3D Human Pose Estimation with Occlusions: Introducing BlendMimic3D Dataset and GCN Refinement

Filipa Lino, Carlos Santiago, Manuel Marques
Institute for Systems and Robotics, LARSyS, Instituto Superior Técnico, Portugal
{filipa.lino, carlos.santiago}@tecnico.ulisboa.pt, manuel@isr.tecnico.ulisboa.pt

Abstract

In the field of 3D Human Pose Estimation (HPE), accurately estimating human pose, especially in scenarios with occlusions, is a significant challenge. This work identifies and addresses a gap in the current state of the art in 3D HPE concerning the scarcity of data and strategies for handling occlusions. We introduce our novel BlendMimic3D dataset, designed to mimic real-world situations where occlusions occur for seamless integration in 3D HPE algorithms. Additionally, we propose a 3D pose refinement block, employing a Graph Convolutional Network (GCN) to enhance pose representation through a graph model. This GCN block acts as a plug-and-play solution, adaptable to various 3D HPE frameworks without requiring retraining them. By training the GCN with occluded data from BlendMimic3D, we demonstrate significant improvements in resolving occluded poses, with comparable results for non-occluded ones. Project web page is available at https://blendmimic3d.github.io/BlendMimic3D/.

1. Introduction

Human pose estimation (HPE) from visual data has become crucial in computer vision, with wide-ranging applications from sports analysis to enhancing smart retail experiences. It involves interpreting a person's position and orientation from images or videos. Despite the emergence of various techniques [5, 25, 30, 35] and datasets [14, 24, 37], 3D HPE remains challenging, particularly with monocular camera views in occluded scenarios. Under occlusions, estimating 3D poses becomes even harder, due to the increased ambiguity, and the lack of datasets specifically targeting occluded scenarios makes most state-of-the-art approach struggle with this type of data.

To fill this gap, we introduce a new synthetic dataset, called BlendMimic3D[1], illustrated in Figure 1, that aims to serve as a novel benchmark for HPE with occlusions. Our dataset, built using Blender [10], comprises a variety of scenarios mirroring real-world complexities, and purposely contains several types of occlusions, including self, object-based and out-of-frame occlusions. This makes BlendMimic3D an invaluable tool for both training HPE models and benchmarking their performance in occluded scenarios.

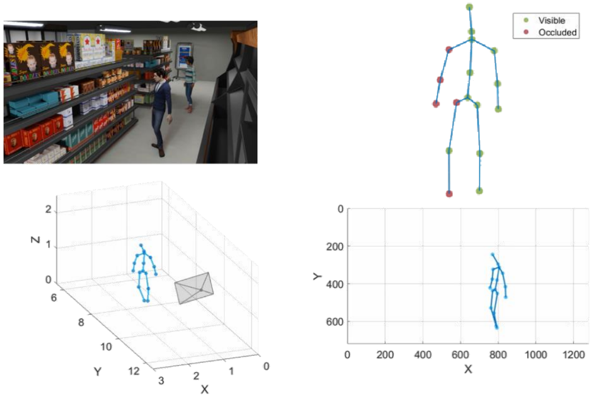

Figure 1. BlendMimic3D, our synthetic dataset for 3D HPE occlusion benchmarking, features diverse multi-camera scenarios with up to three subjects. It includes Blender animations (top left), keypoint visibility (top right), cameras' parameters, 3D poses (bottom left) and 2D pose representations (bottom right).

Additionally, this work proposes a new pose refinement module[2], designed to overcome the limitations of the current state of the art. Our approach is based on a graph convolutional network (GCN) [17] that takes into account spatial and temporal information and is compatible with various 2D-to-3D HPE backbones, including VideoPose3D [30], PoseFormerV2 [40], and D3DP [34]. It works as a plugin feature that enhances occluded keypoint estimates and does not require training or fine-tuning the HPE backbone, substantially simplifying its use.

The main contributions are the following:

[1] Available at https://github.com/FilipaLino/BlendMimic3D

[2] Available at https://github.com/FilipaLino/GCN-Pose-Refinement

- BlendMimic3D dataset: a comprehensive, realistic synthetic benchmark dataset focused on occlusions, aiding in the training and evaluation of HPE models.
- A novel GCN for 3D pose refinement, leveraging spatial-temporal keypoint relationships. It integrates with most current monocular 3D HPE methods and is designed to address occlusions without additional retraining.

Extensive evaluation with two different 2D keypoint detection algorithms [7, 39] and three state-of-the-art 2D-to-3D algorithms [30, 34, 40], validate the utility of our new benchmark dataset and the efficacy of our proposed refinement module in estimating poses in occluded conditions.

2. Related Work

Advances in deep learning, especially Convolutional Neural Networks (CNNs) [26], have significantly improved HPE, offering enhanced accuracy and speed. Toshev et al.'s DeepPose [36] exemplifies the potential of these methods. In HPE, three primary body modeling approaches are used: kinematic (body keypoints) [1, 2, 14–16, 22, 24, 37]; planar (body contours) [8, 37]; and volumetric (3D meshes) [2, 3, 14, 24, 37]. Our work focuses on the kinematic model due to its versatility.

2.1. From 2D to 3D Transition

When estimating 3D human pose from 2D video inputs, direct 3D estimation [21, 29] is challenging due to the loss of depth information in 2D representations. Instead, researchers have found it more effective to first extract the 2D pose and then infer the corresponding 3D pose [25, 35].

Lee and Chen [19] were early pioneers in 2D joints projection into 3D spaces, but the arrival of deep learning later shifted the focus towards neural network-based methods. Martinez et al. [25] emphasized the critical role of 2D pose data in predicting 3D keypoints.

Current methods have adopted two primary paradigms: bottom-up [6, 11] and top-down [7, 12, 39]. While the former starts with individual body joint estimations, the latter begins by detecting persons. Each approach comes with its set of advantages and challenges, with the trade-off between accuracy and computational speed being paramount.

2.2. 2D HPE

Single-person estimation primarily employs regression methods, such as DeepPose [36], and heatmap-based techniques [6, 20]. Multi-person scenarios see the use of both bottom-up methods, like OpenPose [6], and top-down strategies like AlphaPose [11]. Hybrid methods, highlighted by Miaopeng Li et al. [20], merge these techniques.

Another noteworthy top-down model in this category is Mask R-CNN by Kaiming He et al. [12], initially designed for object detection and semantic segmentation, but later incorporated HPE. Based on that framework, Detectron2 [39] was introduced to handle tasks from object detection to 2D keypoint identification. It uses CNNs to generate heatmaps for keypoints, with the heatmap's peak indicating the exact keypoint location for accurate results. Also following [12], Chen et al. [7] developed the Cascaded Pyramid Network (CPN) to improve multi-person pose estimation, focusing on "hard" keypoints that are occluded or not visible.

Our study employed Detectron2 [39] and CPN [7] for 2D keypoint detection due to their precision and state-of-the-art features, with both achieving a high performance on the COCO [22] benchmark. We also integrated DeepSort [38] for tracking individuals in multi-person scenarios, basing 3D pose predictions from specific 2D keypoints.

2.3. 3D HPE

Despite advances in 2D HPE, 3D HPE struggles with depth ambiguities, limited datasets, and complexities associated with occlusions. Considering monocular RGB images and videos and a two-stage approach (2D to 3D Lifting), Martinez et al. [25] set a benchmark in this domain by using a fully connected residual network to regress 3D joint locations from 2D ones. Another influential work by Tome et al. [35] proposed a multi-stage approach where 2D and 3D poses are processed concurrently.

Additionally, temporal data from videos has been incorporated to address depth issues. Pavllo et al. [30] introduced a temporal dilated convolutional model, named VideoPose3D. While this approach is noted for its simplicity and efficiency, it may encounter difficulties in handling continuous occlusions.

Motivated by that, Cheng et al. [8] presented a network that addressing occlusions through temporal frame analysis. This architecture is particularly effective in scenarios with occluded body parts, but only accounts for self-occlusions, since the testing was conducted using data that primarily featured such occlusions, limiting its applicability.

Zheng et al. [41] introduced PoseFormer, a purely transformer-based model for 3D HPE from videos. This model process both spatial and temporal aspects of human movement. Building on this, Zhao et al. [40] developed PoseFormerV2, which employs the frequency domain to boost efficiency and accuracy of 3D HPE. This approach reduces computational demands and increases robustness to noisy in 2D joint detections, making it effective in complex and occluded scenarios.

Shan et al. [34] presented D3DP, an innovative method for probabilistic 3D HPE. D3DP generates multiple potential 3D poses from a single 2D observation, using a denoiser conditioned on 2D keypoints to refine the poses. The hypotheses for the 3D poses are reprojected onto the 2D camera plane, and the best hypothesis for each joint is selected based on reprojection errors. These selections are combined to form the final pose.

2.4. Graph Convolutional Network

Graph-based approaches, such as GCNs [17], can be used to address occlusions in 3D HPE by representing the body as a graph, where each node represents a body keypoint and each edge represents the relationship between two joints. A notable application is the Dynamic Graph Convolutional Network (DGCN) introduced by Zhongwei Qiu et al. [31], that can model relationships between 2D joints over time. Wenbo Hu et al. [13] proposed representing a 3D human skeleton as a directed graph, to capture hierarchical orders among the joints.

Following the DGCN approach, to further enrich the 3D HPE domain, Cai et al. [5] proposed a graph-based approach leveraging spatial-temporal relationships. They formulated 2D pose sequences as graphs and designed a network to capture multi-scale features and temporal constraints. Later, Yu Cheng et al. [9] presented a novel framework for estimating 3D multi-person poses from monocular videos with two directed GCNs, one dedicated to joints and the other to bones, which together estimate the full pose. This framework integrates GCNs and Temporal Convolutional Networks (TCNs) [18] to handle challenges like occlusions and inaccuracies in person detection. They also include directed graph-based joint and bone GCNs.

Our proposal employs a GCN model, which is designed to represent the 3D human pose as an enhanced, undirected graph, inspired by the method in [5]. We have tailored our model to specifically refine 3D pose predictions, particularly effective in scenarios with occlusions. This is achieved by expanding joint relationships, with training conducted on a variety of cases involving occlusions.

2.5. HPE Datasets

The evolution of HPE approaches has underscored the importance of comprehensive datasets, especially in the context of occlusions. For 2D HPE, datasets like MPII [1], COCO [22], and PoseTrack [15] offer diverse scenarios ranging from static images to dynamic videos, capturing real-world complexities. These datasets facilitate the development of models that generalize to multiple environments.

Well-known 3D HPE datasets, such as Human3.6M [14], SURREAL [37], and AMASS [24], typically require sophisticated equipment like motion capture systems for accurate pose recording. While these datasets offer high precision, they often face challenges in diversity and real-world applicability. For multi-person scenarios, datasets like CMU Panoptic [16], 3DPW[2] and AGORA [28] become crucial as they capture more complex interactions and dynamics, including occlusions. Table 1 illustrates the diversity and focus of some of these datasets.

Following the discussion on existing datasets, the introduction of the BEDLAM dataset [3] represents a significant advancement. As a synthetic dataset designed for 3D human pose and shape (HPS) estimation, BEDLAM demonstrated that neural networks trained solely on synthetic data can achieve state-of-the-art accuracy in 3D HPS estimation from real images.

Table 1. 3D HPE Datasets. Data type ‘R’ and ‘S’ denote ‘Real’ and ‘Synthetic’. † non-self occlusions (object-based, multi-person, and out-of-frame). ‡ annotations of keypoint visibility.

	Data	No. of	Action	Single-	Multi-	Occlusions	
Dataset	**Type**	**Frames**	**Tags**	**Person**	**Person**	**Complex†**	**Labels‡**
Human3.6M [14]	R	3.6M	✓	✓	✕	✕	✕
CMU Panoptic [16]	R	1.5M	✕	✓	✓	✕	✕
SURREAL [37]	S	6M	✕	✓	✕	✕	✕
3DPW [2]	R	51K	✕	✓	✓	✓	✕
AGORA [28]	S	17K	✕	✕	✓	✓	✕
BEDLAM [3]	S	380K	✕	✓	✓	✓	✕
BlendMimic3D	**S**	**136K**	✓	✓	✓	✓	✓

While both the COCO [22] and Human3.6M [14] datasets have been instrumental in advancing state-of-the-art algorithms, they present limitations. COCO’s human-curated nature is prone to errors, whereas Human3.6M, although providing high-precision pose data, lacks in representing occluded scenarios. Wandt et al. [33] showed that state-of-the-art 3D HPE models significantly underperform when faced with synthetic occlusions.

Addressing the challenges in 3D HPE occlusion handling highlighted by Wandt et al., we introduce BlendMimic3D. Inspired by Human3.6M and leveraging BEDLAM’s synthetic capabilities, BlendMimic3D offers advanced occlusion management across various levels. It sets a new benchmark in occlusion-aware 3D HPE with action-oriented labeled activities and occlusions, marking keypoints’ visibility per frame as shown in Table 1.

3. BlendMimic3D Dataset

As the need for HPE grows, so does the demand for detailed datasets to train and test models. The efficacy of these datasets is judged by their accuracy, completeness, and variety. Creating 3D HPE datasets is complex and usually requires special tools such as MoCap systems and wearable devices, resulting in datasets created in controlled settings. As argued by Wandt et al. [33], despite the progress achieved with Human3.6M [14] dataset, there remains a notable gap that synthetic datasets can address.

Using Blender [10], a popular open-source 3D computer graphics software, we introduce a synthetic dataset tailored to address challenges such as self, object-based and out-of-frame occlusions. To ensure its adaptability and relevance, we designed it to comprise a diverse set of scenarios, from simple environments resembling Human3.6M [14], to more complex ones with numerous occlusions and multi-person contexts. Figure 2 illustrates examples of frames from videos in our dataset, showcasing the range of settings and multi-person contexts of BlendMimic3D. More detailed

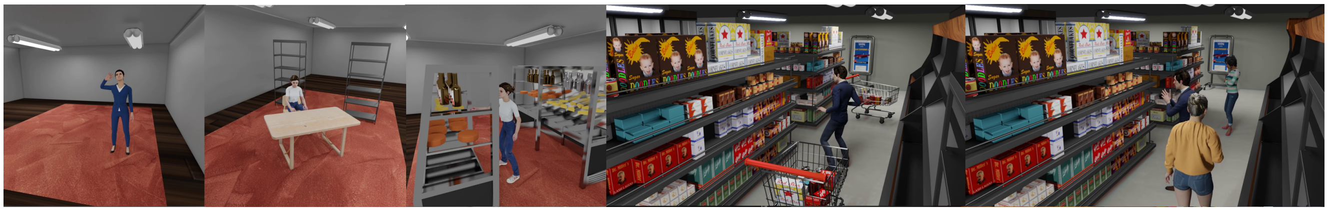

Figure 2. Visual representation of different scenes from BlendMimic3D datasets. From left to right: synthetic subjects, SS1, SS2 and SS3.

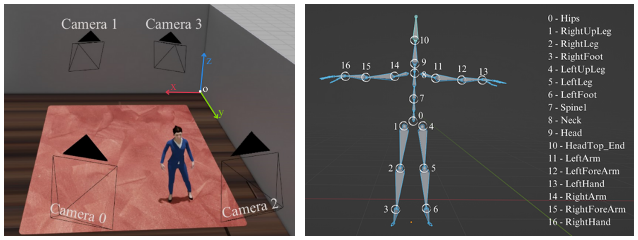

Figure 3. Left: Camera distribution with the world coordinate system at the origin, with subject SS1 of BlendMimic3D dataset. Right: Visualization of 3D character armature, highlighting the specific keypoints used for coordinate extraction.

examples are available in the supplementary material.

In the process of crafting BlendMimic3D, four cameras were positioned within the virtual environment, as depicted in Figure 3 (Left). A skeletal framework, shown in Figure 3 (Right), was attached to a 3D character model, enabling animation of our synthetic subjects. Utilizing resources from Mixamo[3] [4], the characters were animated to simulate a range of actions, such as “Arguing”, “Greeting”, or “Picking Objects”. From each camera’s perspective, videos were generated utilizing Blender’s rendering engine. The resulting dataset comprises:

1. 3 scenarios, from a simple environment to more complex and realistic ones;
2. 3 subjects, each one performing several different actions;
3. Single and multi-person settings, with up to 3 subjects;
4. A total of 128 videos with an average duration of 35 seconds (1050 frames).

Metadata is available for all videos, including the parameters used for camera calibration, 2D and 3D positions of keypoints, as well as a binary array depicting which keypoints were occluded in each frame. All this extracted data is illustrated in Figure 1.

BlendMimic3D is organized in the same manner as the Human3.6M dataset, with videos categorized by subject and action. The dataset includes synthetic subjects designated as SS1, SS2 and SS3. While SS1 focuses on self-occlusions, SS2 addresses object and out-of-frame occlusions. Each of these subjects covers 14 distinct actions. SS3, set in a smart store environment, manages both occlusions and multi-person scenarios. It offers two variations of the same action—one in a single-person context and the other in a multi-person setting. Just like with Human3.6M, each synthetic action is captured in four videos, each with a different perspective.

4. Pose Refinement with GCN

Our proposed methodology is a pose refinement stage, illustrated in Figure 4, where we introduce our Graph Convolutional Network (GCN) as a plugin to enhance the estimated 3D poses. Our GCN is trained on BlendMimic3D dataset, which provides a diverse range of occlusion scenarios. This allows the network to learn and adapt to various occlusion types, refining the pose estimation for occluded joints.

The GCN not only considers spatial relationships between body joints but also temporal continuity across frames. It conceptualizes the human body as a graph structure, where nodes are body keypoints and edges represent joint connections. Unlike traditional models that primarily link a keypoint to its immediate neighbors [13], our model, inspired by the work of Cai et al. [5] and Yu Cheng et al. [9], establishes broader connections across consecutive frames, as depicted in Figure 5. This extended connectivity is crucial for accurately inferring occluded or ambiguous keypoints in challenging environments.

Drawing on the formulation by Kipf and Welling [17], their GCN approach refines spectral graph convolutions to enhance efficiency and scalability. Given a graph $\mathcal{G}$ with

[3] https://www.mixamo.com/#/

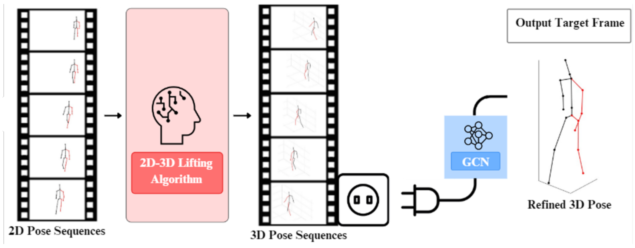

Figure 4. Overview of the proposed framework. After any chosen 3D HPE algorithm, our Graph Convolutional Network (GCN) refines the estimated 3D poses by integrating spatial and temporal insights, leading to enhanced and precise 3D pose estimation, particularly effective in handling occlusions.

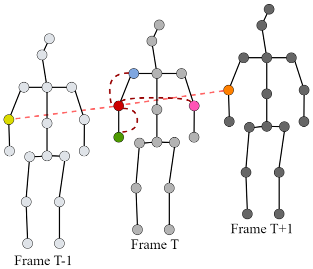

Figure 5. Illustration of the graph dynamics for the right elbow keypoint, with neighboring nodes categorized into six classes: (1) Center (red). (2) Physically-connected node closer to the spine (blue). (3) Physically-connected farther from the spine (green). (4) Symmetric node (pink). (5) Time-forward node (orange). (6) Time-backward (yellow).

an adjacency matrix A, the propagation rule in their GCN model for each layer is expressed as

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), \qquad (1)$$

where $H^{(l)}$ is the activation matrix for the l-th layer. $H^{(0)}$ denotes the input feature matrix, with each row representing a feature vector for every node. $W^{(l)}$, often referred to as the kernel, is the weight matrix for the l-th layer. σ is an activation function, typically the ReLU. The augmented adjacency matrix, $\tilde{A} = A + I$, includes self-connections, and $\tilde{D}$ is its corresponding diagonal node degree matrix.

Kipf and Welling's strategy uses the normalized adjacency matrix to spread node features across the graph. Normalization by the degree matrix $\tilde{D}$ ensures stable gradients and effective training. In (1), the kernel $W^{(l)}$, is shared by all 1-hop neighboring nodes, suggesting a consistent treatment of these immediate neighbors.

To enhance this approach, we expanded from merely considering 1-hop neighbors, recognizing the need for distinct kernels tailored to different neighboring nodes based on their semantics. Following (1), we devised a spatial-temporal undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$. In this graph, $\mathcal{V} \in \mathbb{R}^{T\times J}$ signifies the vertices set corresponding to T consecutive frames (one for each, past, present, and future), with J joints in each frame. $\mathcal{E}$ represents the nodes' connections. The adjacency matrix $A \in \mathbb{R}^{P\times P}$, considering $P = TJ$, that $a_{ij} = 0$ if $(i,j) \notin \mathcal{E}$ and $a_{ij} = 1$ if $(i,j) \in \mathcal{E}$. This adjacency matrix captures both spatial and temporal dynamics across frames.

By classifying neighboring nodes and understanding their semantic relationships (as illustrated in Figure 5), we apply distinct kernels for each class of neighborhood. Drawing from [5], consider an input signal $X \in \mathbb{R}^{P\times C}$ that represents C-dimensional features of P vertices on the graph. The convolved signal matrix $Z \in \mathbb{R}^{P\times C}$, is given by the graph convolution, articulated as

$$Z = \sum_k D_k^{-\frac{1}{2}} A_k D_k^{-\frac{1}{2}} X W_k, \qquad (2)$$

in which k indexes the neighbor class, W_k denotes the filter matrix for the k-th type of 1-hop neighboring nodes. In relation to the normalized $\tilde{A} = A + I_P$ from equation (1), expression (2) decomposes it into k sub-matrices with $\tilde{A} = \sum_k A_k$. Here, $D_k^{ii} = \sum_j A_k^{ij}$ represents the degree matrix that normalizes A_k.

Our model combines graph convolution operation from (2) and 3D convolutions. The primary architecture, depicted in Figure 6, merges both spatial and temporal graph convolutions. The input, a tensor representing 3D keypoints, undergoes normalization for stability. This input tensor has dimensions (N,C,T,V,M), with N as the batch size, C as the number of features, T as the temporal dimension (input sequence length), V as the graph nodes for each frame, and M as the number of instances in a frame.

The core of the model comprises several spatial-temporal graph convolutional (ST-GCN) layers, depicted in Figure 6 , designed for feature extraction and refinement. A non-local block is also incorporated, capturing long-range dependencies and relationships between different input parts. The resulting features are then passed through a fully connected layer producing the final refined 3D pose.

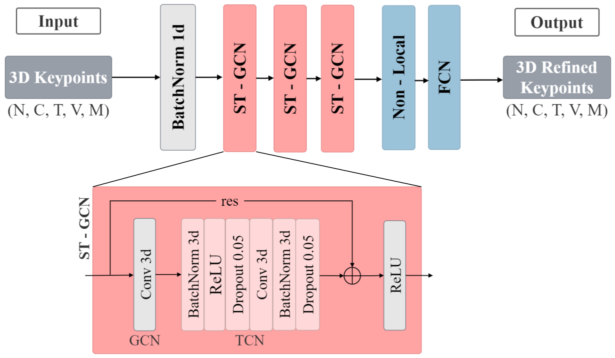

Figure 6. Graph-based 3D human pose refinement architecture with detailed architecture of the Spatial-Temporal Graph Convolutional (ST-GCN) layer.

The details of the ST-GCN layers are shown in Figure 6. Each layer begins with an operation that applies a graph convolution, incorporating the spatial structure and connections defined by an adjacency matrix. Following the graph convolution, the output undergoes a temporal 3D convolution, capturing the temporal relationships across frames. A residual connection is employed to facilitate faster convergence and mitigate the vanishing gradient problem. The final output of each ST-GCN layer passes through a ReLU function for a non-linear transformation. The combination of these operations ensures that our model understands the spatial-temporal dynamics of human actions, enabling accurate 3D pose estimation even in challenging scenarios.

5. Experimental Setup

Our architecture provides an end-to-end solution for 3D pose estimation from video, handling occlusions—a common real-world challenge. Our tests on standard benchmarks show it outperforms existing top methods, especially in occluded scenarios, while also maintaining strong performance in standard situations. The experimental framework was implemented using Pytorch [27], an Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz and two NVIDIA GeForce GTX 1080 Ti.

5.1. Datasets and Evaluation Metrics

Datasets. A primary dataset in our study is Human3.6M [14], which serves as the training foundation for several 3D HPE algorithms [5, 8, 9, 30, 34, 40, 41]. Human3.6M furnishes 3.6 million human poses and corresponding images. Captured in controlled indoor environments, it documents 15 unique actions, with two versions each, from four viewpoints. It is important to note that, due to privacy concerns, data from only 7 subjects is available: S1, S5, S6, S7, S8, S9, S11, totaling 840 videos. For performance evaluation, we sourced all available subjects from Human3.6M [14]. These were combined with our synthetic subjects from BlendMimic3D: SS1, SS2 and SS3.

In line with established practices in 3D HPE research, as seen in previous works [5, 8, 9, 30, 34, 40, 41], we have selected a specific set of subjects for our training and testing phases. For the training of our GCN, we utilize the 3D pose predictions from 6 subjects of Human3.6M, S1, S5, S6, S7, S8 and S11, along with our synthetic subjects SS1, and SS2. We then evaluate the performance of our model on two different subjects, S9 and SS3. Both 3D pose predictions and 3D refined poses are represented in the camera's coordinate system and a single model is used to train all camera views for all actions.

Evaluation metrics. Our 3D human pose estimation evaluation harnesses the Mean Per-Joint Positional Error (MPJPE) [42], a metric that calculates the average ℓ_2-norm difference between estimated and true 3D poses, represented by the equation

$$\text{MPJPE} = \frac{1}{N}\sum_{i=1}^{N} \|J_i - J_i^*\|_2 \,, \tag{3}$$

where N represents the joint count, and J_i and J_i^* denote the true and estimated positions of the i_{th} joint, respectively.

5.2. Implementation Details

2D HPE. This stage focuses on accurately capturing the skeletal structure in two dimensions, which lays the foundation for the subsequent 3D pose estimation. To identify subjects in video frames and extract their 2D keypoints, we use two detection algorithms in our 2D HPE process: CPN [7] and Detectron2 [39]. For keypoints detection using Detectron2 [39], we utilize a pretrained model that uses Mask R-CNN [12] with ResNet-101-FPN [23] as backbone. Regarding CPN [7], which is an extension of FPN as suggested by [30], we employ the 2D keypoint predictions provided from their fine-tuned CPN model for the Human3.6M dataset. For our synthetic 2D pose predictions, we re-implement CPN, using a ResNet-101 backbone with

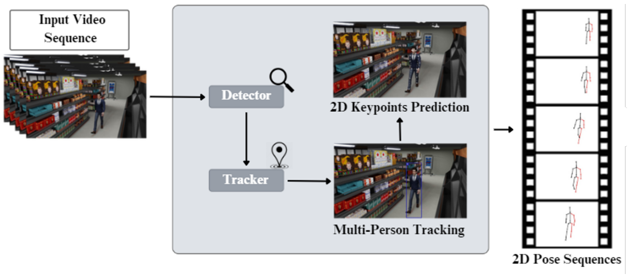

Figure 7. Overview of the proposed preprocessing strategy for 2D HPE. It begins with (1) employing a detection algorithm for pinpointing subjects and capturing their 2D keypoints, followed by (2) a tracking mechanism to maintain focus on a target subject, supplying a sequence of 2D poses.

a 384×288 resolution. This model uses externally provided bounding boxes generated by Detectron2.

To handle dynamic scenes with multiple people, we integrate the DeepSort [38] algorithm, modified to track a specific individual with a unique ID. Our method assumes that the individual remains in the frame throughout the monitoring in a multi-person environment. In order to maintain the tracking continuity, especially when the target ID is temporarily lost, we select the closest bounding box based on centroid distance, prioritizing those with high confidence scores. Also, to improve detection performance, our approach resizes subsequent frames according to the previous tracked bounding box and implementing an region of interest (ROI) cropping strategy focused on the target ID. This entire preprocessing approach, encompassing both detection and tracking phases, is illustrated in Figure 7.

2D-to-3D Pose Conversion We extracted 2D keypoints as inputs for our 2D-to-3D pose lifting module and assessed the performance of various algorithms in handling occlusions with our BlendMimic3D dataset. These algorithms include VideoPose3D [30], PoseFormerV2 [40], and D3DP [34], for which we utilized their available pretrained models. For all three algorithms, we used an input sequence length of 243 frames. Specifically, for PoseFormerV2, we inputted 27 frames into the spatial encoder along with 27 DCT coefficients. For D3DP, we configured the model to use 1 hypothesis and one iteration.

GCN Pose Refinement Our GCN model is trained for 40 epochs with a mini-batch size of 256, using the AMSGrad [32] optimizer and an initial learning rate of 0.001.

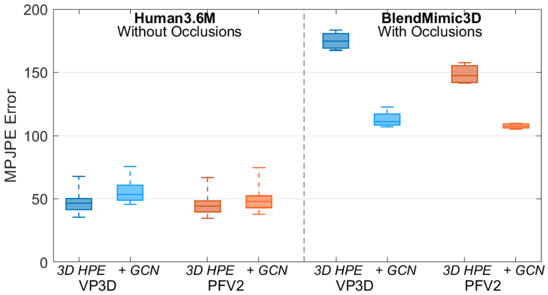

Figure 8. Evaluation of our GCN pose refinement block against previous methods: VideoPose3D (VP3D) and PoseFormerV2 (PFV2), showcasing performance on CPN-based detections across Human3.6M and BlendMimic3D test sets.

The learning rate is reduced by 0.1 every 5 epochs and shrinks by a factor of 0.95 after each epoch, with a more significant reduction of 0.5 every 5 epochs. Training batches are created by a generator based on pre-split subject IDs for training and testing groups, as detailed in Section 5.1. The model is updated through backpropagation based on these batches. We follow the training losses of [5], including 3D pose loss (MPJPE), derivative loss (measuring the Euclidean distance of the first derivative between predicted and ground truth velocities), and symmetry loss (focusing on the accuracy of left and right bone pairs). Test data is solely used for evaluation.

6. Experimental Results

6.1. Quantitative results

Our analysis on Human3.6M and BlendMimic3D datasets, using CPN-based and Detectron2-based 2D detections, demonstrates the positive impact of the GCN pose refinement block in handling occlusions, as evidenced by MPJPE improvements particularly on the occlusion-heavy BlendMimic3D dataset. Figure 8 visually summarizes the enhancements and trade-offs introduced by our GCN across VideoPose3D and PoseFormerV2.

Figure 8 shows that the results with our GCN achieve a comparable error on the non-occluded Human3.6M dataset. However, the baseline approaches exhibit worse MPJPE in occluded scenarios (BlendMimic3D dataset), while our GCN reduces this error escalation. The proposed approach leads to a notable decrease of more than 30% on the average

Table 2. Evaluation of our GCN pose refinement block against previous methods, with CPN and Detectron2, on BlendMimic3D test set. Best results are highlighted in green.

2D HPE	**3D HPE Model –MPJPE (Avg [mm])**					
	VP3D [30]	+ GCN	PFV2 [40]	+ GCN	D3DP [34]	+ GCN
CPN [7]	175.0	112.7	148.6	107.5	100.7	95.3
Detectron2 [39]	198.0	127.7	155.0	106.9	99.9	95.3

errors with occlusions.

Unlike PoseFormerV2 (PFV2) and VideoPose3D (VP3D), D3DP incorporates mechanisms that can handle occlusions, utilizing a diffusion process to add noise and a denoiser conditioned on 2D keypoints, leading to a variety of hypotheses that can capture the possible variations in pose. GCN integration addresses the occlusion management challenges in both VP3D and PFV2 models, as demonstrated in Table 2. This table also highlights the GCN's broader impact, including its application to D3DP, within the BlendMimic3D test set.

Table 2 underscores the GCN's versatility, showing consistent performance enhancements across different 2D detection methods (CPN and Detectron2). It also shows improvements in D3DP's performance, affirming the GCN's value even in models already equipped for occlusion management. Detailed results, categorized by action for each model, can be found in the supplementary material.

This evaluation highlights BlendMimic3D's role in overcoming the self-occlusion bias of datasets like Human3.6M, emphasizing its importance for enhancing 3D HPE robustness through diverse occlusions. It showcases synthetic data's role in model development and affirms our GCN's advancement in occlusion management, setting a new benchmark for occlusion handling in 3D HPE systems.

6.2. Qualitative results

To test our approach in a real-world scenario featuring occlusions, Figure 9 showcases qualitative results. It compares the 3D human pose estimation from VideoPose3D with and without our refined pose, both derived from the same input video. As shown in Figure 9, the GCN approach improves the estimation results, particularly for occluded legs. This suggests that our GCN is effective in handling occlusions – a critical benefit for real-world applications where such occlusions are frequent.

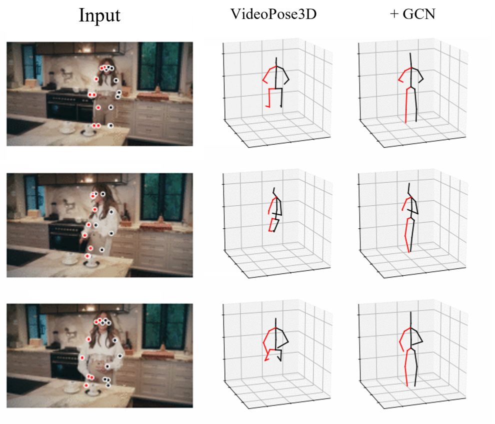

Figure 9. Example showcasing three frames from a real "in the wild" video with the corresponding 3D HPE using VideoPose3D, on Detectron2 detections, with and without the proposed GCN.

7. Conclusion

This work introduces BlendMimic3D, a new benchmark designed to train and evaluate 3D HPE with occlusions. Unlike traditional datasets such as COCO and Human3.6M, with controlled settings and limited occlusion variations, our BlendMimic3D replicates real-world complexities. A standout feature of BlendMimic3D is its expandability and ease of modification[4] , requiring only Blender for animation generation. Additionally, we propose a GCN pose refinement block, that can be plugged in with state-of-the-art 3D HPE algorithms to improve their performance for occluded poses, requiring no further training of the HPE backbone. This ensures that performance improvements in occluded conditions do not compromise accuracy in standard, non-occluded settings. Future efforts will aim to fully preserve performance in these scenarios upon integrating the GCN.

Acknowledgements

This work was supported by research grant 10.54499/2022.07849.CEECIND/CP1713/CT0001 and LARSyS funding (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020, 10.54499/UIDB/50009/2020), through Fundação para a Ciência e a Tecnologia, and by the SmartRetail project [PRR - C645440011-00000062], through IAPMEI - Agência para a Competitividade e Inovação.

[4] https://github.com/FilipaLino/BlendMimic3D-DataExtractor

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 2, 3

[2] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3395–3404, 2019. 2, 3

[3] Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8726–8737, 2023. 2, 3

[4] Sue Blackman. Rigging with mixamo. *Unity for Absolute Beginners*, pages 565–573, 2014. 4

[5] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2272–2281, 2019. 1, 3, 4, 5, 6, 7

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 2

[7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018. 2, 6, 8

[8] Yu Cheng, Bo Yang, Bo Wang, Yan Wending, and Robby Tan. Occlusion-aware networks for 3d human pose estimation in video. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 723–732, 2019. 2, 6

[9] Yu Cheng, Bo Wang, Bo Yang, and Robby T Tan. Graph and temporal convolutional networks for 3d multi-person pose estimation in monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1157–1165, 2021. 3, 4, 6

[10] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 1, 3

[11] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2334–2343, 2017. 2

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 6

[13] Wenbo Hu, Changgong Zhang, Fangneng Zhan, Lei Zhang, and Tien-Tsin Wong. Conditional directed graph convolution for 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 602–611, 2021. 3, 4

[14] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014. 1, 2, 3, 6

[15] Umar Iqbal, Anton Milan, and Juergen Gall. Posetrack: Joint multi-person pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2020, 2017. 3

[16] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015. 2, 3

[17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1, 3, 4

[18] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks for action segmentation and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[19] Hsi-Jian Lee and Zen Chen. Determination of 3d human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 30(2):148–168, 1985. 2

[20] Miaopeng Li, Zimeng Zhou, Jie Li, and Xinguo Liu. Bottom-up pose estimation of multiple person with bounding box constraint. In *2018 24th international conference on pattern recognition (ICPR)*, pages 115–120. IEEE, 2018. 2

[21] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II 12*, pages 332–347. Springer, 2015. 2

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 3

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 6

[24] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 1, 2, 3

[25] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017. 1, 2

[26] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015. 2

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

[28] Priyanka Patel, Chun-Hao P. Huang, Joachim Tesch, David T. Hoffmann, Shashank Tripathi, and Michael J. Black. AGORA: Avatars in geography optimized for regression analysis. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[29] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7307–7316, 2018. 2

[30] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7753–7762, 2019. 1, 2, 6, 7, 8

[31] Zhongwei Qiu, Kai Qiu, Jianlong Fu, and Dongmei Fu. Dgcn: Dynamic graph convolutional network for efficient multi-person pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11924–11931, 2020. 3

[32] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019. 7

[33] István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. How robust is 3d human pose estimation to occlusion? *arXiv preprint arXiv:1808.09316*, 2018. 3

[34] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14761–14771, 2023. 1, 2, 6, 7, 8

[35] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2500–2509, 2017. 1, 2

[36] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 2

[37] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 1, 2, 3

[38] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 2, 7

[39] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2: A pytorch-based modular object detection library. *Meta AI*, 10, 2019. 2, 6, 8

[40] Qitao Zhao, Ce Zheng, Mengyuan Liu, Pichao Wang, and Chen Chen. Poseformerv2: Exploring frequency domain for efficient and robust 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8877–8886, 2023. 1, 2, 6, 7, 8

[41] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11656–11665, 2021. 2, 6

[42] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1):1–37, 2023. 6