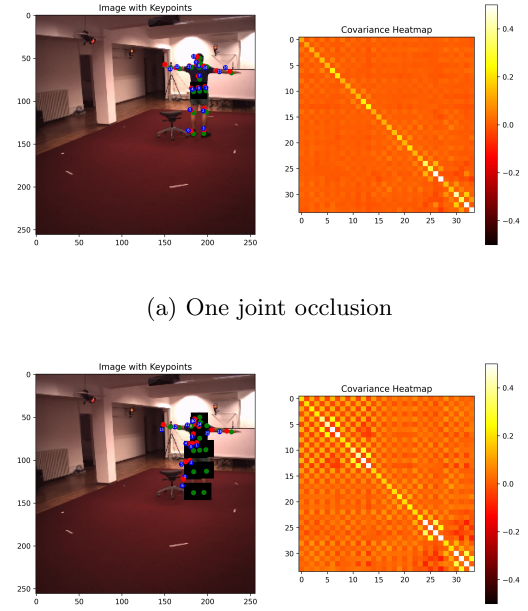
Effect of probabilistic pose estimator on downstream tasks

Valentin Mehdi Mathieu Perret
valentin.perret@epfl.ch

Abstract

Downstream tasks following human keypoints prediction are sensitive to the input data. Such tasks include 2D-3D keypoint lifting, action recognition. Probabilistic pose estimators, with recent work Shukla et al. (2024) show the possibility of evaluating uncertainty by predicting a covariance matrix and a mean vector of the prediction. In this project we aim at proving the potential of such model to make downstream task more robust. For this we present the preliminary work done on building a framework to evaluate the effect of probabilistic human pose estimator. This project builds on the work of Shukla et al. (2024) by extending the code to work on COCO dataset and delves into 2D-3D human keypoint lifting and action recognition. The two GitHub repository can be found on: probabilistic pose and framework.



(a) One joint occlusion

(b) Nine joint occlusion

Figure 1: Figure comparing the effect of occlusion on the prediction and covariance matrix. Green is the ground truth, red is the mean position prediction and blue is the point sampled from the predicted mean and covariance.

1 Introduction

The choice of the 2D model predictor is a source of error for the future assignment. If one would choose model A trained on generated data by a 2D keypoint estimator model B and evaluate it on the data generated by a 2D keypoints estimator model C, we expect the results to be worse than if we evaluate it on the same data used for training.

In addition, adding constant noise to the different keypoints could be seen as sampling data from a Gaussian distribution with the covariance matrix $\Sigma = \epsilon I$, where I is the identity matrix. Each joint in this matrix has the same noise and doesn't impact the other joint's uncertainty. Having a full covariance matrix enables us to better model the uncertainty and the relation between the various joints. Therefore, we believe that sampling from such a distribution could be seen as a better augmentation method and reduce overfitting. Using a probabilistic pose estimator to generate the data used for future tasks would make the downstream model perform better to unseen data. On Fig1, the effects of the probabilistic pose estimator are shown. The covariance matrix demonstrates the uncertainty the model has to predict a certain joint.

This paper contains a part about the work done on the probabilistic estimator and the main part on the framework method and results.

Table 1: Comparison of the performance of various state of the art 2D-3D lifting models with the MPJPE evaluation metric on Protocol 1 and protocol 2 and the floating points operations per second required.

	P1/P2 (mm)	FLOPs (M)
STCFormer	40.5/31.8	156215
MotionBert	39.2/32.9	-
MotionAGFormer	**38.4/32.5**	155634
GraphMLP	43.8/34.9	**356**

2 Related work

This project's most related paper, *Evaluating Recent 2D Human Pose Estimators for 2D-3D Pose Lifting* Mehraban, Qin & Taati (2024), evaluates different 2D Human Pose estimators. It arrives at the conclusion to merge the 2D estimation for less noisy predictions.

Furthermore, we heavily used two 2D-3D lifting models it is worth mentioning them in this section: MotionAGFormer Mehraban, Adeli & Taati (2024) and GraphMLP Li et al. (2025). The first one is sequence to sequence, whereas the second model is a sequence to frame model. GraphMLP uses the provided ground-truth, however MotionAGFormer preprocesses the input by multiplying by a certain factor. This latter model, reports state of the art performances on the Human3.6M Ionescu et al. (2013). However, as we see on Table 1, GraphMLP is much quicker to train despite not obtaining the best results. Due to time constraint and the performance being close, we decided to continue our experiment with GraphMLP.

The second downstream task we used is action recognition. The model chosen for it is SkateFormer Do & Kim (2025). Its performance is evaluated on the NTU Shahroudy et al. (2016) dataset. One images can contain two humans on it, it contains 60 possible actions. We chose it because it reports state of the art results.

Finally, like said previously, this work builds on Shukla et al. (2024). The final aim is to use a probabilistic pose estimator.

3 Method implemented

The current model of TIC-TAC works for human pose estimation on the MPII dataset for humans with all joints present in the image, it contains 14 keypoints. Including a dataloader for the MS COCO dataset enables us, to further evaluate the TIC-TAC framework for human pose. This dataset contains more images with 17 keypoints than MPII, most of those images contain non present joints. Enabling the framework to work on image, where certain joints are not necessarily present would enhance the evaluation of the model.

To evaluate the effect of 2D human prediction on downstream tasks and see the benefit of probabilistic pose, we need 2D Human keypoint prediction models: ViTPose Xu et al. (2022) and Stacked Hourglass Xu & Takano (2021). Following Fig. 2, the models are used to predict 2D human keypoints. Those predictions are then used to train two models of GraphMLP. Finally, we evaluate both trained models on the data used for the other model's training and compare the results. Adding noise to the ground-truth and training on this data is used as a comparison.

Finally, we implement the probabilistic pose estimator in our framework. We sample from the computed mean vector and covariance matrix for our lifting model. We train a model on the sampled keypoints and evaluate it on the other 2D keypoints.

On top of this we chose action recognition as a second downstream task. The model used is SkateFormer. As a preliminary experience we first validate the performance, and then add noise to the keypoints prediction to see how the evaluation result evolves.

The framework is built around the Human 3.6M dataset. It provides various frames of videos, 2D keypoints and 3D keypoints and is the dataset used as benchmark for keypoints lifting. Individual humans are visible in all frames. We used 17 keypoints. We use this dataset for keypoints lifting but also action recognition.

The NTU dataset is the dataset used as benchmark for action recognition. SkateFormer performance is reported on this dataset.

The various experiments are evaluated with the following metrics: average precision (AP), mean per joints position error (MPJPE) under two different protocols P1 and P2, cross subject (X-Sub) and cross view (X-View) evaluation with joint modality only.

The experiments were run with one GPU. The probabilistic pose model is trained with 150 epochs on MPII and COCO, and 10 epochs on Human3.6M. The GraphMLP model is always trained with 20 epochs and SkateFormer is trained with 500 epochs. Finally, ViTPose and Hourglass were trained with 10 epochs.

4 Experiment

The first part of this project is on the human pose probabilistic estimator. After building a MS-COCO dataloader to work with the already existing code, we looked at ways to not necessarily have to keep images

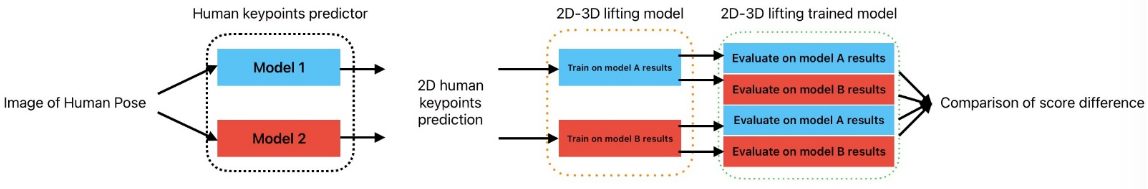

Human keypoints predictor
Image of Human Pose
Model 1
Model 2
2D human keypoints prediction
2D-3D lifting model
Train on model A results
Train on model B results
2D-3D lifting trained model
Evaluate on model A results
Evaluate on model B results
Evaluate on model A results
Evaluate on model B results
Comparison of score difference

Figure 2: Flow of the experiment, example with 2D-3D keypoints lifting.

Table 2: Number of single person used for the experiment from the different dataset configuration.

	Train	Test
MPII	10614	2416
MS-COCO all joints	8475	336
MS-COCO Bbox + 1gt	74762	3035

where all joints of the person are present. Therefore, we keep frames where at least one joint of a person is in the image (occluded or visible) and where the bounding box is big enough. The bounding box is defined as the min and max coordinates (in pixels of the images) of the present coordinates. This leaves with an increased dataset size see Table 2.

In addition we added a small neural network to predict if the joints is visible it works in parallel to the auxiliary network. This network outputs a Sigmoid binarized to 0 or 1 with a threshold of 0.5, to recognize if the joints are in the image or not. Knowing if the joint is present or not, enables us to drop the corresponding row in the mean vector and the corresponding row and column in the covariance matrix. This is done for the x and y coordinates.

Now about the framework, first we generate the 2D keypoints with the ViTPose and the Stacked-Hourglass (SH) models of the Human3.6M dataset. We obtained an average precision (AP) with threshold 50 of respectively: **0.9936** and **0.997**. If we apply the trained models on data coming from MS-COCO we observe the Human3.6M's poses on it, which shows that they overfit on the Human3.6M dataset.

Following this we train the model separately on ViTPose and Stacked-Hourglass 2D keypoints. We call the models ViT2-3D, SH2-3D for the models trained on the ViTPose and Hourglass respectively. We then evaluate both models on the 2D keypoints generated from both models. The results are in Table 3. As a reference when we train and evaluate with the ground-truth data we obtain a MPJPE of P1 = 81.86 mm and P2 = 44.70mm. From the results, we see that both models perform better when they use data they have been trained for. Furthermore, the result of both models is more influenced by the dataset used, than by the model used.

Finally, we add Gaussian Noise with 0 mean and a standard deviation of 0.001, 0.005, 0.01 and 0.02. We add this noise on the 2D ground-truth keypoints and train a model on it, which we then evaluate. Furthermore, we evaluate this model on the the ViTPose and Stacked Hourglass 2D keypoints. We can see the results on Table 4, adding noise during training does enable the model to obtain better results on the ViTPose and Stacked Hourglass keypoints for the P2 metric. With the right noise of 0.005 we obtain the best results; we even see an improvement of 3.2% on Stacked Hourglass generated keypoints.

We trained a model on the probabilistic pose estimator (ProbPose) data and then evaluated on the different 2D keypoints: the ground-truth (GT), the ViTPose and Stacked Hourglass (SH) generated 2D keypoints. The results can be found in Table 5. We see that the best results are obtained with the ground truth data for evaluation. The other data sources for evaluation all obtain a worse result than if the data was used for their own training, or one of the other data set. Generally, from all our experiments when training a model, the evaluation set mostly influences the results. As the training set is never far from the ground truth the model learns to predict the precision for the different data sets. Even though, the probabilistic pose estimator acts as a data augmentation, it did not improve performance. Probably it didn't have time to converge and come as close to the ground truth as what we had with the ViTPose and Stacked Hourglass models.

The 2nd downstream task was implemented in the framework. We adapted certain hyperparameters,

Table 3: Comparing ViT2-3D, SH2-3D on ViTPose and Stacked hourglass (SH) generated 2D coordinates, aswell as the ground-truth (GT) 2D keypoints. The metrics used is MPJPE with P1/P2 [mm].

2D-3D	ViTPose	SH	GT
ViT2-3D	**195.7/115.8**	143.5/89.8	103.7/59.1
SH2-3D	190.0/118.0	**140.2/89.8**	98.2/58.6

Table 4: Results on the various 2D keypoints datasets using the MPJPE metric on the evaluation set, by adding noise to the dataset set. The ground-truth column provides the results of the trained model with the amount of noise for comparison purposes. The other two columns are the results of the model, with the said 2D-keypoints as evaluation. The best result is in bold. The results are in mm. As a reference the smallest link has a length of 0.02 in our dataset.

Std	Ground-Truth [mm]	ViTPose [mm]	SH [mm]
0	P1 = 81.86, P2 = 44.7	P1 = 189.80, P2 = 115.97	P1 = 134.98, P2 = 86.09
0.001	P1 = 86.36, P2 = 46.27	P1 = 192.23, P2 = 117.31	P1 = 138.41, P2 = 86.65
0.005	P1 = 82.67, **P2 = 41.36**	P1 = 190.59, **P2 = 114.43**	P1 = 136.45, **P2 = 83.32**
0.01	P1 = 102.14, P2 = 51.42	P1 = 198.86, P2 = 117.67	P1 = 149.87, P2 = 88.10
0.02	P1 = 94.76, P2 = 51.15	P1 = 190.16, P2 = 114.71	P1 = 139.96, P2 = 85.75

Table 5: Results on the various 2D keypoints datasets using the MPJPE metric with both protocols, on the evaluation set. The training set samples from the probabilistic pose estimator mean and covariance matrix.

Metric	ProbPose	GT	ViTPose	SH
P1 [mm]	167.39	113.12	191.61	146.30
P2 [mm]	111.11	75.59	120.26	98.37

so that it could predict based on 2D keypoints. We used the name of the labels to name the different action labels of the frame sequences, to apply it to the Human3.6M dataset. However, the model reaches poor results with 0.13 on the X-Sub and 0.15 on the X-View metric. Our explanation of this phenomenon is that the label we chose are not the real labels, and that the dataset is not made for action recognition tasks.

Nevertheless we went back to the NTU dataset, and applied noise on the different 3D joint positions as a final experiment for this project. The applied noise is Gaussian noise with 0 mean and 0.01, 0.02, 0.05 and 0.1 of standard deviation. The results are respectively 0.92, 0.91, 0.91, 0.89, 0.86 with an accuracy with the X-Sub60 metric. Much like keypoints lifting we do not see a big drop in performance.

5 Further work

It is important to decide around which dataset to build the framework or at least if it is necessary to do so. Furthermore, the framework needs to be built around the NTU dataset, to make the same evaluation as for keypoints lifting. Having a second downstream task to fully evaluate the potential of the probabilistic pose estimator is a need.

6 Conclusion

Through this work, with keypoints lifting, we have seen the potential of probabilistic pose estimators for downstream tasks. Having those keypoints could make the downstream tasks models more robust to unseen data. We also saw the effect of noise on action recognition. Furthermore, we were able to incorporate images where all joints are not necessarily present in the probabilistic pose estimator and this extending his work potential.

7 Acknowledgements

I would like to take a moment to heartfully thank Megh Shukla and Reyhaneh Hosseininejad. Their insights and expertise where truly helpful, their opinion proved most valuable and enabled a non negligeable gain of knowledge. I would also like to extend my acknowledgement to Prof. Alahi for the opportunity to do my semester project in the VITA lab.

References

Do, J. & Kim, M. (2025), Skateformer: Skeletal-temporal transformer for human action recognition, *in* 'European Conference on Computer Vision', Springer, pp. 401–420.

Ionescu, C., Papava, D., Olaru, V. & Sminchisescu, C. (2013), 'Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments', *IEEE transactions on pattern analysis and machine intelligence* **36**(7), 1325–1339.

Li, W., Liu, M., Liu, H., Guo, T., Wang, T., Tang, H. & Sebe, N. (2025), 'Graphmlp: A graph mlp-like architecture for 3d human pose estimation', *Pattern Recognition* **158**, 110925.

Mehraban, S., Adeli, V. & Taati, B. (2024), Motionagformer: Enhancing 3d human pose estimation with a transformer-gcnformer network, *in* 'Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision', pp. 6920–6930.

Mehraban, S., Qin, Y. & Taati, B. (2024), Evaluating recent 2d human pose estimators for 2d-3d pose lifting, *in* '2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)', IEEE, pp. 1–5.

Shahroudy, A., Liu, J., Ng, T.-T. & Wang, G. (2016), Ntu rgb+ d: A large scale dataset for 3d human activity analysis, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1010–1019.

Shukla, M., Salzmann, M. & Alahi, A. (2024), Tictac: A framework for improved covariance estimation in deep heteroscedastic regression, *in* 'Proceedings of the 41st International Conference on Machine Learning (ICML) 2024'.

Xu, T. & Takano, W. (2021), Graph stacked hourglass networks for 3d human pose estimation, *in* 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition', pp. 16105–16114.

Xu, Y., Zhang, J., Zhang, Q. & Tao, D. (2022), 'Vitpose: Simple vision transformer baselines for human pose estimation', *Advances in Neural Information Processing Systems* **35**, 38571–38584.