A Appendix

A.1 Supplementary Results

Ablation studies raw results. We present the raw table form of the bar graph shown in Fig. 7. Upper body per-joint MPJPE from Fig. 7 (a)-i is shown in Tab. 4 and the lower body (a)-ii is shown in Tab. 5. Per-action MPJPE from Fig. 7 (b) and the standard deviation for our proposed work in Tab. 1 are reported in Tab. 6. Mean MPJPE over 3 random seeds with standard deviation is reported.

Table 4: **Ablation Studies Per-Joint Error for Upper Body Joints.** Refers to Fig. 7 (a)-i.

Approach	Head	Neck	Left arm	Left fore arm	Left hand	Right arm	Right fore arm	Right hand
Tome et al. [21]	16.4 ± 0.6	3.7 ± 0.8	34.9 ± 3.7	59.4 ± 1.3	89.1 ± 8.3	32.6 ± 3.1	61.0 ± 0.4	86.7 ± 7.6
Tome et al. [21] with ℓ_1-norm	16.4 ± 1.1	2.4 ± 0.2	31.8 ± 3.0	55.0 ± 3.0	78.9 ± 5.1	31.3 ± 2.8	57.5 ± 3.2	76.5 ± 5.1
Temporal TFM	21.7 ± 3.4	11.2 ± 4.8	34.9 ± 1.9	55.6 ± 0.7	68.6 ± 1.1	37.8 ± 6.6	57.1 ± 3.1	68.2 ± 5.1
Direct 3D reg.	14.9 ± **0.4**	7.0 ± 0.2	20.9 ± 0.6	38.6 ± **0.3**	73.2 ± **0.7**	21.0 ± **0.4**	38.4 ± **0.0**	69.3 ± **1.3**
Spatial-only TFM	13.1 ± 1.4	7.4 ± 2.3	20.6 ± 2.9	40.1 ± 4.0	76.0 ± 4.1	19.7 ± 2.3	43.3 ± 2.7	80.6 ± 5.2
Ego-STAN Avg (Ours)	**10.6** ± 1.6	2.3 ± 0.2	18.3 ± 1.2	**30.5** ± 1.6	60.2 ± 3.7	18.9 ± 1.2	**34.1** ± 2.6	64.1 ± 3.8
Ego-STAN Slice (Ours)	10.7 ± 0.5	2.4 ± **0.1**	**17.3** ± **0.3**	32.3 ± 0.8	60.0 ± 3.3	**17.6** ± 0.7	35.6 ± 2.5	65.9 ± 3.7
Ego-STAN FMT (Ours)	11.4 ± 0.5	**1.3** ± 0.2	17.7 ± 1.2	32.8 ± 4.1	**53.1** ± 1.2	17.7 ± 1.0	34.9 ± 3.7	**56.6** ± 1.8

Table 5: **Ablation Studies Per-Joint Error for Lower Body Joints.** Refers to Fig. 7 (a)-ii.

Approach	Left up leg	Left leg	Left foot	Left toe base	Right up leg	Right leg	Right foot	Right toe base
Tome et al. [21]	61.9 ± 7.5	79.2 ± 1.0	87.4 ± 1.7	98.3 ± 1.0	61.3 ± 9.4	82.3 ± 2.3	93.6 ± 2.2	103.9 ± 3.5
Tome et al. [21] with ℓ_1-norm	52.8 ± 4.2	76.5 ± 5.1	82.7 ± 6.1	90.5 ± 7.4	53.0 ± 2.6	77.5 ± 2.5	85.8 ± 2.4	93.0 ± 2.9
Temporal TFM	62.6 ± 2.0	65.7 ± 4.8	64.0 ± 5.9	72.6 ± 7.2	63.2 ± 4.7	68.1 ± 4.1	66.7 ± 5.4	73.3 ± 5.8
Direct 3D reg.	46.4 ± 1.9	60.2 ± 1.6	73.2 ± 1.8	81.2 ± 3.0	46.5 ± 1.9	61.2 ± 2.9	80.5 ± 2.9	86.2 ± 4.1
Spatial-only TFM	48.6 ± 1.6	61.8 ± 2.0	74.7 ± 5.0	80.7 ± 5.5	48.9 ± 1.8	61.9 ± 2.7	78.4 ± 6.4	84.2 ± 6.2
Ego-STAN Avg (Ours)	40.1 ± 3.8	52.1 ± 3.0	57.6 ± 3.2	65.8 ± 3.6	**40.6** ± 3.9	51.5 ± 4.0	60.2 ± 4.8	66.3 ± 4.7
Ego-STAN Slice (Ours)	**39.4** ± **1.1**	51.9 ± **0.9**	60.0 ± 1.6	64.4 ± 2.4	**40.6** ± 1.2	52.8 ± **0.5**	65.7 ± **1.2**	68.9 ± 1.6
Ego-STAN FMT (Ours)	41.9 ± 1.2	**50.6** ± 1.2	**54.1** ± **0.5**	**61.2** ± **1.9**	41.7 ± **0.8**	**50.8** ± 1.4	**57.5** ± 2.4	**63.0** ± **1.5**

Video qualitative analysis. The video files showing further qualitative comparisons (such as those shown in Fig. 4) are attached in the supplementary materials. Here, we present randomly sampled actions for the testset for a fair comparison with the dual-branch baseline [21].

A.2 Experimental Details

A.2.1 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

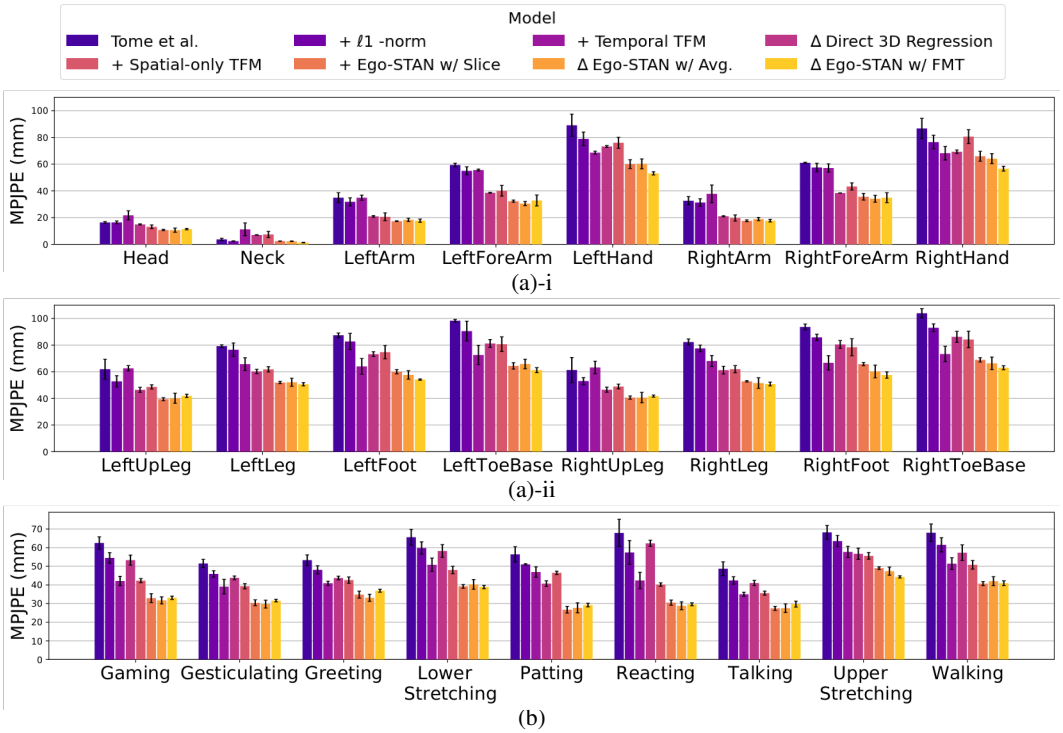

Figure 7: **Per-Joint and Per-Action MPJPE Bar Plot**. As compared to the reproduced SOTA baseline [21], Ego-STAN has significant improvements over heavily occluded joints (farthest from the camera), and challenging actions (upper stretching and lower stretching). While the other seven actions are very 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.

Table 6: **Ablation Studies Per-Action Error.** Refers to Fig. 7 (b).

Approach	Game.	Gest.	Greet.	Lower Stretch.	Pat.	React.	Talk.	Upper Stretch.	Walk.
Tome et al. [21]	62.5 ± 3.3	51.5 ± 2.2	53.3 ± 2.8	65.6 ± 4.2	56.4 ± 4.1	67.9 ± 7.3	48.7 ± 3.7	68.2 ± 3.7	68.0 ± 4.7
Tome et al. [21] with ℓ_1-norm	54.5 ± 2.8	45.9 ± 1.7	48.1 ± 2.2	59.8 ± 3.3	51.0 ± 0.4	57.4 ± 6.4	42.4 ± 2.0	63.5 ± 3.0	61.5 ± 3.8
Temporal TFM	53.3 ± 2.7	43.8 ± 1.0	43.7 ± 1.0	58.2 ± 3.4	40.7 ± 1.5	62.3 ± 1.7	41.0 ± 1.4	56.7 ± 3.0	57.3 ± 4.2
Direct 3D reg.	42.3 ± 1.1	39.3 ± 1.4	42.6 ± 1.7	48.0 ± 2.0	$46.5 \pm$ **0.9**	40.2 ± 0.9	$35.6 \pm$ **1.1**	55.6 ± 1.8	50.8 ± 2.3
Spatial-only TFM	42.1 ± 2.5	39.1 ± 3.9	40.9 ± 1.1	50.8 ± 3.6	46.9 ± 2.8	42.4 ± 4.4	$35.0 \pm$ **1.1**	57.7 ± 3.0	51.4 ± 3.2
Ego-STAN Avg (Ours)	**31.7** ± 1.9	**29.7** ± 2.1	**33.1** ± 1.9	40.3 ± 2.6	27.7 ± 2.7	**28.8** ± 2.1	27.5 ± 2.3	47.4 ± 2.2	42.0 ± 2.4
Ego-STAN Slice (Ours)	32.9 ± 2.4	30.4 ± 1.6	34.8 ± 1.9	39.2 ± 1.0	**26.7** ± 1.8	30.5 ± 1.4	**27.4** $\pm$ **1.1**	49.1 ± 0.7	**40.7** $\pm$ **1.1**
Ego-STAN FMT (Ours)	$33.1 \pm$ **0.9**	$31.6 \pm$ **0.7**	$36.9 \pm$ **0.8**	**38.9** $\pm$ **0.9**	29.2 ± 1.0	$29.6 \pm$ **0.8**	29.7 ± 1.6	**44.3** $\pm$ **0.6**	40.9 ± 1.3

Table 7: Quantitative evaluation on Mo^{2}Cap^2 dataset. Ego-STAN outperforms the SOTA [21] 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. [21]	114.1
Ego-STAN FMT (Ours)	**102.4**

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 [21].

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.

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

Table 8: Standard deviation of Tab. 2 across 3 seeds (22, 42, and 102); here Sld: Shoulder, Elb: Elbow.

Approach (STD)	Sld	Elb	Wrist	Hip	Knee	Ankle	Spine	All
Sun [36]	0.104	0.070	0.061	0.039	0.021	0.023	0.032	0.026
Sun [36] + **Ego-STAN**	0.018	0.022	0.027	0.019	0.009	0.006	0.013	0.006

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 $\widehat{\boldsymbol{C}} \in \mathbb{R}^{J\times 2}$ and label coordinates $\boldsymbol{C} \in \mathbb{R}^{J\times 2}$, 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:

$$PCK^{(n)} = \frac{1}{J}\sum_{j=1}^{J} \mathbf{1}_{||\widehat{\mathrm{C}}_{\mathrm{j}}^{(\mathrm{n})} - \mathrm{C}_{\mathrm{j}}^{(\mathrm{n})}||_2 \,<\, \mathrm{m}} \tag{16}$$

Here the x and y coordinates of $\widehat{\boldsymbol{C}}$ and $\boldsymbol{C}$ 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:

$$PCK_{total} = \frac{1}{N}\sum_{n=1}^{N} PCK^{(n)} \tag{17}$$

A.2.2 Experiments on Human3.6M Dataset

The Human3.6M Dataset [16] 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 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.

Table 9: **Per-Joint Error for Upper Body Joints on Human3.6M.** P1 tests on subject (S9, S11) with MPJPE. P2 tests on subject (S11) using procrustes-aligned MPJPE.

Approach	Head	Neck	Left shoulder	Left elbow	Left wrist	Right shoulder	Right elbow	Right wrist
P1 Tome et al. [21]	146.5	**131.5**	129.4	170.4	205.8	133.9	175.3	212.4
P1 Ego-STAN FMT (Ours)	**145.7**	132.4	**120.5**	**161.4**	**196.5**	**124.5**	**165.5**	**200.2**
P2 Tome et al. [21]	53.9	45.7	54.3	115.9	136.4	47.7	111.0	132.6
P2 Ego-STAN FMT (Ours)	**48.1**	**42.3**	**42.3**	**95.8**	**126.7**	**38.9**	**92.2**	**129.7**

Table 10: **Per-Joint Error for Lower Body Joints on Human3.6M.** Same protocols as Tab. 9.

Approach	Left hip	Left Knee	Left foot	Right hip	Right knee	Right foot	Thorax	Spine
P1 Tome et al. [21]	41.3	125.7	170.3	42.6	126.0	187.3	118.1	74.2
P1 Ego-STAN FMT (Ours)	**30.4**	**95.5**	**134.1**	**30.3**	**96.0**	**127.1**	**113.7**	**63.5**
P2 Tome et al. [21]	**69.7**	87.1	110.0	**65.9**	81.5	107.4	38.2	**43.4**
P2 Ego-STAN FMT (Ours)	74.3	**71.1**	**92.2**	77.6	**68.8**	**86.5**	**34.0**	44.8

Table 11: 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

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 [22]. 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 [21] 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 [22]. We first train the image-to-2D heatmap module on the MPII [58] and LSP [59] 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.2.4 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. 11, 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 [60]. 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 [61] as an activation function. $\lambda_\theta = -10^{-2}$ and $\lambda_L = 0.5$ were used as weights for the 3D loss function in (15).

A.3 Intuitive explanation on Feature Map Token.

We will summarize Sec. 3.1 and Sec. 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.

A.4 Multiply-Accumulate comparison

Multiply-accumulate (MAC) is measure to count the number of operations in a model. Tab. 12 compares the MACs between a popular outline-in pose estimation work [19], SOTA egocentric pose estimation work [21] 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.

Table 12: **Comparison of MACs.** Our proposed method Ego-STAN is compared against a popular outline-in pose estimation method [19] and the SOTA egocentric pose estimation work [21]. 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. [19]	32.1
Tome et. al. [21]	31.7
Ego-STAN (Ours) T=1	38.4
Ego-STAN (Ours) T=5	165.0

A.5 LEARNABLE POSITIONAL EMBEDDING

Detailed information on learnable position embeddings can be found in [47, 55].

B NOTATION

Numbers and Arrays

$\boldsymbol{A}$	A matrix
A	A tensor

Sets and Graphs

$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
$\{0, 1, \dots, n\}$	The set of all integers between 0 and n

Indexing

$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{-i,:}$	Row i from the bottom of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$

Functions

$\|\|\boldsymbol{x}\|\|_p$	L^p norm of $\boldsymbol{x}$
$\|\|\boldsymbol{x}\|\|$	L^2 norm of $\boldsymbol{x}$
$\mathbf{1}_{\text{condition}}$	is 1 if the condition is true, 0 otherwise