Domain-Guided Spatio-Temporal Self-Attention for Egocentric 3D Pose Estimation

Jinman Park
jinman.park@uwaterloo.ca
University of Waterloo, Canada

Kimathi Kaai
kkaai@uwaterloo.ca
University of Waterloo, Canada

Saad Hossain
s42hossa@uwaterloo.ca
University of Waterloo, Canada

Norikatsu Sumi
norikatsu-sumi@mail.nissan.co.jp
Nissan Motor Corporation, Japan

Sirisha Rambhatla
sirisha.rambhatla@uwaterloo.ca
University of Waterloo, Canada

Paul Fieguth
pfieguth@uwaterloo.ca
University of Waterloo, Canada

ABSTRACT

Vision-based ego-centric 3D human pose estimation (ego-HPE) is essential to support critical applications of *x*R-technologies. However, severe self-occlusions and strong distortion introduced by the fish-eye view from the head mounted camera, make ego-HPE extremely challenging. To address these challenges, we propose a domain-guided spatio-temporal transformer model that leverages information specific to ego-views. Powered by this domain-guided transformer, we build Egocentric Spatio-Temporal Self-Attention Network (Ego-STAN), which uses 2D image representations and spatio-temporal attention to address both distortions and self-occlusions in ego-HPE. Additionally, we introduce a spatial concept called *feature map tokens* (FMT) which endows Ego-STAN with the ability to draw complex spatio-temporal information encoded in ego-centric videos. Our quantitative evaluation on the contemporary *x*R-EgoPose dataset, achieves a 38.2% improvement on the highest error joints against the SOTA ego-HPE model, while accomplishing a 22% decrease in the number of parameters. Finally, we also demonstrate the generalization capabilities of our model to real-world HPE tasks beyond ego-views achieving 7.7% improvement on 2D human pose estimation with the Human3.6M dataset. Our code is also made available at: https://github.com/jmpark0808/Ego-STAN.

CCS CONCEPTS

• **Computing methodologies** → **Interest point and salient region detections**; **Computer vision**.

KEYWORDS

Egocentric pose estimation, Spatio-temporal models, Transformers, Self-occlusions in videos, Fish-eye distortion

ACM Reference Format:
Jinman Park, Kimathi Kaai, Saad Hossain, Norikatsu Sumi, Sirisha Rambhatla, and Paul Fieguth. 2023. Domain-Guided Spatio-Temporal Self-Attention for Egocentric 3D Pose Estimation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599312

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599312

1 INTRODUCTION

The rise of virtual immersive technologies, such as augmented, virtual, and mixed reality environments (*x*R) [19, 22, 56], has fueled the need for accurate vision-based human pose estimation (HPE) to support critical applications, such as in medical simulation training [50], among others [5, 13, 15, 26, 32, 44]. Vision-based 3D pose estimation is largely divided on the basis of camera viewpoint: outside-in versus ego-centric view (ego-view). In outside-in 3D HPE, the cameras have a fixed effective recording volume and view angle [6, 27, 33, 60], but such fixed views also lead to lower and less robust accuracies [50]. In contrast, while the mobile and user-centric perspective of ego-view is desired for large-scale cluttered environments [48, 51, 54], it also comes with its unique challenges.

Challenges of Ego-view HPE are vastly different from outside-in views because in ego-views lower body joints are (a) visually much smaller than the upper body joints (*distortion*) and (b) in most cases heavily occluded by the upper torso (*self-occlusion*). Previous outside-in spatio-temporal works attempt to regress 3D pose from an input sequence of 2D keypoints – not images – [8, 57, 59, 63], and focus on mitigating the high output variance of 3D human pose. In addition, outside-in HPE specifically targeting occlusion robust methods require different assumptions such as static camera angles, static background, and supervision on joint visibility [9, 23, 29]. As a result, outside-in pose estimation approaches are not applicable directly, and there is a need to incorporate domain-information specific to ego-views to accomplish accurate HPE.

Recent ego-views HPE works attempt to utilize uncertainty information using a dual-branch autoencoder-based 2D to 3D pose estimator [48] and by incorporating extra camera information [58]. However, a) they only rely on static views, and b) do not consider the complex spatio-temporal interactions across video frames. Moreover, while critical applications of ego-HPE (surgeon training [50]) require accurate and robust estimation of extremities (hands and feet), these methods also suffer from high errors on these very joints, making them unsuitable for these tasks [48, 58].

Summary of our Contributions. Given these challenges, we propose a domain-guided *Egocentric Spatio-Temporal Self-Attention Network* **(Ego-STAN)** to reliably estimate the location of heavily occluded joints and address the distortions in ego-centric views by utilizing spatio-temporal information from a sequence of feature maps. Our contributions can be summarized as follows. First, we design Ego-STAN's hybrid architecture which utilizes spatio-temporal attention endowed by Transformers [49] to self-attend

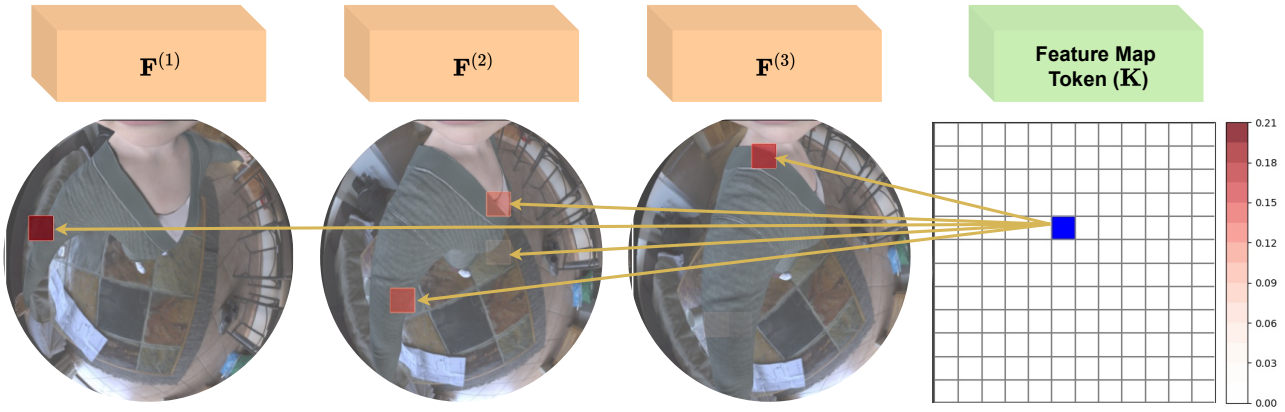


Figure 1: Interpreting Ego-STAN's Attention Mechanism.A sequence of images $\mathbf{I}^{(1)}$, $\mathbf{I}^{(2)}$, and $\mathbf{I}^{(3)}$, yields feature maps $\mathbf{F}^{(1)}$, $\mathbf{F}^{(2)}$, and $\mathbf{F}^{(3)}$, respectively, and are appended with a (learnable) feature map token (K). Ego-STAN's feature map tokens (in blue) can be deconvolved to identify the corresponding attended region(s) in the image sequence (in red), to allow the interpretation of information aggregation from the images.

to a sequence of semantically rich feature maps extracted by Convolutional Neural Networks (ResNet-101) [17]. Complementary to this architecture, we also propose an ℓ_1-based loss function to accomplish robust pose estimation, handling both self-occlusions and visibly difficult (low resolution) joints. Second, to leverage the complex spatio-temporal information encoded in ego-centric videos, we design *feature map token* (FMT), learnable parameters that, alongside our spatio-temporal Transformer, can globally attend to all spatial units of the extracted sequential feature maps to draw valuable information. FMT also provides interpretability, revealing the complex temporal dependence of the attention (Fig. 1). Third, to address the domain-specific distortions, we propose a 2D heatmap to 3D pose regression module significantly reduces the overall MPJPE and the number of trainable parameters as compared the SOTA [48]. We also indirectly evaluate the advantages of this module via HPE on the Mo^2Cap^2 dataset (static ego-HPE) and on the Human3.6M dataset. Finally, we perform comprehensive ablation studies to analyze the impact of each component of Ego-STAN. These ablations thoroughly demonstrate that the composition of the Transformer network, ℓ_1 loss, Direct 3D regression, and the FMT, lead to the superior performance of Ego-STAN. Additionally, these studies reveal a surprising fact: the auto-encoder-based architectures recommended by SOTA may be creating information bottlenecks and be counterproductive for ego-HPE.

On the SOTA sequential ego-views dataset *x*R-EgoPose [48], it achieves an average **improvement of 38.2%** mean per-joint position error (MPJPE) on the highest error joints against the SOTA egocentric pose estimation work [48] while **reducing 22% trainable parameters** on the *x*R-EgoPose dataset [48]. Furthermore, Ego-STAN generalizes to other HPE tasks in static ego-views with an improvement of 11.4% on the Mo^2Cap^2 dataset [54], and outside-in views on the Human3.6M dataset [20] where it reduces the MPJPE by 9% against [48] and improving 2D human pose estimation by 7.7% against [42], demonstrating its ability to generalize to real-world views and adapt to other HPE scenarios. In addition, we also evaluate Ego-STAN on the Human3.6M, an outside-in sequential HPE dataset, showing an improvement of 8% on Percentage of Correct Keypoint (PCK) of 2D joint detection demonstrating the versatility of the proposed attention architecture and FMT.

2 RELATED WORK

This section discusses related work on 3D HPE, for both static (single frame) and sequential (multi-frame) models, alongside Transformer-based self-attention, on two specific camera viewpoints: (1) an outside-in viewpoint, the image capture of a subject from a distance, (2) an egocentric viewpoint, wherein the subject is captured from a head-mounted camera.

Outside-in Static Human Pose Estimation initially regressed directly to 3D pose from images, without intermediate 2D representation [28, 36, 37, 43, 45]; [36, 37] considered the use of volumetric heatmaps to utilize 3D features in images on popular outside-in datasets such as Human3.6M. Soon after, many works applied 2D to 3D lifting models [6, 27, 33, 60], taking advantage of accurate 2D pose for 3D tasks. Works showed that joint estimation of 2D and 3D poses is advantageous both in supervised and unsupervised settings [40, 41]. Our work leverages the supervised joint estimation of 2D and 3D poses, building on [40]. We demonstrate Ego-STAN's ability to overcome occlusions and generalize to these scenarios as compared to popular 2D HPE (HRNet) [42] and SOTA ego-HPE [48] methods.

Outside-in Video 3D Human Pose Estimation utilizes the temporal information of video to improve 3D HPE [4, 9, 10, 46, 52, 61, 62]. More recently, these include the use of deep learning-based sequential models such as Long Short-Term Memory (LSTM) [18] and spatio-temporal relations via temporal Convolutional Neural Networks (CNN) [38]. Enforcing temporal consistency using bone length and direction has been proposed for HPE [7]. More recently, Transformer-based attention mechanisms have gained popularity for factoring in frame significance and receptor-field dependency [8, 30, 57, 63]. Others proposed explicit visibility guidance by pseudo-labeling or human labels [9, 23, 29].

Transformers for 3D Pose Estimation have shown remarkable success in a number of application areas [49], including computer vision via the introduction of Vision Transformers (ViT) [12]. Recent efforts focus on combining CNNs and self-attention mechanisms to reduce parameters and allow for lightweight networks for vision applications [34]. In representing temporal phenomena, Transformers have made their way into many different spatio-temporal tasks [1, 39, 53, 55]. For example, in action recognition, [3] fully utilizes Transformers for feature extraction, while in video object segmentation, [35] extracts features with a CNN backbone. For outside-in HPE, PoseFormer [59] utilized a spatio-temporal sequence of 2D keypoints from an off-the-shelf 2D pose estimator to predict the 3D pose. Unlike PoseFormer, the distorted egocentric views preclude us from using such off-the-shelf methods. Ego-STAN addresses this challenge through the supervised 2D heatmap estimation.

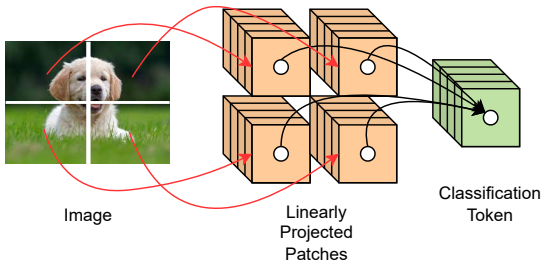
Egocentric Human Pose Estimation. The Mo^2Cap^2 dataset was one of the first large HPE synthetic, single-frame datasets with a cap-mounted fish-eye egocentric camera with static views in the train set and sequences in the test [54]. While Ego-STAN primarily relies on spatio-temporal information in sequential (multi-frame) inputs, it yields competitive results with respect to SOTA on the Mo^2Cap^2 single-frame dataset, demonstrating the effectiveness of direct 3D regression over an autoencoder-based model [48]. More recently, the xR-EgoPose dataset, a sequential ego-views dataset, was released, offering a larger (and more realistic) dataset [48]. The work also proposed a single and a dual-branch auto-encoder structure for 3D ego-HPE. Focusing on the fish-eye distortion, [58] use camera parameters in training. Next, to address the depth ambiguity and temporal instability in egocentric HPE, GlobalPose [51] proposed a sequential variational auto-encoder-based model, that uses [48] as a submodule. Since Ego-STAN accomplishes significant improvements over [48] by leveraging spatio-temporal information, it can also be used with GlobalPose [51].

3 EGO-STAN

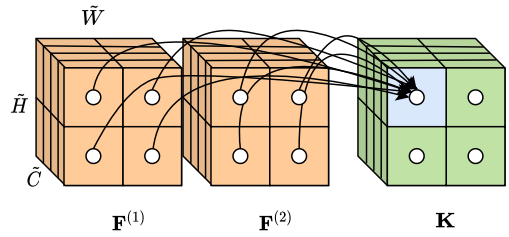
We now develop the proposed **Ego**centric **S**patio-**T**emporal Self-**A**ttention **N**etwork (Ego-STAN) model, shown in Fig 3, which jointly addresses the self-occlusion and the distortion introduced by the ego-centric views. In doing so, we also conduct an in-depth analysis of the relationship between the 2D heatmap and 3D pose estimation. Ego-STAN consists of four modules. Of these, the **feature extraction** and **spatio-temporal Transformer** modules aim to address the self-occlusion problem by regressing information from multiple time steps, while the **heatmap reconstruction** and **3D pose estimator** modules accomplish uncertainty saturation with lighter 2D-to-3D lifting architectures. Our code is made available at: https://github.com/jmpark0808/Ego-STAN

3.1 Feature extraction

The feature extraction module in Fig. 3 extracts feature maps that identify regions of interest from ego-centric images via multiple non-linear convolutional filters. Building on a ResNet-101 [17] backbone for extracting image-level features, we introduce a specialized set of learnable parameters – *feature map token* (FMT) – utilized



(a) Classification token [11, 12]



(b) Feature map token (this work).

Figure 2: Difference between classification token (top) and feature map token (FMT) (bottom). Classification tokens aggregate feature map information to a single semantic vector that is passed to a classification head. Feature map tokens are a collection of globally aggregated vectors that are deconvoluted to estimate 2D heatmaps.

by our Transformer to draw valuable pose information across time-steps. By combining information from different time-steps, Ego-STAN accomplishes 2D heatmap estimation even in challenging cases where views suffer from extreme occlusions, as follows.

ResNet-101. Ego-STAN leverages the intermediate ResNet-101 representations to form image-level feature maps. Let $R(\cdot)$ represent ResNet-101's non-linear function that extracts a feature map from a given image $\mathbf{I} \in \mathbb{R}^{H\times W\times C}$ of height H, width W and channels C. Then, given an image sequence $\mathbb{I}_T = \{\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, ..., \mathbf{I}^{(T)}\}$ of length T, where $\mathbf{I}^{(t)}$ is an image at time t, we obtain a sequence of feature maps $\mathbb{F}_T = \{\mathbf{F}^{(1)}, \mathbf{F}^{(2)}, ..., \mathbf{F}^{(T)}\}$ by applying $R(\cdot)$ to each image to form $\mathbf{F}^{(t)} \in \mathbb{R}^{\tilde{H}\times\tilde{W}\times\tilde{C}}$ as

$$\mathbf{F}^{(t)} = R(\mathbf{I}^{(t)}). \tag{1}$$

Feature map token. To leverage information from past frames to counter occlusions, we require a way to aggregate input feature maps over different times-steps. Specifically, dynamic aggregation to address variable magnitudes of occlusions over frames. To this end, we propose learnable parameters – feature map token (FMT) $\mathbf{K} \in \mathbb{R}^{\tilde{H}\times\tilde{W}\times\tilde{C}}$ – which learns where to pay attention for feature aggregation in conjunction with self-attention [49]. FMT are related to recent works which introduce learnable parameters for classification or *classification tokens* [11] with some key differences which make them a powerful way to aggregate information. As shown in Fig. 2, while a classification token is a single unit token that computes a weighted sum of the feature representations specifically for classification, our proposed feature map token has multiple feature map points, each of which can aggregate from all semantic tokens

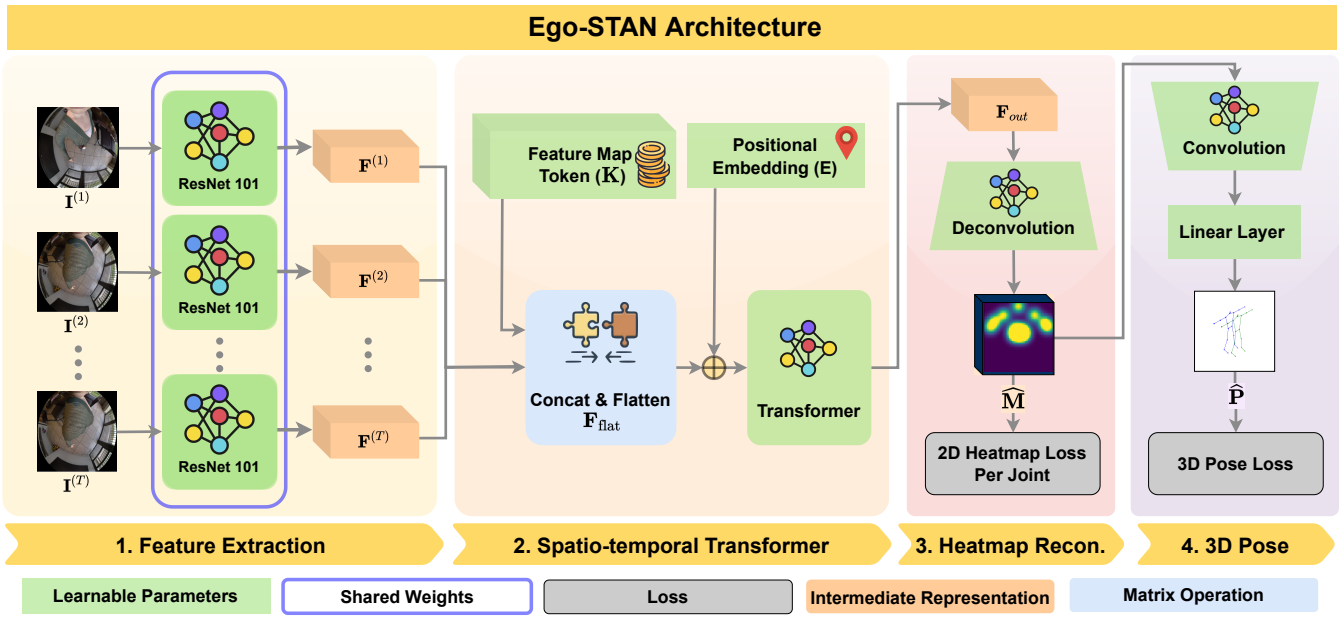


Figure 3: Ego-STAN Overview. The proposed Ego-STAN model captures the dynamics of human motion in ego-centric images using Transformer-based spatio-temporal modeling. Ego-STAN uses ResNet-101 as a feature extractor. The proposed Transformer architecture leverages *feature map token* to facilitate spatio-temporal attention to semantically rich feature maps. Our heatmap reconstruction module estimates the 2D heatmap using deconvolutions, which are used by the 3D pose estimator to estimate the 3D joint coordinates.

that are distributed spatially and temporally based on the attention weights, corresponding to a particular location in an image for intermediate 2D heatmap representation. As a result, each unit of the FMT $\mathbf{K}$ learns how to represent accurate semantic features for the heatmap reconstruction module. Furthermore, the corresponding attention matrix can be visualized for interpretability as shown in Fig. 1. We randomly initialize the token, $\mathbf{K}$, and concatenate it with the feature map sequence $\{\mathbf{F}^{(t)}\}_{t=1}^{T}$, denoted by $\texttt{Concatenate}(\cdot)$ along the $\tilde{W}$ dimension to obtain $\mathbf{F}_{\text{concat}} \in \mathbb{R}^{\tilde{H}\times\tilde{W}(T+1)\times\tilde{C}}$ as

$$\mathbf{F}_{\text{concat}} := \texttt{Concatenate}(\mathbf{K}, \mathbb{F}_T) = [\mathbf{K}, \mathbf{F}^{(1)}, \mathbf{F}^{(2)}, ..., \mathbf{F}^{(T)}]. \quad (2)$$

We flatten the non-channel dimensions with the $\texttt{Flatten}(\cdot)$ operation (mode-3 fibers [25]) in order to serialize the input for the Transformer module to obtain $\boldsymbol{F}_{\text{flat}} \in \mathbb{R}^{\tilde{H}\tilde{W}(T+1)\times\tilde{C}}$ as

$$\boldsymbol{F}_{\text{flat}} := \texttt{Flatten}(\mathbf{F}_{\text{concat}}). \quad (3)$$

3.2 Spatio-temporal Attention using Feature Map Token

Now that we have $\boldsymbol{F}_{\text{flat}}$, that contains both feature maps from multiple time steps and the feature map token, we are ready for spatio-temporal learning. Self-attention learns to map the pairwise relationship between *input tokens* $\boldsymbol{F}_{\text{flat}[r,:]}$ for $r = \{1, \ldots, \tilde{H}\tilde{W}(T+1)\}$. This is especially important because it allows the feature map token $\mathbf{K}$ (the first $\tilde{H}\tilde{W}$ rows in $\mathbf{F}_{\text{flat}}$) to look across all of the input tokens in the spatio-temporally distributed sequences to learn where to pay attention.

Positional Embedding. Transformer networks need explicit information about the relative position of input tokens [49]. As our input space often has repetitive background or body positions, it is important to inject positional guidance in order for the network to be able to distinguish identical input tokens. To accomplish this, we add a learnable position embedding $\boldsymbol{E} \in \mathbb{R}^{\tilde{H}\tilde{W}(T+1)\times\tilde{C}}$ element-wise to $\boldsymbol{F}_{\text{flat}}$ to form $\boldsymbol{F}_{\text{pe}} \in \mathbb{R}^{\tilde{H}\tilde{W}(T+1)\times\tilde{C}}$ as

$$\boldsymbol{F}_{\text{pe}} = \boldsymbol{F}_{\text{flat}} + \boldsymbol{E}. \quad (4)$$

Self-attention with Feature Map Token. Our Transformer module – $\texttt{Transformer}(\cdot)$ – encodes spatio-temporal information in feature map $\boldsymbol{F}_{\text{pe}}$ with self-attention and returns $\boldsymbol{F}_{\text{tfm}} \in \mathbb{R}^{\tilde{H}\tilde{W}(T+1)\times\tilde{C}}$. Ego-STAN learns FMT weights, $\mathbf{K}$, and the linear projections of the Transformer encoder [49] to understand which tokens are important in the sequence via a hybrid CNN backbones and Transformers motivated from [34, 35]. In the self-attention module, there are three sets of learnable parameters (implemented as a linear layer) that enable this dynamic aggregation via FMT $\boldsymbol{L}_{\text{q}} \in \mathbb{R}^{\tilde{C}\times D}$, $\boldsymbol{L}_{\text{r}} \in \mathbb{R}^{\tilde{C}\times D}$, and $\boldsymbol{L}_{\text{v}} \in \mathbb{R}^{\tilde{C}\times D}$, which are used to form query $\boldsymbol{Q}$, key $\boldsymbol{R}$, and value $\boldsymbol{V}$ for the Transformer module as

$$\boldsymbol{Q} := \boldsymbol{F}_{\text{pe}}\boldsymbol{L}_{\text{q}}, \qquad \boldsymbol{R} := \boldsymbol{F}_{\text{pe}}\boldsymbol{L}_{\text{r}}, \qquad \boldsymbol{V} := \boldsymbol{F}_{\text{pe}}\boldsymbol{L}_{\text{v}}. \quad (5)$$

Given these matrices, the attention matrix $\boldsymbol{A} \in \mathbb{R}^{\tilde{H}\tilde{W}(T+1)\times\tilde{H}\tilde{W}(T+1)}$ is computed as

$$\boldsymbol{A} := \texttt{Softmax}(\boldsymbol{Q}\boldsymbol{R}^{\top}), \quad (6)$$

and the subsequent aggregation $\boldsymbol{A}_{\text{v}}$ using the value matrix $\boldsymbol{V}$ as

$$\boldsymbol{A}_{\text{v}} := \boldsymbol{A}\boldsymbol{V}. \quad (7)$$

Finally, $\boldsymbol{A}_{\text{v}}$ is passed through the feed forward block to form $\boldsymbol{F}_{\text{tfm}}$. These three learnable parameters can therefore dynamically determine the aggregation weights depending on the semantics of the

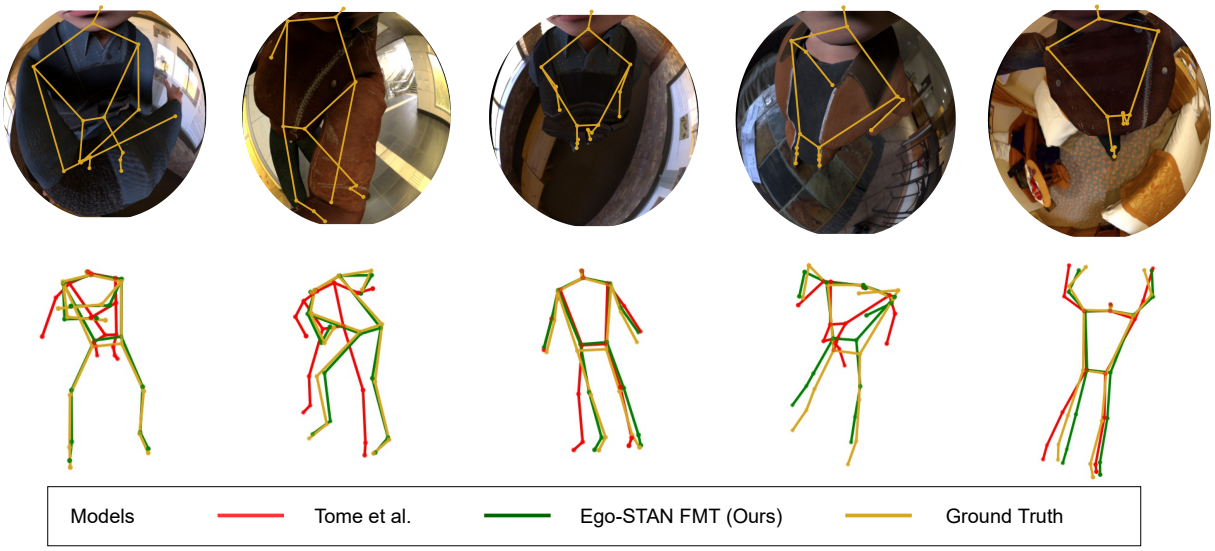


Figure 4: Qualitative evaluation on highly occluded frames. We demonstrate the qualitative performance of Ego-STAN with feature map token (FMT), compared with the SOTA dual-branch model [48] on self-occluded frames. The top row shows the frames superimposed with the ground truth 2D joint location skeleton (in gray). We observe that Ego-STAN is significantly more robust to occlusions relative to the dual-branch model [48].

feature maps; can be on independent interest in application that require aggregation of semantics from the feature maps that are distributed spatio-temporally. Note that this aggregation is for a single head in a multi-head attention module. Finally, the action of our Transformer module can be represented as

$$\boldsymbol{F}_{\text{tfm}} := \texttt{Transformer}(\boldsymbol{F}_{\text{pe}}) \tag{8}$$

$$\text{or alternatively } \boldsymbol{F}_{\text{tfm}} := \texttt{FeedForward}(\boldsymbol{A}_{\text{v}}). \tag{9}$$

We only take the first $\tilde{H}\tilde{W}$ tokens corresponding to the feature map token $\mathbf{K}$ from $\boldsymbol{F}_{\text{tfm}}$ and reshape into a $\tilde{H} \times \tilde{W} \times \tilde{C}$ tensor to form the spatio-temporal Transformer output $\mathbf{F}_{\text{out}} \in \mathbb{R}^{\tilde{H}\times\tilde{W}\times\tilde{C}}$ as

$$\mathbf{F}_{\text{out}} := \texttt{Reshape}(\boldsymbol{F}_{\text{tfm}[:\tilde{H}\tilde{W},:]}). \tag{10}$$

As a result, these modules, and specifically the feature map token, create an accurate semantic map for heatmap reconstruction (further discussed in section 3.3).

Slice and average variant. To explore the impact of FMT, we compare with two spatio-temporal model variants without FMT. The first variant is called *slice*. Since we are interested in estimating the 3D pose of the current frame from given a past frame sequence, we take the indices of the tokens that are respective to the current frame in the token sequence. Given a sequence of tokens (without FMT), we take the last $\tilde{H}\tilde{W}$ indices from $\boldsymbol{F}_{\text{tfm}} \in \mathbb{R}^{\tilde{H}\tilde{W}T\times\tilde{C}}$ to be deconvolved. Formally we have:

$$\mathbf{F}_{\text{out-slice}} := \texttt{Reshape}(\boldsymbol{F}_{\text{tfm}[-\tilde{H}\tilde{W}:,:]}). \tag{11}$$

The *avg* variant reduces the spatial dimension by averaging over spatially same but temporally different tokens. Specifically, we take $\boldsymbol{F}_{\text{tfm}} \in \mathbb{R}^{\tilde{H}\tilde{W}T\times\tilde{C}}$ from (9) and average over the T dimension,

$$\mathbf{F}_{\text{out-avg}} := \texttt{Average}(\boldsymbol{F}_{\text{tfm}}, \text{dim} = T). \tag{12}$$

3.3 Heatmap reconstruction

Feature map to heatmap. Our goal is to leverage deconvolution layers to reconstruct ground truth 2D heatmaps of time T, $\mathbf{M} \in \mathbb{R}^{h\times w\times J}$, of height and width, $h \times w$, for each major joint in the human body (J). To this end, $\mathbf{F}_{\text{out}}$ is passed through two deconvolution layers to estimate $\widehat{\mathbf{M}} =\in \mathbb{R}^{h\times w\times J}$ as

$$\widehat{\mathbf{M}} := \texttt{Deconv}(\mathbf{F}_{\text{out}}), \tag{13}$$

trained via a mean square error, $\texttt{MSE}(\cdot)$-based loss $\mathcal{L}_{2D}$:

$$\mathcal{L}_{2D}(\mathbf{M}, \widehat{\mathbf{M}}) = \texttt{MSE}(\mathbf{M}, \widehat{\mathbf{M}}). \tag{14}$$

3.4 3D pose estimation

Heatmap to pose. We leverage a simple convolution block followed by linear layers to lift the 2D heatmaps to 3D poses. As opposed to the SOTA egocentric pose estimator [48], which uses a dual branched auto-encoder structure aimed at preserving the uncertainty information from 2D heatmaps, we (somewhat surprisingly) find that complex auto-encoder design is in fact not required, and our simple architecture accomplishes this task more accurately (see section 4.1). Therefore, given the predicted heatmap $\widehat{\mathbf{M}}$, we predict the 3D coordinates of the joints $\widehat{\boldsymbol{P}} \in \mathbb{R}^{J\times 3}$ at time T as

$$\widehat{\boldsymbol{P}} := \texttt{Linear}\big(\texttt{Convolution}(\widehat{\mathbf{M}})\big). \tag{15}$$

To estimate the 3D pose using the reconstructed 2D heatmaps (14), we use three different types of loss functions – i) squared ℓ_2-error $\mathcal{L}_{\ell_2}(\cdot)$, ii) cosine similarity $\mathcal{L}_{\theta}(\cdot)$, and iii) ℓ_1-error $\mathcal{L}_{\ell_1}(\cdot)$ between $\widehat{\boldsymbol{P}}$ and $\boldsymbol{P}$. These loss functions impose the closeness between $\boldsymbol{P}$ and $\widehat{\boldsymbol{P}}$ in multiple ways. As compared to [48], our ℓ_1-norm promotes the solutions to be robust to outliers [21], as corroborated by our ablations in section 4.2. As a result, our 3D loss for regularization

Table 1: Comparative quantitative evaluation of Ego-STAN against the SOTA Ego-HPE methods. Proposed Ego-STAN variants have the highest accuracies across the nine actions with the feature map token (FMT) variant having the lowest overall MPJPE (lower is better); our results are averaged over three random seeds.

Approach	Evaluation error (mm)	Game	Gest.	Greet	Lower Stretch	Pat	React	Talk	Upper Stretch	Walk	All
Martinez et. al. [33]	Upper body	58.5	66.7	54.8	70.0	59.3	77.8	54.1	89.7	74.1	79.4
	Lower body	160.7	144.1	183.7	181.7	126.7	161.2	168.1	159.4	186.9	164.8
	Average	109.6	105.4	119.3	125.8	93.0	119.7	111.1	124.5	130.5	122.1
Tome et. al. [48] single-branch	Upper body	114.4	106.7	99.3	90.0	99.1	147.5	95.1	119.0	104.3	112.5
	Lower body	162.2	110.2	101.2	175.6	136.6	203.6	91.9	139.9	159.0	148.3
	Average	138.3	108.5	100.3	133.3	117.8	175.6	93.5	129.0	131.9	130.4
Tome et. al. [48] dual-branch	Upper body	48.8	50.0	43.0	36.8	48.6	56.4	42.8	49.3	43.2	50.5
	Lower body	65.1	50.4	46.1	65.2	70.2	65.2	45.0	58.8	72.2	65.9
	Average	56.0	50.2	44.6	51.5	59.4	60.8	43.9	53.9	57.7	58.2
Zhang et. al. [58]	Upper body	-	-	-	-	-	-	-	-	-	-
	Lower body	-	-	-	-	-	-	-	-	-	-
	Average	36.8	34.1	36.7	50.1	57.2	34.4	32.8	54.3	52.6	50.0
Liu et. al. [31]	Upper body	29.7	29.5	39.3	45.9	28.8	26.7	29.4	60.2	50.0	50.6
	Lower body	47.0	35.6	66.4	129.5	77.4	39.6	40.2	136.6	129.4	116.4
	Average	37.0	32.4	51.1	90.0	49.0	32.6	34.0	100.5	89.2	84.7
Tome et. al. [47] Self-Pose	Upper body	-	-	-	-	-	-	-	-	-	-
	Lower body	-	-	-	-	-	-	-	-	-	-
	Average	60.4	54.6	44.7	56.5	57.7	52.7	56.4	53.6	55.4	54.7
Ego-STAN Slice (Ours)	Upper body	27.2	30.0	36.3	24.0	21.3	25.4	25.3	34.2	25.5	30.2
	Lower body	38.5	**30.9**	**33.2**	54.5	**32.1**	35.6	**29.5**	64.0	**55.9**	55.5
	Average	32.9	30.4	34.8	39.2	**26.7**	30.5	**27.4**	49.1	**40.7**	42.8
Ego-STAN Avg. (Ours)	Upper body	**25.4**	**26.7**	**31.2**	25.9	**20.7**	**23.3**	**23.9**	33.7	26.7	29.9
	Lower body	**38.1**	32.7	35.0	54.7	34.6	**34.3**	31.2	61.2	57.2	54.3
	Average	**31.7**	**29.7**	**33.1**	40.3	27.7	**28.8**	27.5	47.4	42.0	42.1
Ego-STAN FMT (Ours)	Upper body	25.8	28.7	35.4	**23.4**	22.6	24.1	25.9	**30.9**	**25.2**	**28.2**
	Lower body	40.3	34.5	38.3	**54.4**	35.9	35.0	33.4	**57.6**	56.5	**52.6**
	Average	33.1	31.6	36.9	**38.9**	29.2	29.6	29.7	**44.3**	40.9	**40.4**

parameters λ_θ and λ_{ℓ_1} is

$$\mathcal{L}_{3D}(\boldsymbol{P}, \widehat{\boldsymbol{P}}) = \mathcal{L}_{\ell_2}(\boldsymbol{P}, \widehat{\boldsymbol{P}}) + \lambda_\theta \mathcal{L}_\theta(\boldsymbol{P}, \widehat{\boldsymbol{P}}) + \lambda_{\ell_1} \mathcal{L}_{\ell_1}(\boldsymbol{P}, \widehat{\boldsymbol{P}}) \quad (16)$$

$$\text{where, } \mathcal{L}_{\ell_2}(\boldsymbol{P}, \widehat{\boldsymbol{P}}) := \|\widehat{\boldsymbol{P}} - \boldsymbol{P}\|_2^2,\ \mathcal{L}_\theta(\boldsymbol{P}, \widehat{\boldsymbol{P}}) := \sum_{i=1}^{J} \frac{\langle \boldsymbol{P}_i, \widehat{\boldsymbol{P}}_i \rangle}{\|\boldsymbol{P}_i\|_2 \|\widehat{\boldsymbol{P}}_i\|_2}, \quad (17)$$

$$\text{and } \mathcal{L}_{\ell_1}(\boldsymbol{P}, \widehat{\boldsymbol{P}}) := \sum_{i=1}^{J} \|\widehat{\boldsymbol{P}}_i - \boldsymbol{P}_i\|_1.$$

Thus, the overall loss function to train Ego-STAN comprises of the 2D heatmap reconstruction loss and the 3D loss, as shown in (14) and (16), respectively.

3.5 Intuitive explanation on Feature Map Token

We will summarize section 3.1 and section 3.2 with some notes. FMT begins as a set of randomly initialized weights with the same dimensions as a single feature map ($\mathbf{K} \in \mathbb{R}^{\tilde{H} \times \tilde{W} \times \tilde{C}}$). Then these weights are concatenated and flattened to a sequence of T feature maps ($T \times \tilde{H} \times \tilde{W} \times \tilde{C}$) returning $\mathbf{F}_{\text{flat}} \in \mathbb{R}^{\tilde{H}\tilde{W}(T+1) \times \tilde{C}}$. Positional embedding is then added to $\mathbf{F}_{\text{flat}}$ to inject spatial and temporal distinction and passed through the Transformer block to return $\mathbf{F}_{\text{tfm}} \in \mathbb{R}^{\tilde{H}\tilde{W}(T+1) \times \tilde{C}}$. What this output implies is that all the tokens

($\tilde{H}\tilde{W}(T+1)$) are aggregated based on the normalized attention matrices. Once we take the indices of the FMT (which are concatenated at the beginning), we are left with FMT that has been aggregated with the feature maps that are distributed spatially and temporally. Intuitively, the weights of FMT are updated so that it understands where to pay attention to, given a sequence of feature maps from the CNN backbone. In other words, FMT learns how to position its direction of the token vectors so that given a set of feature maps of certain visibility (occlusion), the linear projections Q and K can determine the weight of the attention matrix for aggregation on the past or the current frame.

4 EXPERIMENTS

We now analyze the performance of Ego-STAN as compared to the SOTA ego-HPE methods. Additionally, we carry-out a systematic analysis of the incremental contributions by each component of Ego-STAN via extensive ablations. Our code is made available at: https://github.com/jmpark0808/Ego-STAN

Overview of experiments. We analyze the performance on the xR-EgoPose dataset [48], the only dataset with a sequential ego-view training set for detailed ablations and analysis via the Mean Per-Joint Position Error (MPJPE) metric, to the best of our knowledge. In addition, we also evaluate on Human3.6M [20], an outside-in sequential real-world 3D HPE dataset, and on Mo^2Cap^2 [54], an ego-HPE dataset with static synthetic train set and real sequential test using MPJPE, to analyze generalization, and adaptability with other pose estimation backbones. On Human 3.6M, we compare the results with and without Ego-STAN on a popular outside-in HPE method [42] and also against the SOTA ego-HPE model [48]. Here, in addition to MPJPE, we also report the Percentage of Correct Keypoint (PCK), a popular metric for Human3.6M, to gauge 2D joint estimation accuracy. Since 3D HPE crucially depends on accurate heatmap (2D) estimation, PCK reveals the capabilities of learned representations. Our results report the average performance across three random seeds; details to allow *reproducibility and the code* are listed in A.1 and in the supplementary materials.

Evaluation Metrics. The standard metric for 3D HPE is MPJPE (Mean Per Joint Position Error). It is measured by taking the ℓ_2-norm of the difference between predicted joint coordinates $\hat{\mathbf{P}}_j^{(n)}$ and the ground truth coordinates $\mathbf{P}_j^{(n)}$ and averaging across all frames and joints in the following way:

$$E_{\text{overall}}(\mathbf{P}, \hat{\mathbf{P}}) = \frac{1}{N}\frac{1}{J}\sum_{n=1}^{N}\sum_{j=1}^{J}\left\|\mathbf{P}_j^{(n)} - \hat{\mathbf{P}}_j^{(n)}\right\|_2 \qquad \text{(Overall MPJPE)}$$

where N, and J are total number of frames, and number of joints respectively. Per-joint MPJPE only averages across the number of frames and reports individual joints ℓ_2-norm averages:

$$E_{\text{per-joint}}(\mathbf{P}, \hat{\mathbf{P}}) = \frac{1}{N}\sum_{n=1}^{N}\left\|\mathbf{P}^{(n)} - \hat{\mathbf{P}}^{(n)}\right\|_2 \qquad \text{(Per-joint MPJPE)}$$

For 2D HPE, Percentage of Correct Keypoint (PCK) is commonly used to measure the accuracy of keypoint detection. It is measured by converting the heatmap prediction to coordinates and then counting the number of correct keypoints respective to the number of total keypoints. PCK is normally accompanied by an arbitrary normalized threshold that indicates the distance respective to the image dimension that the predictions can be off from the label to be considered correct. Formally, given prediction coordinates $\widehat{C} \in \mathbb{R}^{J\times 2}$ and label coordinates $C \in \mathbb{R}^{J\times 2}$, with a threshold m, PCK with respect to a single frame $PCK^{(n)} : n \in N$ where N is the total number of frames, is measured as follows:

$$PCK^{(n)} = \frac{1}{J}\sum_{j=1}^{J}\mathbf{1}_{||\widehat{\mathrm{C}}_{\mathrm{j}}^{(\mathrm{n})} - \mathrm{C}_{\mathrm{j}}^{(\mathrm{n})}||_2 \,<\, \mathrm{m}} \tag{18}$$

Here the x and y coordinates of $\widehat{C}$ and C are normalized by the horizontal and vertical heatmap dimensions. The PCK for the total set of frames PCK_{total} is simply the average PCK of each frame:

$$PCK_{total} = \frac{1}{N}\sum_{n=1}^{N} PCK^{(n)} \tag{19}$$

xREgoPose Dataset. The xREgoPose synthetic dataset was designed to focus on scalability with augmentation of characters, environments, and lightning conditions. It has a total of 383K images, which are split into three sets: Train-set: 252K images; Test-set 115K images; and Validation-set: 16K images. The gender distribution for each set is the following: Train-set: 13M/11F, Test-set: 7M/5F and Validation-set: 3M/2F. The partitioning of the dataset based on actions and the details about the dataset setup can be referred to [48].

4.1 Results

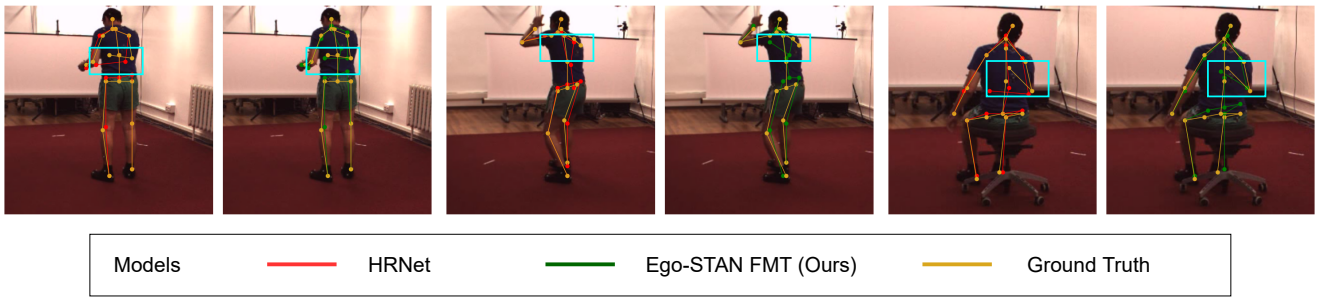
Results (xR-EgoPose). Tab. 1 shows the MPJPE achieved by Ego-STAN and its variants on the xR-EgoPose test set, as compared to SOTA ego-HPE models [48] (a dual-branch autoencoder model, and its single branch variant), a popular outside-in baseline [33], and [58]. For fair comparison, since [58] requires camera parameters for training, we compare against the dual-branch model of [48]. Ego-STAN variants perform the best across different actions and individual joints, as shown in Tab. 1 and Fig. 7, respectively, with Ego-STAN FMT achieving the best average performance. Ego-STAN FMT outperforms the dual-branch model proposed in [48] by a substantial **17.8 mm (30.6%)**, averaged over all actions and joints (Tab. 1). Remarkably, across joints in Fig. 7, **Ego-STAN FMT shows an improvement of 40.9 mm (39.4%) on joints with the highest error in the SOTA [48], with an average improvement of 35.6 mm (38.2%)** over these (left hand, right hand, left foot, right foot, left toe base, and right toe base) joints. Ego-STAN FMT is also most robust to occlusions evident from the lower and upper stretching actions, which suffer from heaviest occlusions (Fig. 7(b)). This robustness is also exhibited by Ego-STAN variants in the violin plots shown in Tab. 3. For a qualitative comparison, we show the estimation results on a few highly self-occluded frames in Fig. 4, further demonstrating the superior properties of Ego-STAN FMT over the SOTA ego-HPE methods.

Results (Mo^2Cap^2). Since Mo^2Cap^2 contains a static train set, this allows us to analyze the impact of direct 3D regression. Here, Ego-STAN improves the MPJPE by 10% on the Mo^2Cap^2 test set over the SOTA [48]. Additional details in section A.2.3 and Tab. 6.

Results (Human3.6M). Outside-in views do not suffer from the same level of self-occlusions and distortions. As a result, our results

Table 2: Quantitative evaluation on Human3.6M for HPE. Accuracy of both 2D HPE and 3D HPE are improved with Ego-STAN even under high occlusions; here Sld: Shoulder, Elb: Elbow.

Approach (2D, PCK@0.05, ↑)	Sld	Elb	Wrist	Hip	Knee	Ankle	Spine	All
Sun [42]	0.763	0.761	0.713	0.807	0.916	0.921	0.900	0.847
Sun [42] + **Ego-STAN**	**0.941**	**0.851**	**0.781**	**0.918**	**0.923**	**0.933**	**0.950**	**0.912**
Approach (3D, MPJPE(mm), ↓)	**Sld**	**Elb**	**Wrist**	**Hip**	**Knee**	**Ankle**	**Spine**	**All**
Tome [48] Protocol 1	131.7	172.9	209.1	42.0	125.9	178.8	74.2	119.4
Ego-STAN (ours) Protocol 1	**122.5**	**163.5**	**198.4**	**30.4**	**95.8**	**125.6**	**63.5**	**109.3**
Tome [48] Protocol 2	51.0	113.5	134.5	67.8	84.3	108.7	**43.4**	73.8
Ego-STAN (ours) Protocol 2	**40.6**	**94.0**	**128.2**	**76.0**	**70.0**	**89.4**	44.8	**68.9**



(a) Directions (b) Taking Photograph (c) Sitting

Figure 5: Qualitative evaluation on Human3.6M dataset. We demonstrate the qualitative performance of Ego-STAN on occluded frames of Human3.6M. Compared to a popular static 2D outside-in HPE method [42], Ego-STAN better estimates occluded joints (highlighted with light blue box). Video attached to sup. material.

highlight Ego-STAN's ability to leverage spatio-temporal information via FMT, of independent interest for HPE in-general. As demonstrated in Tab. 2, wrapping Ego-STAN on a 2D HPE backbone improves the PCK by 8%, underscoring its adaptability and its ability to generalize to real-world data. Moreover, the improvements of 9% on Protocol 1 and 7% on Protocol 2 against the SOTA egocentric HPE [48] strengthens the point that Ego-STAN can be used for real-world data. Additional details are presented in section A.2.2.

4.2 Ablation Studies

We perform a series of ablation studies on xR-EgoPose to analyze the incremental effect of each element of Ego-STAN. We begin by presenting short descriptions of these elements. Here, + represents the addition of certain element and Δ indicates replacing an element with another.

- **Baseline.** Reproduced model [48] trained by $\mathcal{L}_{2D}(\boldsymbol{M}, \widehat{\boldsymbol{M}})$ (14), $\mathcal{L}_{\ell_2}(\boldsymbol{P}, \widehat{\boldsymbol{P}})$, & $\mathcal{L}_{\theta}(\boldsymbol{P}, \widehat{\boldsymbol{P}})$ (15).
- **+ ℓ_1−norm.** Above **Baseline** with the addition of $\mathcal{L}_{\ell_1}(\boldsymbol{P}, \widehat{\boldsymbol{P}})$ in the cost function in section 3.3.
- **+ Temporal TFM.** Temporal Transformer (TFM) which attends to the sequence of latent vectors produced by the autoencoder structure in the **Baseline** + ℓ_1−**norm**.
- **Δ Direct 3D Regression.** Replaces the dual branch autoencoder and the Temporal TFM with a simple neural network to directly regress to 3D pose from heatmaps; see section 3.4,
- **+ Spatial-only TFM.** Addition of self-attention on the feature map generated by a single frame.
- **+ Ego-STAN w/ Slice.** Addition of temporal attention leads to Ego-STAN. This variant of Ego-STAN uses sliced tokens of the current frame (11).
- **Δ Ego-STAN w/ avg.** Replaces slicing with token averaging across the T dimension (12).
- **Δ Ego-STAN w/ FMT.** Our main proposed method, which replaces averaging with FMT (2).

Tab. 3, Fig. 6, and Fig. 7 show the performance of each incremental model, illustrating each effect on the overall performance of Ego-STAN averaged across three random seeds. We observe the following.

Where we employ temporal attention matters. Temporal attention on the feature map sequence yields better performance than on the latent vector sequence arising from autoencoder structure (Temporal TFM vs. Ego-STAN variants). This demonstrates that 2D heatmap-based representations are adequate for HPE, and autoencoders may create unnecessary information bottlenecks.

Direct 3D regression works better than auto-encoder structure(s) indicating that as opposed to the conjecture in SOTA [48], the uncertainty information is effectively captured by the 2D heatmaps obviating the need for an autoencoder structure. Direct 3D regression can also be viewed as a variant of [58] without extra information about the camera parameters. We hypothesize that replacing the

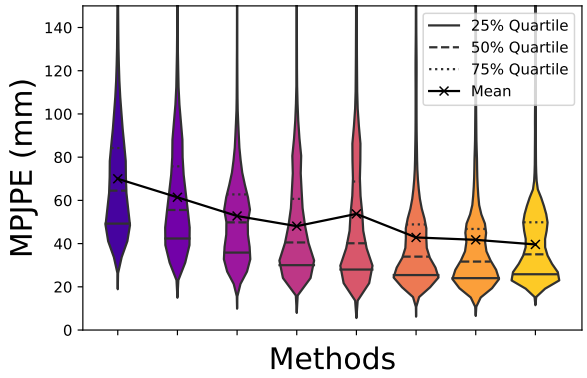


Figure 6: Overall MPJPE analysis across different methods. The violin plots demonstrate the contribution of each component of Ego-STAN for seed 42. Ego-STAN and its variants exhibit superior mean and variance properties (colors correspond to legend shown in Tab. 3).

autoencoder structure may also be the primary source of improvements reported in [58] for the static case. This is encouraging since camera information may be impractical to obtain in the real world.

Spatio-temporal information is essential. From Tab. 3 we note a slight performance dip (increased MPJPE) when only spatial attention is used. This indicates that for a static setting, using raw feature maps is better than using spatial attention. Moreover, incorporating a temporal aspect significantly improves the performance, underscoring its role in Ego-STAN variants. Further improvement due to FMT demonstrates that how we choose to aggregate information from feature maps matters.

Reducing trainable parameters. Ego-STAN variants lead to a **reduction of 31M (22%) trainable parameters as compared to the SOTA [48]**. This is attributed to our hybrid architecture which a) replaces the auto-encoder with direct 3D regression module (-28%), and b) leverages a FMT-based Transformer encoder-only module (+6%) obviating the need for a decoder [35]. These findings are in line with recent works which show improvements with CNN-Transformer hybrids [34].

Consistent and accurate HPE. Finally, in Fig. 6 we observe that as we progress to the right, in addition to the reduction in the overall MPJPE, the error distribution becomes lower and more consistent, indicating better variance properties (shorter vertically and wider at the bottom). This robustness can also be attributed to our ℓ_1-based 3D-loss (16).

Overall, our results demonstrate that Ego-STAN effectively handles distortions and self-occlusions.

5 DISCUSSION

Summary. Ego-HPE is challenging due to self-occlusions and distorted views. These challenges are unique to ego-HPE and the limited applicability of outside-in methods highlight the importance of developing ego-HPE specific solutions. To address these challenges, we design a domain-guided spatio-temporal hybrid architecture which leverages CNNs and Transformers using learnable parameters (FMT) that accomplish spatio-temporal attention, significantly reducing the errors caused by self-occlusion, especially in joints which suffer from high error in SOTA works. Our proposed model(s) – Ego-STAN – accomplishes consistent and accurate ego-HPE and HPE in general, while notably reducing the number of trainable parameters, making it suitable for cutting-edge full body motion tracking applications such as activity recognition, surgical training and immersive xR applications. This resulting transformer makes foundational contributions to spatio-temporal data analysis, impacting advances in ego-pose estimation and beyond.

Limitations, and future work. Although Ego-STAN demonstrates generalization capabilities on outside-in HPE datasets, there are no real-world ego-HPE sequential datasets. And while such datasets are developed, our future efforts will focus on developing transfer learning-based models which can work under domain shifts and variations in camera positions. This will lead to robust HPE models which can adapt to a variety of environments for real world critical applications.

Ethics statement. Human pose estimation applications include surveillance by public or private entities, which raises privacy invasion and human rights concerns. There is a need to educate practitioners and the users of applications relying on such technologies about such potential risks. Research on privacy preserving machine learning offers a way to mitigate these risks. Simultaneously, there is also a need to provide more legal protections for users and their data, and regulations for entities utilizing this data.

Table 3: Overview of ablations. From top to bottom, + and Δ denote cumulative and change via replacement, respectively. MPJPE for each model is reported with sample standard deviation from 3 different seeds.

Legend	Method	Parameters (Millions)	MPJPE (mm)
	Baseline (Tome et. al. [48])	141	65.7 ± 4.0
	+ ℓ_1-norm	141	60.1 ± 3.1
	+ Temporal TFM	141	55.7 ± 2.7
	Δ Direct 3D Regression	101	50.8 ± 1.7
	+ Spatial-only TFM	109	52.5 ± 3.0
	+ Ego-STAN w/ Slice	110	42.8 ± 0.0
	Δ Ego-STAN w/ Avg.	110	42.1 ± 2.1
	Δ Ego-STAN w/ FMT	110	40.4 ± 0.1

ACKNOWLEDGMENTS

This research was enabled by support from Mobility and AI lab (Nissan Motor Corp, Japan), Paul Fieguth and Sirisha Rambhatla's compute resource allocation award (2022-24) from the Digital Research Alliance of Canada (alliancecan.ca), and Sirisha Rambhatla's NSERC Discovery Grant (2022). Opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect sponsor's views.

REFERENCES

[1] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. 2021. A spatio-temporal transformer for 3D human motion prediction. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 565–574.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2014. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095* 2, 3 (2021), 4.

[4] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. 2019. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2272–2281.

[5] Michael Carroll, Ethan Osborne, and Caglar Yildirim. 2019. Effects of VR gaming and game genre on player experience. In *2019 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 1–6.

[6] Ching-Hang Chen and Deva Ramanan. 2017. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7035–7043.

[7] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. 2021. Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 1 (2021), 198–209.

[8] Yu Cheng, Bo Yang, Bo Wang, and Robby T Tan. 2020. 3d human pose estimation using spatio-temporal networks with explicit occlusion training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10631–10638.

[9] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T Tan. 2019. Occlusion-aware networks for 3d human pose estimation in video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 723–732.

[10] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. 2018. Learning 3d human pose from structure and motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 668–683.

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

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

[13] Hongyang Du, Dusit Niyato, Jiawen Kang, Dong In Kim, and Chunyan Miao. 2021. Optimal Targeted Advertising Strategy For Secure Wireless Edge Metaverse. *arXiv preprint arXiv:2111.00511* (2021).

[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 9)*, Yee Whye Teh and Mike Titterington (Eds.). PMLR, Chia Laguna Resort, Sardinia, Italy, 249–256. https://proceedings.mlr.press/v9/glorot10a.html

[15] Benzar Glen Grepon and Aldwin Lester Martinez. 2021. Architectural Visualization Using Virtual Reality: A User Experience in Simulating Buildings of a Community College in Bukidnon, Philippines. *arXiv preprint arXiv:2103.06238* (2021).

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

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

[19] Wen-Tsung Hsieh and Shao-Yi Chien. 2021. Learning to Perceive: Perceptual Resolution Enhancement for VR Display with Efficient Neural Network Processing. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 133–138. https://doi.org/10.1109/ISMAR-Adjunct54149.2021.00036

[20] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2014. 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 (jul 2014), 1325–1339.

[21] Katarzyna Janocha and Wojciech Marian Czarnecki. 2017. On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659* (2017).

[22] David C. Jeong, Jackie Jingyi Xu, and Lynn C. Miller. 2020. Inverse Kinematics and Temporal Convolutional Networks for Sequential Pose Analysis in VR. In *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. 274–281. https://doi.org/10.1109/AIVR50618.2020.00056

[23] Kyung-Min Jin, Gun-Hee Lee, and Seong-Whan Lee. 2022. OTPose: Occlusion-Aware Transformer for Pose Estimation in Sparsely-Labeled Videos. *arXiv preprint arXiv:2207.09725* (2022).

[24] Sam Johnson and Mark Everingham. 2010. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *Proceedings of the British Machine Vision Conference*. doi:10.5244/C.24.12.

[25] Tamara G. Kolda and Brett W. Bader. 2009. Tensor Decompositions and Applications. *SIAM Rev.* 51, 3 (aug 2009), 455–500. https://doi.org/10.1137/07070111X

[26] Valentyna Kovalenko, Maiia Marienko, and Alisa Sukhikh. 2022. Use of augmented and virtual reality tools in a general secondary education institution in the context of blended learning. *arXiv preprint arXiv:2201.07003* (2022).

[27] Chen Li and Gim Hee Lee. 2019. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9887–9895.

[28] Sijin Li, Weichen Zhang, and Antoni B Chan. 2015. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*. 2848–2856.

[29] Qihao Liu, Yi Zhang, Song Bai, and Alan Yuille. 2022. Explicit Occlusion Reasoning for Multi-person 3D Human Pose Estimation. In *European Conference on Computer Vision*. Springer, 497–517.

[30] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. 2020. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5064–5073.

[31] Zhenguang Liu, Haoming Chen, Runyang Feng, Shuang Wu, Shouling Ji, Bailin Yang, and Xun Wang. 2021. Deep dual consecutive network for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 525–534.

[32] Stoyan Maleshkov and Dimo Chotrov. 2013. Post-processing of engineering analysis results for visualization in VR systems. *arXiv preprint arXiv:1308.5847* (2013).

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

[34] Sachin Mehta and Mohammad Rastegari. 2021. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178* (2021).

[35] Jianbiao Mei, Mengmeng Wang, Yeneng Lin, Yi Yuan, and Yong Liu. 2021. Transvos: Video object segmentation with transformers. *arXiv preprint arXiv:2106.00588* (2021).

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

[37] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. 2017. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7025–7034.

[38] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 2019. 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*. 7753–7762.

[39] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. 2021. Spatial temporal transformer network for skeleton-based action recognition. In *International Conference on Pattern Recognition*. Springer, 694–701.

[40] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. 2017. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3433–3441.

[41] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. 2019. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE transactions on pattern analysis and machine intelligence* 42, 5 (2019), 1146–1161.

[42] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5693–5703.

[43] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. 2017. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*. 2602–2611.

[44] Markku Suomalainen, Alexandra Q Nilles, and Steven M LaValle. 2020. Virtual reality for robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11458–11465.

[45] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. 2016. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180* (2016).

[46] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. 2016. Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 991–1000.

[47] Denis Tome, Thiemo Alldieck, Patrick Peluse, Gerard Pons-Moll, Lourdes Agapito, Hernan Badino, and Fernando De la Torre. 2020. Selfpose: 3d egocentric pose estimation from a headset mounted camera. *arXiv preprint arXiv:2011.01519* (2020).

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

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[50] Neil Vaughan and Bogdan Gabrys. 2020. Scoring and assessment in medical VR training simulators with dynamic time series classification. *Engineering Applications of Artificial Intelligence* 94 (2020), 103760.

[51] Jian Wang, Lingjie Liu, Weipeng Xu, Kripasindhu Sarkar, and Christian Theobalt. 2021. Estimating egocentric 3d human pose in global space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11500–11509.

[52] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2020. Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision*. Springer, 764–780.

[53] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. 2020. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908* (2020).

[54] Weipeng Xu, Avishek Chatterjee, Michael Zollhoefer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. 2019. Mo 2 cap 2: Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE transactions on visualization and computer graphics* 25, 5 (2019), 2093–2101.

[55] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. 2021. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10448–10457.

[56] Keming Zeng and Guoyuan Cao. 2021. Application of VR Technology in Museum Narrative Design with Computer Vision Models. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. 913–916. https://doi.org/10.1109/ICCMC51019.2021.9418483

[57] Jinlu Zhang, Zhigang Tu, Jianyu Yang, Yujin Chen, and Junsong Yuan. 2022. 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*. 13232–13242.

[58] Yahui Zhang, Shaodi You, and Theo Gevers. 2021. Automatic Calibration of the Fisheye Camera for Egocentric 3D Human Pose Estimation from a Single Image. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1771–1780. https://doi.org/10.1109/WACV48630.2021.00181

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

[60] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. 2019. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2344–2353.

[61] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. 2016. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4966–4975.

[62] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. 2018. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE transactions on pattern analysis and machine intelligence* 41, 4 (2018), 901–914.

[63] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. 2022. MotionBERT: Unified Pretraining for Human Motion Analysis. *arXiv preprint arXiv:2210.06551* (2022).

A APPENDIX

A.1 Training and reproducibility details.

The implementation is done with PyTorch Lightning with three random seeds 22, 42, and 102.

Data Augmentation. For data augmentations, we first crop each image between index 180 and 1120 on the x-axis to remove the dark background that is not needed. Then we resize each image to 368×368 resolution. We attempted 9 distinct combinations of sequence length and sampling rate (number of frames skipped) to identify and utilize the best one. As illustrated in Tab. 5, there was no consistent pattern found in the experiments that displayed a trend favoring a certain number of frames or sequence length. Our chosen model had a sequence length of 5 and skip rate of 5.

Learning parameters. AdamW with base learning rate of $1e^{-4}$ and weight decay of0.01 is chosen as the optimizer for stable Transformer training.

Pre-training. Pre-trained ResNet-101 weights from ImageNet1K are loaded for initialization. The remaining modules are initialized with Xavier initialization [14]. The first 100K iterations are only trained on $\mathcal{L}_{2D}$ while using linear warmup on the learning rate so that LR @ 100K = $1e^{-4}$. After 100K iterations, the whole model is trained with the objective function as the sum of the 2D and 3D loss. We train our model with a maximum of 10 epochs with an early stopping patience of 7 on the validation MPJPE.

Compute Infrastructure. A batch size of 16 is fed to a single NVIDIA A100 GPU for accelerated training with AMD Milan 7413 CPU available via the shared high performance computing infrastructure.

Hyperparameters. The Transformer encoder in the spatio-temporal Transformer module has the following hyperparameters: hidden dimension of 512, depth of 3, 8 heads, MLP dimension of 1024, head dimension of 64, and 0.4 dropout. Deconvolution block in the heatmap reconstruction module is comprised of 2 deconvolution layers with kernel size = 3 and stride = 2 where the channels decrease from 2048 to 1024 and then from 1024 to 15. In the 3D pose estimator module, convolution block has 3 layers of 2D convolution layers with kernel side = 4 and stride = 2. The channels increase from 15 to 64, 64 to 128, and finally from 128 to 512. The linear block that follows the convolution block decreases the flattened features from the convolution block into the following dimensions: 18432, 2048, 512, and 48. All the layers in the 3D pose estimator and the heatmap reconstruction module have PReLU [16] as an activation function. $\lambda_\theta = -10^{-2}$ and $\lambda_L = 0.5$ were used as weights for the 3D loss function in (16).

A.2 Experimental Details

A.2.1 Data Requirements. Note that Ego-STAN does not require any additional labeling than those required by other pose estimation methods that leverage motion capture systems (whether ego-centric or not). Specifically, there is no special labeling required for the occluded joints since the subjects wear trackers for 3D pose coordinates, while the 3D coordinates to 2D image mapping is accomplished using camera intrinsics. In other words, any appropriate motion capture data can be used to render ego-centric views to generate training data for ego-pose estimation, which makes our model and training data flexible and versatile.

Table 4: Comparison of MACs. Our proposed method Ego-STAN is compared against a popular outline-in pose estimation method [33] and the SOTA egocentric pose estimation work [48]. Since Ego-STAN uses T of 5, it is expected that Ego-STAN has roughly $\times 5$ the MACs to the other two static models.

Approach	MACs (G)
Martinez et. al. [33]	32.1
Tome et. al. [48]	31.7
Ego-STAN (Ours) T=1	38.4
Ego-STAN (Ours) T=5	165.0

Table 5: Ego-STAN FMT overall MPJPE based on sequence length and number of frames skipped

MPJPE (mm)		Sequence Length 3	5	7
Frames Skipped	3	39.1	47.1	40.8
	5	39.5	40.4	40.1
	7	45.2	46.7	39.4

Table 6: Quantitative evaluation on Mo^{2}Cap^2 dataset. Ego-STAN outperforms the SOTA [48] demonstrating its ability to generalize to real-world sequential views despite being trained on static views (no temporal component), also highlighting the leverage provided by FMT. PA-MPJPE refers to procrustes aligned-MPJPE; details in A.2.3.

Approach	Error (PA-MPJPE) mm
Tome et al. [48]	114.1
Ego-STAN FMT (Ours)	**102.4**

A.2.2 Experiments on Human3.6M Dataset. The Human3.6M Dataset [20] is one of the largest and most popular benchmarks for 3D Human Pose Estimation owing to its impressive arsenal of real-world images with individuals performing a variety of activities in motion capture lab setting, which renders it practical for single-person 3D HPE tasks. The images of this data-set are captured from an outside-in viewpoint with frames present from 4 different camera perspectives, thus enriching its viewpoint diversity. There are two popular protocols when evaluating methods on the Human3.6M Dataset, with Protocol 1 training on subjects (S1, S5, S6, S7, S8) and testing on subjects (S9, S11), whereas Protocol 2 trains on (S1, S5, S6, S7, S8, S9) and tests on (S11) using procrustes-aligned poses. For the 3D HPE, we evaluate on both protocols while sampling every 16 frames for training without any data augmentations. Seed 42 was used for this experiment. For the 2D HPE, protocol 2 was used

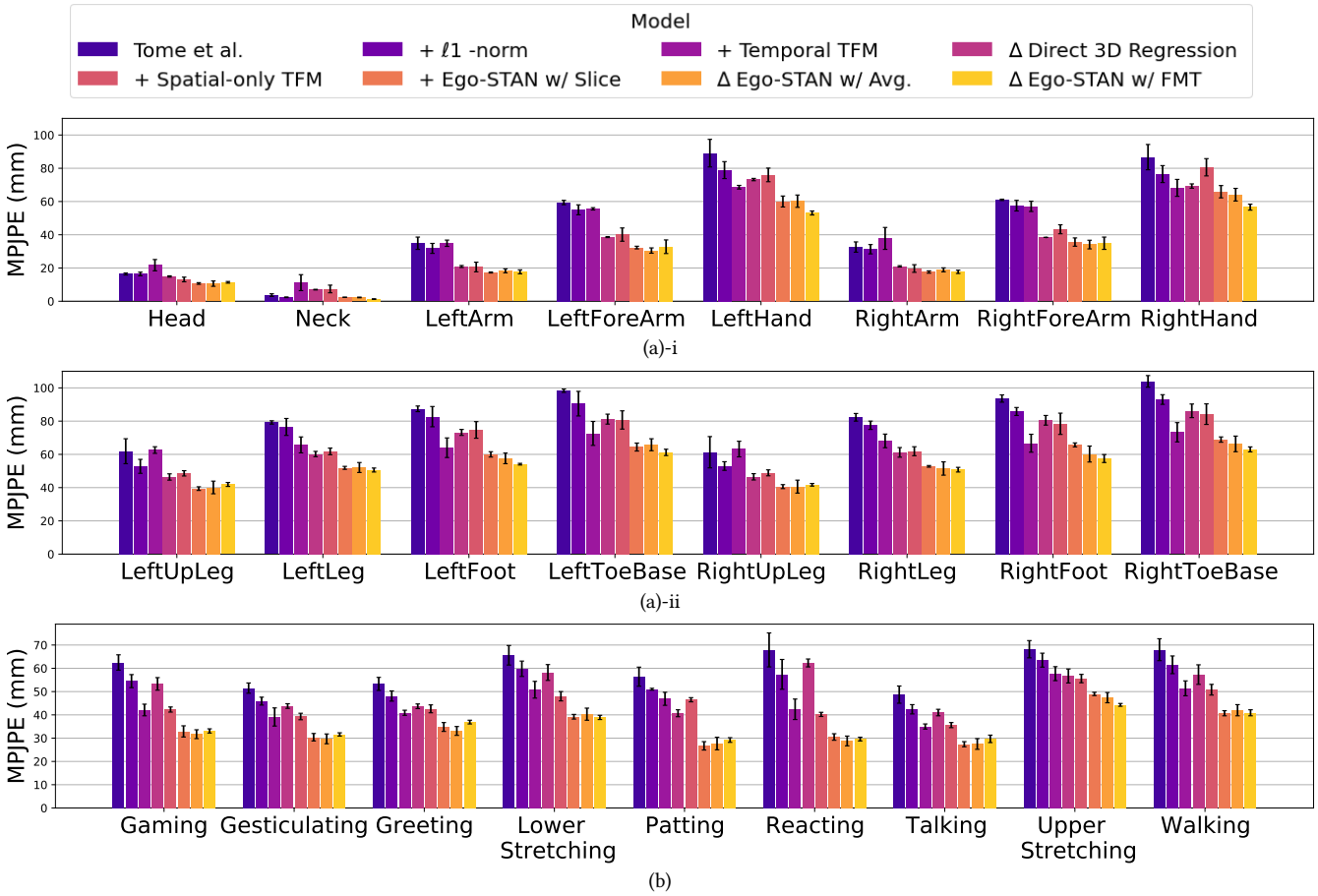


Figure 7: Per-Joint and Per-Action MPJPE Bar Plot. As compared to the reproduced SOTA baseline [48], Ego-STAN has significant improvements over heavily occluded joints (farthest from the camera), and challenging actions (upper stretching and lower stretching). While the other seven actions are very close between the Ego-STAN variants, Ego-STAN FMT exhibits superior performance. The results for the 8 incremental models in (a)-i presents MPJPE for upper body joints, (a)-ii for lower body joints, and (b) for actions.

for train/test split while sampling every 16 frames. Similarly, no data augmentations were used for each approach. Average of seeds 42, 22, and 102 was reported.

A.2.3 Experiments on Mo2Cap2 Dataset. The Mo2Cap2 dataset was one of the first large HPE synthetic datasets with a cap-mounted fish-eye egocentric camera [54]. The dataset consists of static images, and is not amenable for spatio-temporal modeling. While a pioneer in the corpus of ego-centric data-sets, its limiting factors include the quality of the synthetically generated images. Their evaluation set on the other hand, is composed of two videos,supplemented with 3D pose labels, captured from an ego-centric viewpoint for both in-door and out-door motion capture settings.

Since the pre-computed heatmaps that [48] use for 2D to 3D estimation are not publicly available and the main goal of Ego-STAN is to create accurate feature maps, we setup out training pipeline similar to [54]. We first train the image-to-2D heatmap module on the MPII [2] and LSP [24] dataset. Then, we reduce the learning rate by a factor of 50 to the first 86% of the layers in resnet. The image-to-2D module is trained for 50k iterations following by a 70k training iteration of 2D-to-3D module while the image-to-2D module is frozen. Seed 42 was used for this experiment.

A.3 Multiply-Accumulate comparison

Multiply-accumulate (MAC) is measure to count the number of operations in a model. Tab. 4 compares the MACs between a popular outline-in pose estimation work [33], SOTA egocentric pose estimation work [48] and Ego-STAN. The FLOPS will naturally increase since CNN will compute T many times for T steps and the addition of a transformer network will increase the computations. However, as demonstrated in Tab. 3, the number of parameters decrease with the introduction of direct regression and the weight-sharing of Resnet.

A.4 Learnable Positional Embedding

Detailed information on learnable position embeddings can be found in [11, 12].