Attention-Propagation Network for Egocentric Heatmap to 3D Pose Lifting

Taeho Kang
Seoul National University, South Korea
taeho.kang@hcs.snu.ac.kr

Youngki Lee
Seoul National University, South Korea
youngkilee@snu.ac.kr

Abstract

We present EgoTAP, a heatmap-to-3D pose lifting method for highly accurate stereo egocentric 3D pose estimation. Severe self-occlusion and out-of-view limbs in egocentric camera views make accurate pose estimation a challenging problem. To address the challenge, prior methods employ joint heatmaps-probabilistic 2D representations of the body pose, but heatmap-to-3D pose conversion still remains an inaccurate process. We propose a novel heatmap-to-3D lifting method composed of the Grid ViT Encoder and the Propagation Network. The Grid ViT Encoder summarizes joint heatmaps into effective feature embedding using self-attention. Then, the Propagation Network estimates the 3D pose by utilizing skeletal information to better estimate the position of obscure joints. Our method significantly outperforms the previous state-of-the-art qualitatively and quantitatively demonstrated by a 23.9% reduction of error in an MPJPE metric. Our source code is available on GitHub [1].

1. Introduction

The increasing use of Virtual Reality(VR) and Augmented Reality(AR) applications has prompted efforts to perform various vision tasks with minimal wearable sensors. Specifically, head-mounted cameras in the egocentric setup (Fig. 1) received increasing attention thanks to their accessibility. Here, accurate 3D pose estimation is noted as a task critical for seamlessly integrating virtual selves into the real world. However, existing egocentric pose estimation methods still suffer from accuracy challenges [7].

Conventional 3D pose estimation methods typically derive 3D pose directly from 2D pose information [10, 15, 28]. However, this approach faces challenges in egocentric setups due to inaccuracies in 2D pose estimation resulting from limited camera views and self-occlusion. To address this, egocentric pose estimation methods use joint heatmaps—probabilistic 2D representations of joints [17].

[1] https://github.com/tho-kn/EgoTAP

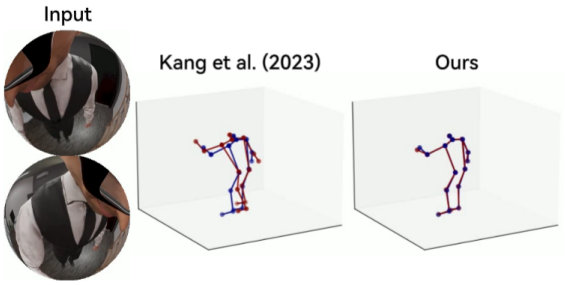


Figure 1. The stereo egocentric input and the comparison of the estimated pose of the state-of-the-art method [7] and ours. Blue color for the ground truth and red color for the respective method's estimation

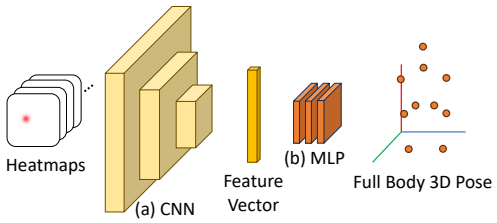


Figure 2. The architecture of the common baseline heatmap-to-3D approach. This architecture is adopted by monocular xR-EgoPose [16] and stereo UnrealEgo [1] for 3D pose inference.

These heatmaps employ probability distributions of likely joint positions rather than exact locations. Following this approach, methods generate heatmaps for key joints from egocentric camera input, consolidate them into a unified feature embedding vector, and perform full-body 3D pose estimation (Fig. 2). However, two critical problems in the heatmap-to-3D lifting process significantly impact position estimation accuracy.

Inefficiency in feature embedding. Obtaining an effective feature embedding from the heatmap poses a significant challenge. A robust embedding vector is crucial for accurately reconstructing the 3D pose, given the indirect mapping between the probabilistic, high-dimensional heatmaps and the 3D pose. However, the standard design, utilizing a CNN (Convolutional Neural Network) encoder, proves inadequate for feature summarization. The CNN encoder

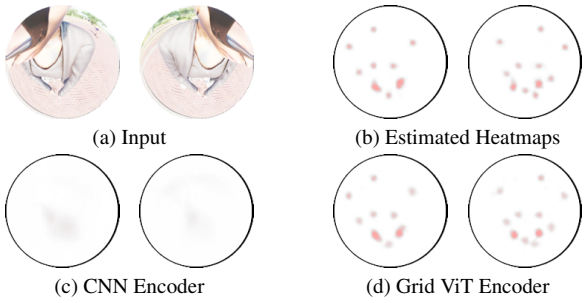


Figure 3. Comparison of the reconstructed heatmaps from the encoded heatmap features, with the frozen encoder from (c) CNN Encoder and (d) Grid ViT Encoder of the pose estimation model.

fails to preserve correspondence between specific heatmaps and joint poses, as features are merged into a single shared embedding. Furthermore, the spatial locality assumption of CNNs does not hold in an egocentric setup, where related joints may be distant in pixel space due to the proximity of ego-centric cameras to body parts and biased positions. The 3D pose lifting employs heatmap reconstruction loss [1, 7, 16, 26] to recover heatmap information, but full recovery becomes challenging once the embedding vector has significantly lost information, as illustrated in Fig. 3.

Feature Importance-agnostic 3D Lifting. Secondly, there is a significant inaccuracy in estimating a full-body 3D pose without effectively distinguishing between important and unimportant features, as seen in the conventional pipeline using Multi-Layer Perception (Fig. 2 (b)). The prior methods [1, 7, 16, 26] do not consider the certainty of joints or the physical relationships between them, relying solely on the motion distribution within the training data. This approach may result in obscure joint features adversely affecting joints with clear visual cues in the camera or those estimable from nearby joint information. The supplementary material highlights that body extremities with less visibility exhibit higher estimation errors.

To tackle these challenges, we introduce EgoTAP (Egocentric Transformer-Attention Propagation Network). EgoTAP incorporates two key techniques: Grid ViT (Vision Transformer) Heatmap Encoder and Propagation Network. We design the former to generate an effective feature embedding that (i) preserves the correspondence between heatmaps and feature embedding and (ii) captures meaningful relationships between distant pixels. The latter assigns weights to evident joint features with clearer visual cues and predicts the position of less visible joints using the skeletal information of body limbs. Through these techniques, we achieve a substantial improvement in pose error metrics, demonstrating a 23.9% reduction in MPJPE and a 17.7% decrease in PA-MPJPE compared to state-of-the-art methods.

Grid ViT Heatmap Encoder addresses the inefficiency of the CNN encoding process. The Grid ViT Heatmap Encoder consolidates all joint heatmaps into a single image and divides them into patches, with each patch corresponding to a heatmap. Subsequently, self-attention is applied across all patches, generating per-patch feature embeddings. The ViT Heatmap Encoder offers two key advantages. Firstly, the per-patch embedding better preserves the position information of the original joint heatmaps. Secondly, self-attention facilitates the effective embedding of inter-joint relationships, particularly useful for joint features in distant areas.

Propagation Network propagates various features from the neck joint, likely to have the evident features, to the body's extremities with less visibility, following the body hierarchy. To enable propagation, we devise an LSTM [6]-inspired cell, PU (Propagation Unit). The PU takes the parent joint's feature, the relational (limb) features as a hidden state, and the child joint's features as input to predict the final 3D position. The PU has an additional gate to forget the parent and relational features in case the child joint features are evident, limiting the role of the predictive estimation only for obscure joints. This design explicitly leverages the physical relationships of joints rather than implicitly inferring them from the training data, thereby contributing to higher pose estimation accuracy.

In summary, our contributions are the following:

- The first egocentric 3D pose estimation method using a vision transformer for efficient feature embedding.
- The Propagation Network that enables the predictive estimation for obscure joints using skeletal hierarchy.
- The Propagation Unit, to control the importance of the propagated features.
- EgoTAP outperforms the state-of-the-art stereo egocentric pose estimation both qualitatively and quantitatively.

2. Related Works

2.1. Egocentric Pose Estimation

Egocentric pose estimation can be classified into two main categories. The first category focuses on estimating the pose of other people within the camera's field of view, as in Ng et al.[12] while the second category estimates the pose of the user self [9]. Our work belongs to the second category, especially with a downward-oriented egocentric camera.

EgoCap [13] showcased its potential using stereo cameras on a helmet-mounted stick. Mo^{2}Cap^2 [21] and xR-EgoPose [16] have introduced single-camera methods, which handle occlusion. The former proposes a two-branched heatmap, one for the lower body with a magnified view. The latter adds a heatmap reconstructor to preserve the probabilistic information of heatmaps. Recent methods utilize an external camera view to make a weakly labeled large-scale dataset [19] and a scene depth estimation model

to estimate 3D pose with volumetric heatmaps [20]. These methods, however, require additional external cameras or depth datasets from specific views.

Recently, a stereo egocentric setup has gained attention for a wide-view stereo perspective. EgoGlass [26] introduces an unobtrusive eyeglass-mounted stereo camera setup, minimizing obtrusiveness. It incorporates an additional segmentation branch on the heatmap estimator module to improve the awareness of body parts and pixel correspondence. UnrealEgo [1] introduces a publicly available synthetic large-scale dataset based on the EgoGlass setup and proposes to share weights and merge features across the stereo view in the heatmap estimator. Ego3DPose [7] suggests making an independent estimate of the 3D orientation of each limb, using the concatenated orientation vector for the final decoder. We observed two problems in these prior works, i.e., information loss in feature embedding and data-dependant estimation of obscure joints, and propose two corresponding techniques to address the problems.

2.2. 3D Human Pose Estimation with Transformer

The transformer-based architecture has been explored for the 3D pose estimation task. Epipolar Transformers [5] utilizes attention to match features along the epipolar line from the stereo view. Most methods focused on using transformers for 2D to 3D pose lifting spatially and temporally. PoseFormer [28] is the first transformer-based 2D-to-3D pose lifting method consisting of spatial and temporal transformer networks. MixSTE [25] and PoseFormerV2 [27] improved it with the per joint temporal characteristics and frequency domain feature. Unlike prior works, we exploit the transformer to effectively embed heatmap information for accurate heatmap-to-3D pose lifting.

2.3. Skeletal Network Models

Multiple works utilize skeletal hierarchy for vision tasks. For instance, Liu et al. [11] uses spatio-temporal LSTM to iterate through all joints for action recognition. Most recent efforts utilize a graph-based model to represent skeletal hierarchy. The Graph Convolutional Networks [8] is widely utilized for activity recognition [3] while ST-GCN [22] models a dynamic skeletal graph in a spatiotemporal manner. The graph-based models are adapted for the pose estimation [22–24], using dynamic skeletal graphs with action-specific edges or adopting adaptive ST-GCN [22, 23].

Our work is the first to leverage skeletal information in the ego-centric setup. Specifically, we address the challenge of obscure features, particularly for body extremities, which impact the pose estimation of all body parts. Introducing a skeleton-aware uni-directional Propagation Network model, we leverage clear visual cues from camera-proximate joints to estimate the pose of body parts with obscure visual features.

3. Method

3.1. Overview

Overall Architecture. Fig. 4 illustrates the comprehensive architecture of EgoTAP. It comprises two essential components: the Grid ViT Heatmap Encoder and the Propagation Network. The Grid ViT Heatmap Encoder takes joint heatmaps as input and generates effective feature embeddings for each joint. The Propagation Network processes these embeddings with awareness of the skeletal structure to estimate the 3D pose accurately. Notably, the per-joint feature embedding is propagated through a skeletal hierarchy, represented as a tree structure with a root representing the head. In Fig. 4, a simplified skeleton is depicted, showcasing the propagation from the head to the hand, highlighted in red. The feature propagation utilizes the PU (Propagation Unit in Fig. 5), which calculates joint states based on the parent joint's states and other self-joint features. The hidden states of the last PU layer are concatenated with the joint features from the Grid ViT encoder and linearly projected to estimate the 3D pose of each joint.

Input and Output. Our method utilizes a pre-trained and frozen heatmap estimator that takes stereo RGB images $I \in \mathbb{R}^{2\times256\times256\times3}$ and estimates stereo heatmaps for N_J joints $\mathbf{H_J} \in \mathbb{R}^{2N_J\times64\times64}$ and N_L limbs $\mathbf{H_L} \in \mathbb{R}^{2N_L\times2\times64\times64}$. EgoTAP takes the heatmaps and reconstructs the 3D pose $P \in \mathbb{R}^{N'_J\times3}$ of N'_J joints relative to the user's root defined in the dataset. Note that the number of estimation targets N'_J can differ from the number of joints with heatmap N_J depending on the dataset.

Loss. We use the Euclidean distance and the cosine similarity-based loss between the ground-truth pose and the estimated pose to train the Attention-Propagation network. The loss formulation is in the supplementary material.

Heatmaps. Two types of heatmaps for joints and limbs are used. We follow the standard definition of joint heatmap [17] where pixel values represent the probability that the joint is in that 2D coordinate. The limb heatmaps have two channels and are used to get relational features between two joints for the Propagation Network in Sec. 3.3. We use a limb heatmap suggested by Kang et al. [7], representing 3D information along with limb visibility as a line connecting joints. From the next section, we denote two types of heatmaps: *joint heatmaps* and *limb heatmaps*. We use a pre-trained ResNet-18 [4] based U-Net [14] architecture with a shared weight for two input image encoders and shared decoder, suggested by Akada et al. [1] for heatmap estimation.

3.2. Grid ViT Heatmap Encoder

Our encoder, described in Fig. 4, combines all joint heatmaps into a large single grid image. The grid is split into patches, linearly projected to make the input embed-

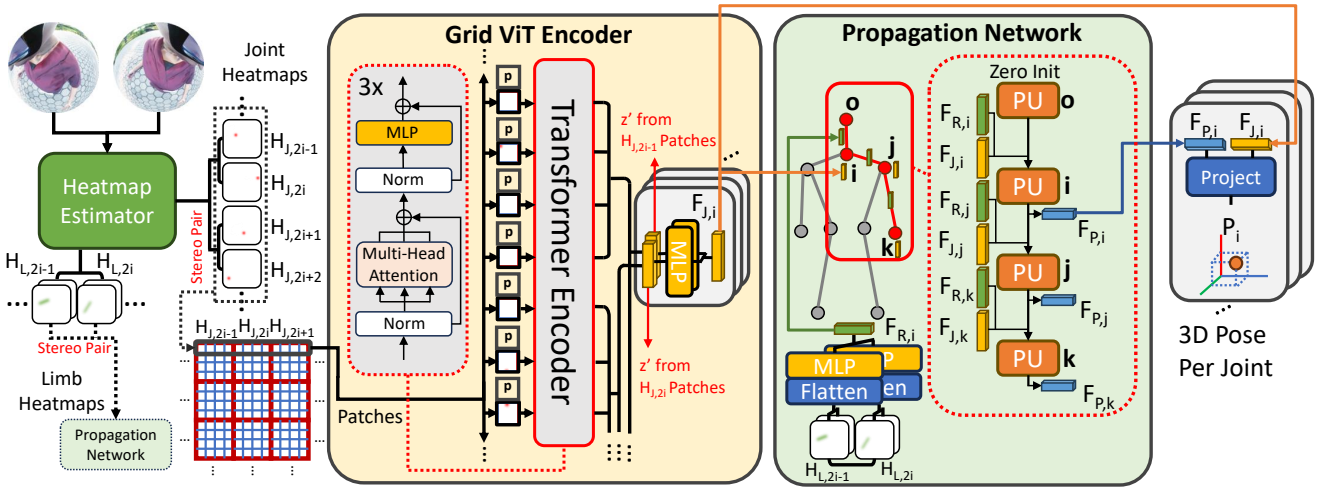


Figure 4. Overall network architecture of EgoTAP. EgoTAP takes heatmaps from pre-trained heatmap estimators taking stereo input images and lifts the heatmaps to the 3D pose with the Grid ViT Encoder, Propagation Network, and finally, a projection layer.

ding, and fed to a transformer [18] encoder architecture with multi-head attention. The transformer encoding process preserves the correspondence between a patch and the input feature embedding in the output. The output feature embeddings corresponding to individual input patches are concatenated and re-encoded to form a feature embedding vector for the heatmap.

Unlike the CNN encoder, where the communication occurs within the nearby pixels of different heatmaps, the Grid ViT Heatmap Encoder allows communication between heatmap patches that are far spatially. This allows features to be shared without downsampling, minimizing the loss of information. The efficiency of the encoder is demonstrated by the precisely reconstructed heatmaps from the embeddings in Fig. 3 and Table 3, and improved pose estimation accuracy.

To formulate the process, let $\{\mathbf{H_{J,i}} \in \mathbb{R}^{64\times64} | i = 1, 2, \ldots, 2N_J\}$ be sets of $2 \times N_J$ stereo joint heatmaps. Heatmaps are arranged into a single grid image. The image is subsequently split to total $4 \times 4 \times 2N_J$ patches $\{X_i \in \mathbb{R}^{16\times16} | i = 1, 2, \ldots, 32N_J\}$ where 16 patches corresponds to a heatmap. $X_{16(i-1)+1}$ to X_{16i} corresponds to i-th heatmap for simplicity.

Each patch X_i is then projected to an input embedding space $\mathbb{R}^{1024}$ with a learnable projection matrix $W_z \in \mathbb{R}^{1024\times256}$. Additionally, learnable positional encodings $\mathbf{p}_i \in \mathbb{R}^{1024}$ are added, resulting in the transformer input embedding z_i. The projected embedding with positional encoding for each patch is:

$$z_i = W_z \cdot Flatten(X_i) + \mathbf{p}_i \tag{1}$$

$z = [z_1, z_2, \ldots, z_{32N_J}]$ is encoded by three ViT transformer encoder [2] layers with multi-head attention to output $z' = [z'_1, z'_2, \ldots, z'_{32N_J}]$. For the j-th heatmap, the corresponding output embeddings from 16 patches are concatenated to $\mathcal{Z}_j$ and then re-encoded to smaller dimensional feature embedding k_j through multiple fully connected layers denoted as E_K. The process is formulated as follows:

$$z' = TransformerEncoder(z) \tag{2}$$

$$\mathcal{Z}_j = [z'_{16(j-1)+1}, z'_{16(j-1)+2}, \ldots, z'_{16j}] \tag{3}$$

$$k_j = E_K(\mathcal{Z}_j) \tag{4}$$

A joint feature $\mathbf{F_{J,i}} \in R^{256}$ that corresponds to a specific joint is obtained by concatenating the stereo heatmap features. Let's say $2i-1$ and $2i$-th heatmap correspond to i-th joint.

$$\mathbf{F_{J,i}} = [k_{2i-1}, k_{2i}], \text{ for } 1 \leq i \leq N_J \tag{5}$$

3.3. Propagation Network

Propagation Process. The Propagation Network estimates the joint positions using their parent joints' positions and the relationships between the joints. The Propagation Network is inspired by the stereo setup's capability to estimate 3D pose without the help of other joints and the general trend of higher visibility on joints closer to the camera in the egocentric setup. Sec. 4.3.2 shows that the Propagation Network effectively takes advantage of accurate estimation of the parent joint with a Propagation Potential and Propagation Effect metric.

The Propagation Network comprises a relational feature encoder and the 2-layered PU that handles the propagation process. The relational feature encoder takes the estimated limb heatmaps to output the relational feature between joints. The PU handles the propagation process, which takes the parent states, relational and joint features

of the child joint as input and generates the child joint's states. The states of joints are propagated through the tree hierarchy from the head directly attached to the camera to the extremities. During propagation, the reflection of the parent joint information is flexibly determined based on the certainty of the parent and child joint features by the PU.

We leverage the limb heatmaps with 3D information embedded with a trigonometric function of camera view angle [7] to provide information about the connection between the parent and child joint. An encoder with fully connected layers E_R encodes limb heatmaps $\mathbf{H_{L,i}} \in R^{2\times64\times64}$ into a limb feature. Stereo limb features are concatenated to form relational feature $\mathbf{F_R}$. Let's say $\mathbf{H_{L,2i-1}}$ and $\mathbf{H_{L,2i}}$ corresponds to a limb that connects the i-th joint and its parent. The process is:

$$\mathbf{F_{R,i}} = [E_L(\mathbf{H_{L,2i-1}}), E_L(\mathbf{H_{L,2i}})], \text{ for } 1 \leq i \leq N_L \quad (6)$$

The Propagation Network consists of two layers of the Propagation Unit, described later. For a tree hierarchy where $parent(i)$ denotes a parent joint's index, and $PNet((H, C), R, J)$ denotes the Propagation Network, which takes hidden and cell states for two PU layers $H = [h_1, h_2]$, $C = [c_1, c_2]$, relational feature R and joint feature J, the hidden and cell state for i-th joint $\mathbf{H}_i, \mathbf{C}_i$ is computed as follows:

$$\mathbf{S}_i = (\mathbf{H}_i, \mathbf{C}_i) \quad (7)$$

$$\mathbf{H}_0 = \vec{0}, \mathbf{C}_0 = \vec{0} \quad (8)$$

$$\mathbf{S}_i = PNet(\mathbf{S}_{\text{parent}(i)}, \mathbf{F_{J,i}}, \mathbf{F_{R,i}}), \text{ for } 1 \leq i \leq N_J \quad (9)$$

The root joint head is indexed 0 and initialized with zero vector, as it is not visible from an egocentric view and, thus, does not have features. The i-th Propagated Feature $\mathbf{F_{P,i}} \in R^{256}$ is a hidden state from the second layer of the Propagation Network $\mathbf{h}_{2,i}$.

The output of the Propagation Network $\mathbf{F_{P,i}}$ and transformer output joint features $\mathbf{F_{J,i}}$ for each joint are concatenated and projected to estimate the 3D position of each joint.

Propagation Unit. We devise a Propagation Unit inspired by the LSTM cell for the above propagation process. Fig. 5 shows the internal structure of the Propagation Unit. The Propagation Unit weights the parent's hidden state and the relational feature with the joint feature. The joint heatmap from stereo views can be sufficient for precise 3D estimation, and this weighting limits the role of the predictive estimation for obscure joints.

To formulate the Propagation Unit, we denote the weight matrix as W and bias vectors as b. The symbol $\odot$ represents element-wise multiplication. The $+$ sign represents element-wise addition. σ denotes the sigmoid activation.

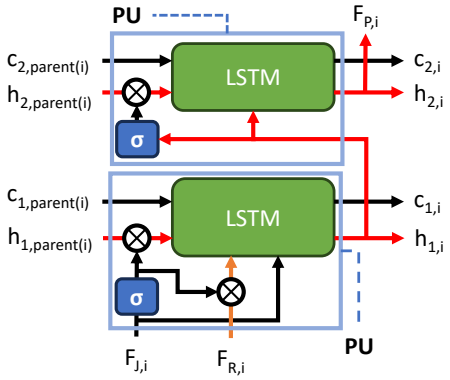


Figure 5. The Propagation Network with two layers of Propagation Unit.

$$f'_i = \sigma(W_{f'} \cdot \mathbf{F_{J,i}} + b_{f'}) \quad (10)$$

$$f''_i = \sigma(W_{f''} \cdot \mathbf{F_{J,i}} + b_{f''}) \quad (11)$$

$$h'_i = f'_i \odot h_{parent(i)} \quad (12)$$

$$r'_i = f''_i \odot \mathbf{F_{R,i}} \quad (13)$$

An additional forget gate is computed from the joint feature and is denoted as f'_i and f''_i. The additional forget gate controls both the parent joint's hidden state and the relational feature between two joints, resulting in the modified hidden state h'_i and the modified relational feature r'_i. Subsequently, these modified states and the joint feature treated as input are used in the standard LSTM architecture, weighted, and then applied non-linearity for the four gates: input, candidate cell state, forget, and output.

For the second layer of the Propagation Network, as there is only a hidden state from the previous layer without relational or joint feature distinction, the hidden state from the previous layer is used to forget the parent joint's hidden state in the current layer.

4. Evaluation

4.1. Experiment Setup

4.1.1 Datasets

Overview. We used two datasets: UnrealEgo [1] and EgoCap [13] for the 3D pose estimation in the stereo egocentric camera setup. We conducted the within-dataset evaluation using each dataset's train and test set split since the egocentric datasets have significantly different setups and resulting views.

UnrealEgo. The UnrealEgo [1] is a synthetic dataset containing 450k frames with 17 characters. The dataset covers a variety of environments and motions that are challenging to capture in a real-world setup. There are a total of 16 joints to estimate. The target local 3D pose is in a pelvis-relative coordinate system, unlike the camera coordinate system in

most datasets, with a head pose to estimate. The pelvis and head do not have corresponding heatmaps and features. We added a learnable matrix for linear projection taking $\mathbf{F_J}$ and $\mathbf{F_P}$ to estimate offset for all joints and head pose. We found that this simple change effectively deals with different pose definitions.

EgoCap. The EgoCap [13] dataset is captured with egocentric cameras attached at the end of the stick on the helmet. It comprises 35k frames for training from six subjects and 1k for testing from one subject with 3D pose annotation. Evaluation with this dataset showcases applicability in a real-world textured image. There are a total of 17 joints to estimate.

4.1.2 Baselines

We experiment with three baseline stereo egocentric pose estimation methods: EgoGlass [26], UnrealEgo [1], and Ego3DPose [7]. We use official implementations with larger embedding and pose decoder dimensions, which gives better accuracy than the original code. EgoGlass [26] implementation is taken from the Ego3DPose [7] as no official source code is provided.

4.1.3 Metrics

The MPJPE and PA-MPJPE metrics are used. The MPJPE is a mean per joint position error in a 3D Euclidian distance. PA-MPJPE applies Procrustes analysis before computing the MPJPE to calculate transform-invariant positional error.

4.2. Overall Performance

4.2.1 Qualatative Results

Fig. 6 presents a qualitative comparison between our method and previous approaches on the UnrealEgo and EgoCap datasets. A more detailed qualitative comparison is available in the supplementary video. Our method demonstrates a significant improvement over baseline methods.

4.2.2 Evaluation on UnrealEgo

The second column of Table 1 presents the quantitative evaluation results on UnrealEgo [1] using MPJPE and PA-MPJPE metrics. Our method demonstrates superior performance compared to state-of-the-art methods, achieving a 23.9% reduction in MPJPE and a 17.7% decrease in PA-MPJPE. These improvements extend across all 31 activity categories detailed in the supplementary material, covering a range of movements from common actions like sitting and standing to less frequent crawling and crouching and more complex motion categories, including sports.

Noteworthy improvements are observed across various categories, with the most substantial enhancement in the

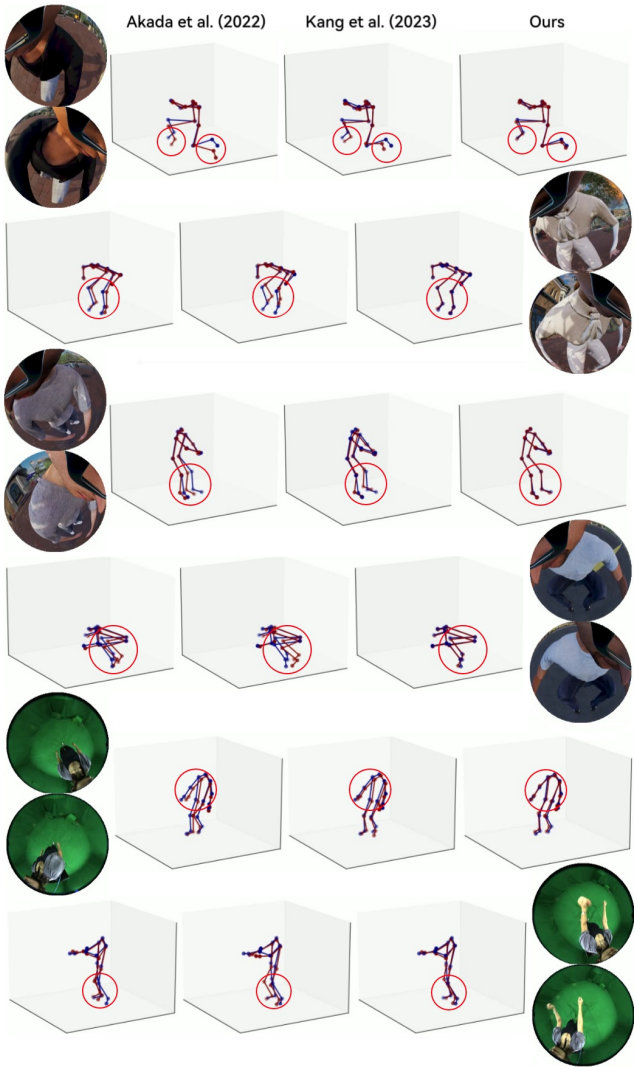
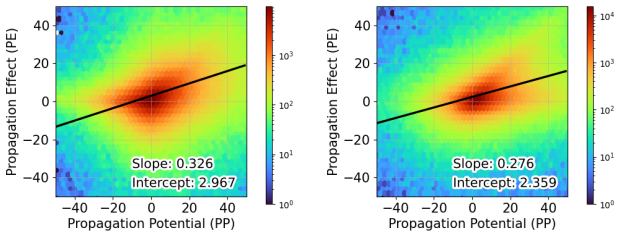


Figure 6. Qualitative comparison of EgoTAP with state-of-the-art stereo egocentric pose estimation methods. The blue is the ground truth, and the red is the estimated pose.

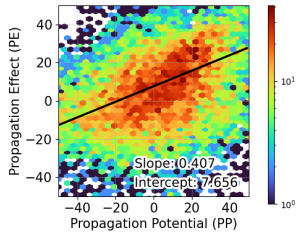
"Crouching-Forward" category, boasting a 31.3% reduction in MPJPE. Conversely, the smallest improvement is noted in the "Crawling" activity, with an 8.8% decrease in MPJPE. It's important to acknowledge that while our method relies on visual cues, the effectiveness varies based on the visibility of body parts. For instance, in activities like "Crouching-Forward," where many body parts are partially visible, our method excels in improving accuracy. On the other hand, in activities like "Crawling," where visible body features are significantly lacking, the challenge of enhancement is more pronounced.

4.2.3 Evaluation on EgoCap

The third column of Table 1 presents the quantitative results on the EgoCap dataset. Our method demonstrates signif-



(a) UnrealEgo [1] (b) UnrealEgo (Camera-relative)



(c) EgoCap [13]

Figure 7. Hexagonal-grid density plot of the Propagation Potential and the Propagation Effect(mm) in our evaluation datasets. The dark line shows linear regression results.

Method	UnrealEgo [1]	EgoCap [13]
EgoGlass [26]	81.55 (61.56)	67.90 (-)
UnrealEgo [1]	63.53 (47.76)	70.77 (52.91)
Ego3DPose [7]	53.99 (43.02)	69.45 (49.98)
Ours	**41.06** (**35.39**)	**55.38** (**45.24**)

Table 1. Evaluation results of state-of-the-art methods and ours on two datasets. The metric is MPJPE, and in the bracket is PA-MPJPE. The bold text indicates the best results.

icant outperformance, surpassing EgoGlass [26] by 22.6% in MPJPE and Ego3DPose [7] by 9.4% in PA-MPJPE. For EgoGlass [26], we report the MPJPE value from their paper, as they do not furnish official code or network details, and the available replication [7] did not match the performance.

The smaller improvement in PA-MPJPE, which discards the effect of the root's transform, could be attributed to prior methods estimating the full body pose as a whole. They might capture the relative pose between joints while the estimation is globally biased. Nevertheless, when integrating the output camera coordinate system pose with the 6-DoF pose of VR and AR devices, precise pose estimation in the correct coordinate frame is crucial for accurate body tracking in the global coordinate system.

We observed that the estimated limb heatmaps in the EgoCap dataset exhibit lower accuracy than those in the UnrealEgo dataset, as illustrated in the supplementary material. This discrepancy could be attributed to the limited volume and the small number of subjects in the EgoCap dataset. Despite these challenges, our Attention-Propagation network effectively lifts the 3D pose from

Method	UnrealEgo [1]	EgoCap [13]
Heatmap Encoder		
CNN	63.53 (47.76)	70.77 (52.91)
Channel ViT	61.62 (47.05)	83.39 (56.29)
Grid ViT	49.03 (41.03)	63.97 (53.17)
Propagation Network		
Grid ViT + RF	48.12 (40.79)	63.09 (52.60)
Grid ViT + LSTM	49.43 (41.31)	60.16 (49.18)
Grid ViT + LSTM RF Alter	44.97 (38.99)	62.60 (50.78)
Grid ViT + LSTM RF Concat	44.77 (38.91)	58.35 (47.06)
Ours (Grid ViT + PU)	**41.06** (**35.39**)	**55.38** (**45.24**)

Table 2. Ablation results of our method for two main components on two datasets. The metric is MPJPE, and in the bracket is PA-MPJPE. The bold text for metrics indicates the best results.

Heatmap Reconstruction Error	10^{-4}/Pixel
Zeros	5.45
CNN Encoder	4.84
Grid ViT Heatmap Encoder	1.68

Table 3. Reconstruction mean square error of the heatmaps from the features encoded with a different frozen encoder architecture, experimented in the UnrealEgo [1] dataset.

heatmaps. However, Ego3DPose [7], which utilizes limb heatmaps, did not perform well. This could be attributed to their explicit inference of orientation for each limb. The final decoder, which takes independent information as an output orientation, struggles with inaccurate information.

4.3. Ablation Study

We performed ablation studies to showcase the effectiveness of each network component, as summarized in Table 2.

4.3.1 Grid ViT Heatmap Encoder

Pose Estimation: We assess the impact of the Grid ViT Heatmap Encoder. "CNN" presents the results from UnrealEgo [1], utilizing a CNN. "Channel ViT" showcases the outcomes with a typical encoder with ViT, where heatmaps are concatenated along the channel axis before being split into patches, resulting in feature embeddings that do not align with the heatmaps. Simply adopting transformers [18] yields minimal improvement, i.e., a 3% reduction in MPJPE, compared to the CNN-based lifting for the UnrealEgo [1] baseline and dataset. However, this approach significantly degrades performance on EgoCap [13]. This observation underscores the importance of addressing the correspondence between feature embedding and heatmaps in the pose estimation process.

Heatmap Reconstruction: We conducted experiments to evaluate the heatmap encoder's efficiency in encoding

heatmap features. To achieve this, a simple decoder is appended to our encoder and baseline encoders. The decoder is trained to reconstruct the estimated heatmaps from the feature embedding. Table 3 presents the reconstruction error of the heatmap in the test set. The "Zeros" row provides the error for a zero-only output for comparison. The results demonstrate that the Grid ViT Heatmap Encoder effectively extracts heatmap features, evidenced by the reconstructed fine details of the heatmap in Fig. 3. In contrast, the heatmaps were not recoverable from features encoded by CNN, highlighting its inefficiency.

4.3.2 Propagation Network

Pose Estimation: We investigate if including relational features alone can significantly enhance accuracy through "+ RF" when incorporated with our Grid ViT encoder. The relational features are concatenated to the joint features for the final projection layer without the involvement of a propagation network. This approach demonstrates marginal impact or even degrades the estimation accuracy. Additionally, we analyze the effect of the Propagation Network with LSTM [6]. In the case of "+ LSTM," only joint features are utilized in the propagation, yielding a marginal effect.

Additional experiments investigate the impact of the Propagation Network without PU, denoted as "+ LSTM RF Alter" and "+ LSTM RF Concat." Relational and joint features are alternately taken in the former, and the propagation feature is output in the joint feature step. The latter takes both as a concatenated vector. Both methods demonstrate improvements, with the latter achieving an 8.7% and 8.8% reduction in MPJPE for two datasets compared to the Grid ViT Heatmap Encoder-only approach. The final model, incorporating PU, maximizes the potential of the Propagation Network, showcasing a 16.3% and 13.4% improvement in MPJPE for the two datasets. This highlights the significance of balancing the role of predictive estimation using parent joints and direct estimation using self-joint features.

Propagation Potential and Effect: The Propagation Network leverages more evident parent joint features to improve the child joint's pose estimation. The hexagonal-grid density plot in Fig. 7 illustrates its impact quantitatively. The x-axis represents the Propagation Potential ($\mathbf{PP}$). $\mathbf{PP}$ approximates the upper bound of the improvement using the parent's feature, with a difference between the parent and child joint's pose estimation error. On the y-axis, the Propagation Effect ($\mathbf{PE}$) is the improvement of the child joint's pose error by the Propagation Network. Using Δ to denote the pose estimation error, subscripts to denote joints, and superscripts to denote the model ($\mathbf{NP}$ without propagation, $\mathbf{P}$ with propagation), we define these metrics as follows.

$$\mathbf{PP} = \Delta^{\mathbf{NP}}_{\text{child}} - \Delta^{\mathbf{NP}}_{\text{parent}}$$
$$\mathbf{PE} = \Delta^{\mathbf{NP}}_{\text{child}} - \Delta^{\mathbf{P}}_{\text{child}}$$

For all datasets, linear regression reveals a positive relationship between $\mathbf{PP}$ and $\mathbf{PE}$ with a p-value of the null hypothesis $< 10^{-3}$, indicating that the Propagation Network is more effective when the parent joint has a more precise estimation, aligning with expectations. The average $\mathbf{PP}$ and $\mathbf{PE}$ were 16.97 and 8.50 for the UnrealEgo dataset [1] and 4.32 and 9.39 for the EgoCap [13] dataset. The UnrealEgo [1] dataset exhibits higher potential due to the cameras closer to the head, unlike cameras around 20cm away from the head in the EgoCap dataset [13].

The effect is more pronounced for the UnrealEgo [1] dataset when the 3D pose is estimated in camera-relative coordinates. This eliminates the global offset (pelvis pose) bias from per-joint improvement. Fig. 7 (b), exhibits trends where $\mathbf{PE}$ is similar to $\mathbf{PP}$ or close to zero. When the $\mathbf{PE}$ is similar to $\mathbf{PP}$, the child joint's pose error is improved close to the parent joint's error. The effect of the Propagation Network is near the upper bound ($\mathbf{PP}$). The propagation cannot improve the child joint's pose error in some cases, possibly due to the occlusion of limbs. Such cases exhibit near zero $\mathbf{PE}$. 66.07% of $\mathbf{PE}$ and 75.62% of $\mathbf{PP}$ in the samples are positive, and 54.16% of samples lie in the first quadrant. The average positive $\mathbf{PE}$ is 10.75, while the average negative $\mathbf{PE}$ is only -0.51, demonstrating that many joints significantly benefit from the propagation.

5. Conclusion

In this study, we introduce a novel heatmap-to-3D lifting method tailored for the stereo egocentric setup, employing a transformer for efficient feature embedding and an attention-driven Propagation Network focused on evident features. We demonstrate effective heatmap feature extraction through the Grid ViT Heatmap Encoder, employing patch-wise communication with self-attention to preserve correspondence between the heatmap and the feature embedding. The Propagation Network utilizes visual cues from the proximate parent joint, leveraging joint relational information to predictively estimate less visible child joint poses. Our experiments highlight significant advancements over state-of-the-art stereo egocentric pose estimation methods, underscoring the efficacy of our proposed approach.

Acknowledgments This work was supported by the NRF Korea grant [No. 2022R1A2C3008495; No.RS-2023-00218601] and the IITP grant [NO.2021-0-01343-004, Artificial Intelligence Graduate School Program (Seoul National University)] funded by the Korea government(MSIT).

References

[1] Hiroyasu Akada, Jian Wang, Soshi Shimada, Masaki Takahashi, Christian Theobalt, and Vladislav Golyanik. Unrealego: A new dataset for robust egocentric 3d human motion capture. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3, 5, 6, 7, 8

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 4

[3] Miao Feng and Jean Meunier. Skeleton graph-neural-network-based human action recognition: A survey. *Sensors*, 22(6):2091, 2022. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, 2016. 3

[5] Y. He, R. Yan, K. Fragkiadaki, and S. Yu. Epipolar transformers. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7776–7785, Los Alamitos, CA, USA, 2020. IEEE Computer Society. 3

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 1997. 2, 8

[7] Taeho Kang, Kyungjin Lee, Jinrui Zhang, and Youngki Lee. Ego3dpose: Capturing 3d cues from binocular egocentric views. In *SIGGRAPH Asia 2023 Conference Papers*, New York, NY, USA, 2023. Association for Computing Machinery. 1, 2, 3, 5, 6, 7

[8] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 3

[9] Jiaman Li, Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17142–17151, 2023. 2

[10] Wenhao Li, Hong Liu, Hao Tang, and Pichao Wang. Multi-hypothesis representation learning for transformer-based 3d human pose estimation. *Pattern Recognition*, page 109631, 2023. 1

[11] Jun Liu, Amir Shahroudy, Dong Xu, Alex C. Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):3007–3021, 2018. 3

[12] Evonne Ng, Donglai Xiang, Hanbyul Joo, and Kristen Grauman. You2me: Inferring body pose in egocentric video via first and second person interactions. *CVPR*, 2020. 2

[13] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei Rezvani Nezhad, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. Egocap: Egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics*, 35, 2016. 2, 5, 6, 7, 8

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. cite arxiv:1505.04597Comment: conditionally accepted at MICCAI 2015. 3

[15] 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. *arXiv preprint arXiv:2303.11579*, 2023. 1

[16] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. xr-egopose: Egocentric 3d human pose from an hmd camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7728–7738, 2019. 1, 2

[17] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *CVPR*, pages 648–656. IEEE Computer Society, 2015. 1, 3

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 4, 7

[19] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, Diogo Luvizon, and Christian Theobalt. Estimating egocentric 3d human pose in the wild with external weak supervision. *CVPR*, 2022. 2

[20] Jian Wang, Diogo Luvizon, Weipeng Xu, Lingjie Liu, Kripasindhu Sarkar, and Christian Theobalt. Scene-aware egocentric 3d human pose estimation. *CVPR*, 2023. 3

[21] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. Mo^{2}Cap^2 : Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019. 2

[22] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. 3

[23] Bruce X.B. Yu, Zhi Zhang, Yongxu Liu, Sheng-hua Zhong, Yan Liu, and Chang Wen Chen. Gla-gcn: Global-local adaptive graph convolutional network for 3d human pose estimation from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8818–8829, 2023. 3

[24] Ailing Zeng, Xiao Sun, Lei Yang, Nanxuan Zhao, Minhao Liu, and Qiang Xu. Learning skeletal graph neural networks for hard 3d pose estimation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11416–11425, 2021. 3

[25] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13232–13242, 2022. 3

[26] Dongxu Zhao, Zhen Wei, Jisan Mahmud, and Jan-Michael Frahm. Egoglass: Egocentric-view human pose estimation from an eyeglass frame. In *2021 International Conference on 3D Vision (3DV)*, pages 32–41, 2021. 2, 3, 6, 7

[27] 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 (CVPR)*, pages 8877–8886, 2023. 3

[28] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 3