose of photoing action (row 2) predicted by GUMLP with UAO (col 4) is similar to the ground truth. As a comparison, the left arms predicted by MGCN and the GUMLP are at a lower position. We further visualize the iterative process of the UAO framework under different constraints, as shown in Figure 7. The results in Figure 7 (row 1) show that relying solely on the projection constraint can cause some well-estimated joints to deviate from the ground truth. However, for the same 2D pose, incorporating the uncertainty constraint (Figure 7 (row 2)) results in an optimized pose that better aligns with the ground truth. This further validates the effectiveness of the optimization process and highlights the necessity of our uncertainty constraint.

To further demonstrate the performance in outdoor scenes, we present the qualitative results on the 3DPW test set using 2D pose detected by HRNet [48], as illustrated in Figure 8. The

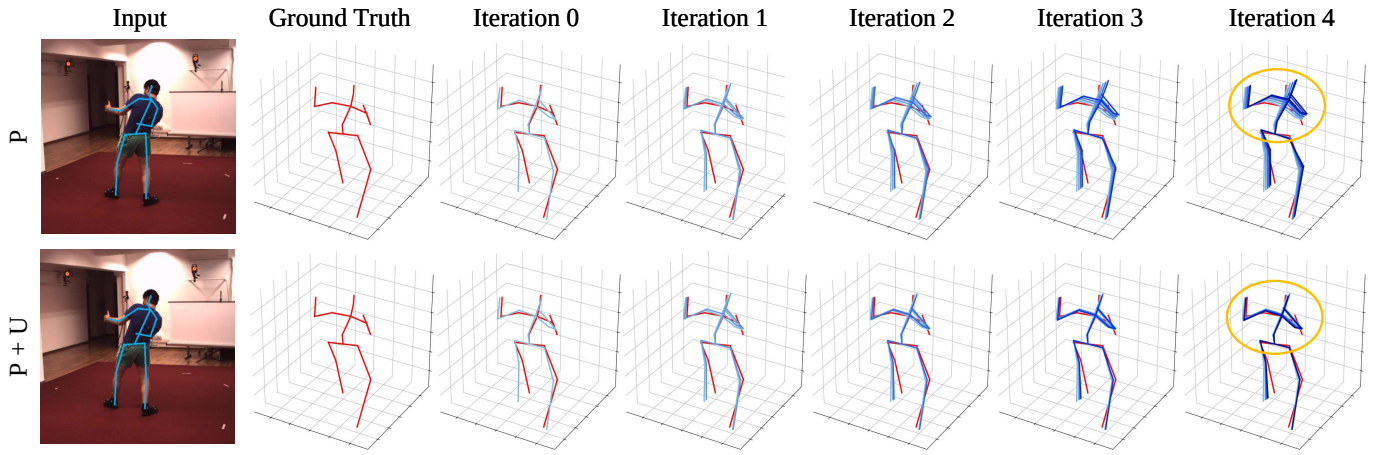

Fig. 7. Visualization of the iterative optimization process. The red pose is the ground truth and the blue pose is our prediction. Color intensity reflects the number of iterations, with darker colors indicating more iterations. The initial 3D pose generated by the pre-trained model (column 3) progressively improves through iterative optimization (columns 4-7). "P" indicates the use of only the projection constraint, "P+U" denotes the combined use of projection and uncertainty constraints.

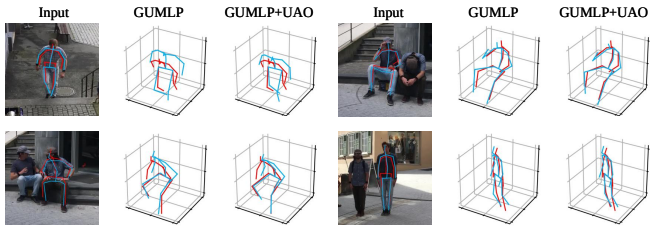

Fig. 8. Visualization results on 3DPW using 2D pose detected by HRNet [48]. The red and blue poses in the Input image represent the ground truth and detected 2D poses, respectively. In the 3D pose columns (2–3), the red pose indicates the ground truth 3D pose, while the blue 3D pose corresponds to the predicted results based on the detected 2D input.

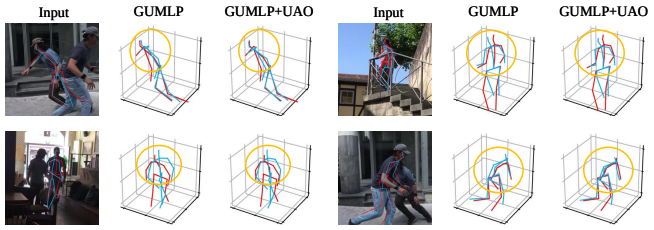

Fig. 9. Occluded samples in 3DPW with 2D poses detected by HRNet [48]. The red and blue poses in the Input image represent the ground truth and detected 2D poses, respectively. In the 3D pose columns (2–3), the red pose indicates the ground truth 3D pose, while the blue 3D pose corresponds to the predicted results based on the detected 2D input.

results indicate that the initial 3D outputs generated by GUMLP are significantly enhanced when using the UAO framework. Figure 9 also shows some occluded samples. Although our optimization strategy (UAO) is not specifically designed for occluded poses, it still performs well even in the presence of occluded body parts.

VII. Conclusion

In this paper, we introduced an Uncertainty-Aware testing-time Optimization (UAO) framework for 3D human pose estimation. During training, our proposed GUMLP network estimates 3D poses along with uncertainty values for each joint. At test time, the UAO framework freezes the pre-trained network parameters and optimizes a latent state initialized by the input 2D pose. To effectively constrain the optimization in both 2D and 3D spaces, we apply projection and uncertainty constraints. Extensive experiments show that our approach achieves state-of-the-art results on popular datasets and demonstrates superior generalization in challenging outdoor scenes.

References

[1] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. Pose-guided human animation from a single image in the wild. In *CVPR*, pages 15039–15048, 2021.
[2] Mikael Svenstrup, Soren Tranberg, Hans Jorgen Andersen, and Thomas Bak. Pose estimation and adaptive robot behaviour for human-robot interaction. In *ICRA*, pages 3571–3576, 2009.
[3] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.
[4] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3D human pose estimation. In *ICCV*, pages 2640–2649, 2017.
[5] 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 *CVPR*, pages 13232–13242, 2022.
[6] Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. MHFormer: Multi-hypothesis transformer for 3D human pose estimation. In *CVPR*, pages 13147–13156, 2022.
[7] Zhiming Zou and Wei Tang. Modulated graph convolutional network for 3d human pose estimation. In *ICCV*, pages 11477–11487, 2021.
[8] 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 *ICCV*, pages 2272–2281, 2019.
[9] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. PoseAug: A differentiable pose augmentation framework for 3D human pose estimation. In *CVPR*, pages 8575–8584, 2021.
[10] Mohsen Gholami, Bastian Wandt, Helge Rhodin, Rabab Ward, and Z Jane Wang. Adaptpose: Cross-dataset adaptation for 3D human pose estimation by learnable motion generation. In *CVPR*, pages 13075–13085, 2022.
[11] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3D human pose from 2D image landmarks. In *ECCV*, pages 573–586, 2012.
[12] Edgar Simo-Serra, Arnau Ramisa, Guillem Alenya, Carme Torras, and Francesc Moreno-Noguer. Single image 3D human pose estimation from noisy observations. In *CVPR*, pages 2673–2680, 2012.

[13] Chunyu Wang, Yizhou Wang, Zhouchen Lin, Alan L Yuille, and Wen Gao. Robust estimation of 3D human poses from a single image. In *CVPR*, pages 2361–2368, 2014.

[14] Zhao Liu, Jianke Zhu, Jiajun Bu, and Chun Chen. A survey of human pose estimation: the body parts parsing based methods. *Journal of Visual Communication and Image Representation*, 32:10–19, 2015.

[15] Mira Kim, Youngjo Min, Jiwon Kim, and Seungryong Kim. Meta-learned initialization for 3D human recovery. In *ICIP*, pages 4238–4242, 2022.

[16] Shanyan Guan, Jingwei Xu, Michelle Z He, Yunbo Wang, Bingbing Ni, and Xiaokang Yang. Out-of-domain human mesh reconstruction via dynamic bilevel online adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[17] Shanyan Guan, Jingwei Xu, Yunbo Wang, Bingbing Ni, and Xiaokang Yang. Bilevel online adaptation for out-of-domain human mesh reconstruction. In *CVPR*, pages 10472–10481, 2021.

[18] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *ICCV*, pages 2252–2261, 2019.

[19] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3D human model fitting towards in-the-wild 3D human pose estimation. In *3DV*, pages 42–52. IEEE, 2021.

[20] Jianfeng Zhang, Xuecheng Nie, and Jiashi Feng. Inference stage optimization for cross-scenario 3D human pose estimation. In *NeurIPS*, pages 2408–2419, 2020.

[21] Luke Bidulka, Mohsen Gholami, Jiannan Zheng, Martin J McKeown, and Z Jane Wang. Escape: Energy-based selective adaptive correction for out-of-distribution 3d human pose estimation. *Neurocomputing*, 611:128605, 2025.

[22] Hyeongjin Nam, Daniel Sungho Jung, Yeonguk Oh, and Kyoung Mu Lee. Cyclic test-time adaptation on monocular video for 3d human mesh reconstruction. In *ICCV*, pages 14829–14839, 2023.

[23] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, pages 529–545, 2018.

[24] Xiaoxuan Ma, Jiajun Su, Chunyu Wang, Hai Ci, and Yizhou Wang. Context modeling in 3D human pose estimation: A unified perspective. In *CVPR*, pages 6238–6247, 2021.

[25] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *CVPR*, pages 7025–7034, 2017.

[26] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, pages 7753–7762, 2019.

[27] Kai Zhai, Qiang Nie, Bo Ouyang, Xiang Li, and ShanLin Yang. Hopfir: Hop-wise graphformer with intragroup joint refinement for 3d human pose estimation. *arXiv preprint arXiv:2302.14581*, 2023.

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

[29] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, pages 7103–7112, 2018.

[30] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, pages 5693–5703, 2019.

[31] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3D human pose regression. In *CVPR*, pages 3425–3435, 2019.

[32] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3D human pose estimation. In *CVPR*, pages 16105–16114, 2021.

[33] Md Tanvir Hassan and A Ben Hamza. Regular splitting graph network for 3d human pose estimation. *IEEE Transactions on Image Processing*, 2023.

[34] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3D human pose estimation with spatial and temporal transformers. In *ICCV*, pages 11656–11665, 2021.

[35] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3D human pose estimation. *IEEE Transactions on Multimedia*, pages 1282–1293, 2022.

[36] Yuanhong Zhong, Guangxia Yang, Daidi Zhong, Xun Yang, and Shanshan Wang. Frame-padded multiscale transformer for monocular 3d human pose estimation. *IEEE Transactions on Multimedia*, 2023.

[37] Ruibin Wang, Xianghua Ying, and Bowei Xing. Exploiting temporal correlations for 3d human pose estimation. *IEEE Transactions on Multimedia*, 2023.

[38] Yong Wang, Hongbo Kang, Doudou Wu, Wenming Yang, and Longbin Zhang. Global and local spatio-temporal encoder for 3d human pose estimation. *IEEE Transactions on Multimedia*, 2023.

[39] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3D human pose and shape regression with pyramidal mesh alignment feedback loop. In *ICCV*, pages 11446–11456, 2021.

[40] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, pages 561–578, 2016.

[41] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, pages 10975–10985, 2019.

[42] Nikolaos Zioulis and James F O'Brien. Kbody: Towards general, robust, and aligned monocular whole-body estimation. In *CVPR*, pages 6215–6225, 2023.

[43] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *ICCV*, pages 11488–11499, 2021.

[44] Zhenyu Wang, Yali Li, Ye Guo, Lu Fang, and Shengjin Wang. Data-uncertainty guided multi-phase learning for semi-supervised object detection. In *CVPR*, pages 4568–4577, 2021.

[45] Tianyuan Yu, Da Li, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. Robust person re-identification by modelling feature uncertainty. In *ICCV*, pages 552–561, 2019.

[46] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *CVPR*, pages 5710–5719, 2020.

[47] Yuhang Zhang, Chengrui Wang, and Weihong Deng. Relative uncertainty learning for facial expression recognition. In *NeurIPS*, volume 34, pages 17616–17627, 2021.

[48] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3349–3364, 2020.

[49] Chuchu Han, Xin Yu, Changxin Gao, Nong Sang, and Yi Yang. Single image based 3d human pose estimation via uncertainty learning. *Pattern Recognition*, 132:108934, 2022.

[50] Jinlu Zhang, Yujin Chen, and Zhigang Tu. Uncertainty-aware 3d human pose estimation from monocular video. In *ACM MM*, pages 5102–5113, 2022.

[51] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Bennamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.

[52] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[53] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[56] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[57] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[58] Scott Workman, Connor Greenwell, Menghua Zhai, Ryan Baltenberger, and Nathan Jacobs. Deepfocal: A method for direct focal length estimation. In *ICIP*, pages 1369–1373. IEEE, 2015.

[59] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3D human pose estimation. In *ICCV*, pages 2262–2271, 2019.

[60] 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 *ECCV*, pages 507–523, 2020.

[61] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. GraFormer: Graph-oriented transformer for 3D pose estimation. In *CVPR*, pages 20438–20447, 2022.

[62] Han Li, Bowen Shi, Wenrui Dai, Hongwei Zheng, Botao Wang, Yu Sun, Min Guo, Chenglin Li, Junni Zou, and Hongkai Xiong. Pose-oriented transformer with uncertainty-guided refinement for 2D-to-3D human pose estimation. In *AAAI*, volume 37, pages 1296–1304, 2023.

[63] Zhenhua Tang, Jia Li, Yanbin Hao, and Richang Hong. Mlp-jcg: Multi-layer perceptron with joint-coordinate gating for efficient 3d human pose

estimation. *IEEE Transactions on Multimedia*, 2023.
[64] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3D pose estimation. In *CVPR*, pages 13041–13051, 2023.
[65] Zhongyu Jiang, Zhuoran Zhou, Lei Li, Wenhao Chai, Cheng-Yen Yang, and Jenq-Neng Hwang. Back to optimization: Diffusion-based zero-shot 3d human pose estimation. In *WACV*, pages 6142–6152, 2024.
[66] Qucheng Peng, Ce Zheng, and Chen Chen. A dual-augmentor framework for domain generalization in 3d human pose estimation. In *CVPR*, pages 2240–2249, 2024.
[67] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013.
[68] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975.
[69] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation in the wild using improved cnn supervision. In *3DV*, pages 506–516, 2017.
[70] Ailing Zeng, Xiao Sun, Lei Yang, Nanxuan Zhao, Minhao Liu, and Qiang Xu. Learning skeletal graph neural networks for hard 3D pose estimation. In *ICCV*, pages 11436–11445, 2021.
[71] 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. In *ECCV*, pages 601–617, 2018.
[72] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 7291–7299, 2017.
[73] Mengxian Hu, Chengju Liu, Shu Li, Qingqing Yan, Qin Fang, and Qijun Chen. A geometric knowledge oriented single-frame 2D-to-3D human absolute pose estimation method. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.
[74] Zhe Wang, Daeyun Shin, and Charless C Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. In *ECCVW*, pages 523–540, 2020.
[75] Wenhao Chai, Zhongyu Jiang, Jenq-Neng Hwang, and Gaoang Wang. Global adaptation meets local generalization: Unsupervised domain adaptation for 3d human pose estimation. In *ICCV*, pages 14655–14665, 2023.
[76] Yingxuan You, Hong Liu, Ti Wang, Wenhao Li, Runwei Ding, and Xia Li. Co-evolution of pose and mesh for 3d human body estimation from video. In *ICCV*, pages 14963–14973, 2023.
[77] Zhenzhen Weng, Kuan-Chieh Wang, Angjoo Kanazawa, and Serena Yeung. Domain adaptive 3d pose augmentation for in-the-wild human mesh recovery. In *3DV*, pages 261–270. IEEE, 2022.

Ti Wang is a master's student in the School of Electronic and Computer Engineering at Peking University (PKU), China, under the supervision of Prof. H. Liu. He is a reviewer for the journal Pattern Recognition. His research focuses on 3D human pose and shape estimation, emphasizing optimization-based methods to enhance the accuracy and reliability of 3D modeling.

Mengyuan Liu is an Assistant Professor at Peking University. He received a Ph.D. degree in 2017 under the supervision of Prof. H. Liu from the School of EE&CS, Peking University (PKU), China. His research interests include human action recognition, human motion prediction, and human motion generation using RGB, depth, and skeleton data. Related methods have been published in T-IP, T-CSVT, T-MM, PR, CVPR, ECCV, ACM MM, AAAI, and IJCAI. He has been invited to be a Technical Program Committee (TPC) member for ACPR, ACM MM, and AAAI.

Hong Liu serves as a Full Professor in the School of EE&CS, Peking University (PKU), China. He received the Ph.D. degree in Mechanical Electronics and Automation in 1996. Prof. Liu has been selected as Chinese Innovation Leading Talent supported by "National High-level Talents Special Support Plan" since 2013. His research interests include computer vision, pattern recognition and robot motion planning. He is also reviewers for many international journals such as Pattern Recognition, IEEE Trans. on Signal Processing, and IEEE Trans. on PAMI.

Bin Ren is a Ph.D. student in the Italian National Artificial Intelligence program co-organized by the University of Pisa and the University of Trento, Italy. He received his Master's degree in 2021 at the School of Electronics and Computer Engineering, Peking University, China. He received his B.Eng. degree in 2016 at the College of Mechanical and Electrical Engineering, Central South University, China. His research interests lie in explore machine learning to low-level vision and multi-modal LLMs.

Yingxuan You is a master's student in the School of Electronic and Computer Engineering at Peking University (PKU), China, under the supervision of Prof. H. Liu. Her primary research interests include 3D human pose and shape estimation, 3D computer vision, and generative models. with related methods published in top conferences like ICCV, ACM MM, ICASSP, and IROS.

Wenhao Li received the B.E. degree in automation in 2019. He is currently working toward the Ph. D. degree in the School of Computer Science, Peking University (PKU), China, under the supervision of Prof. Hong Liu. His research interests lie in deep learning, machine learning, and their applications to computer vision.

Nicu Sebe is a Professor at the University of Trento, Italy, where he is leading the research in the areas of multimedia analysis and human behavior understanding. He was the General Co-Chair of the IEEE FG 2008 and ACM Multimedia 2013. He was a program chair of ACM Multimedia 2011 and 2007, ECCV 2016, ICCV 2017 and ICPR 2020. He is a general chair of ACM Multimedia 2022. He is a fellow of IAPR.

Xia Li received his master's degree from the School of Electronic and Computer Engineering, Peking University, in 2020. He is currently working toward a PhD degree with the Department of Computer Science at ETH Zurich. His research interests include medical image reconstruction, processing, and registration.