공학석사학위논문

Refining 3D Hand Pose Estimation Using Masked Language Model

마스크 언어 모델을 이용한 3차원 손 좌표의 미세조정

2021 년 8 월

서울대학교 대학원

컴퓨터공학부

지 승 근

Refining 3D Hand Pose Estimation Using Masked Language Model

마스크 언어 모델을 이용한 3차원 손 좌표의 미세조정

지도교수 문 병 로
이 논문을 공학석사 학위논문으로 제출함

2021 년 6 월

서울대학교 대학원

컴퓨터공학부

지 승 근

지승근의 공학석사 학위논문을 인준함

2021 년 7 월

위원장	박 근 수
부위원장	문 병 로
위 원	이 재 욱

Abstract

Accurately estimating hand/body pose from a single viewpoint under occlusion is challenging for most of the current approaches. Recent approaches have tried to address the occlusion problem by collecting or synthesizing images having joint occlusions. However, the data-driven approaches failed to tackle the occlusion because they assumed that joints are independent or they only used explicit joint connection.

To mitigate this problem, I propose a method that learns joint relations and refines the occluded information based on their relation. Inspired by BERT in Natural Language Processing, I pre-train a refinement module and add it at the end of the proposed framework. Refinement improves not only the accuracy of occluded joints but also the accuracy of whole joints. In addition, instead of using a physical connection between joints, the proposed model learns their relation from the data. I visualized the learned joint relation in this paper, and it implies that assuming explicit connection hinders the model from accurately predicting joint locations accurately.

keywords: Computer Vision, 3D Hand Pose Estimation, Neural Language Processing

student number: 2019-23476

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

3D Hand Pose Estimation problem is defined as inferring 3D pose given 2D image which contains a hand. Hand pose estimation is one of the major problems in Computer Vision tasks since hands are essential components in VR and AR fields. A set of controllers or physical sensors are needed in VR and AR environments for the interaction between human and these systems. A robust Hand pose estimation algorithm can relieve the physical requirements, which leads our hands-free. However, learning and predicting hand pose from an image is challenging due to occlusion, either self-occlusion or mutual-occlusions between hand and object. Occlusion is the main cause of accuracy drop since it limits the part of joint information. Recently introduced state-of-the-art pose estimation algorithms [14, 37, 10, 4, 46, 13, 22, 21] show impressive performance on hand pose estimation, but still suffer from occlusion problem. Another algorithm [] solves the occlusion problem by contextual learning, and this dissertation is based on it in which I was involved as the second author.

Recent approaches exploits large-scale datasets or synthetic data to mitigate the effect of the occlusion, including diverse occlusion cases to augment training data [11, 9]. It is realistically not possible to alleviate occlusion problem with data-driven approach only because of combinatorially large number of ways in which occlusion can happen.

In addition, earlier approaches to hand pose estimation assumed independent joints [6, 7, 25, 27] or gave their model prior knowledge such as physical connectivity of joints(bones) [22, 4, 2, 5, 21]. However, they overlook the hand skeleton's implicit connectivity, which affects the hand pose even though they are not explicitly connected. To overcome this limitation of previous works, I tried to improve the accuracy by representing the relation between hand joints. Hence, I refine the joints inferred from the regression module by applying contextual learning. I propose a joint embedding method, namely Joint Refinement Module, that captures comprehensive connectivity between the joints and refines incorrect joints. Inspired by the Masked Language Model of BERT [3] in Natural Language Processing (NLP), I found that refining joints pose corresponds to a part of the fill-in-the-blank task of NLP. BERT successfully performs the fill-in-the-blank task using contextual understanding. Inspired by BERT's context learning, I propose Joint Refinement Module that learns to capture joint connectivity and refines the incorrect joints pose using joint relations. The joint relations captured from the Joint Refinement Module give an intuition of where and how much to attend visible joint features to recover and refine the incorrectly inferred joints. Wide experiments on NYU [38] demonstrate advantage of the Joint Refinement Module in 3D Hand Pose Estimation task.

Chapter 2

RELATED WORKS

Hand Pose Estimation (HPE) methods typically involve getting the features of the hand image, and then regressing the hand pose from those features [20, 41, 15, 8, 42, 29, 36, 35]. Many of these methods, like [26] embed priors of the hand structure into their method to guide the pose estimation. Oberwerger et. al, for example, use a bottleneck layer to enforce learning of a lower dimensional space for representing hand pose. Many of these methods, like [15] which use regression, also have an intermediate stage of predicting the 2D heatmap for each joint. Sinha et al [32] use a separate pose regressor for each finger Mueller et al. [23] observed that since many early hand pose datasets are in third person view, methods explicitly tackling self occlusions are uncommon because they mostly happen in first person view. Further, in their paper [23] they propose a two stage pipeline to localize and then regress the hand pose in first person view. In [16], the authors use priors of hands under clicking actions to robustly estimate the occluded fingertip position. [30] also use priors by incorporating task and viewpoint specific synthetic training exemplars in their detection framework. [39, 34, 28] model object and hand together to further constrain the search space to help predict full pose under occlusion. [33] propose using bio-mechanical priors to further constrain the search space for hand pose.

Skeleton Feature Learning. Graph representation has been widely adopted to model the human skeleton [4, 19, 22, 21] because of its simplicity and existence of a robust feature extraction algorithm, namely Graph Convolutional Network (GCN) [18]. Cai *et al.* [2] apply GCN to spatio-temporal graph for 3D pose estimation. Yan *et al.* [43] propose ST-GCN for skeleton-based action recognition by combining spatial graph convolutions and temporal convolutions. Also, Doosti *et al.* [4] introduce graph U-Net to estimate 3D hand pose from 2D pose. However, a skeleton graph doesn't capture all the connectivities. There are semantic connectivities between the joints that can not be represented by bone connectivity. Therefore, modeling semantic connectivity is the critical aspect of skeleton feature learning. Thus, Multi-Scale Graph Convolutional Network (MS-GCN) [21] have been proposed to capture long-range dependencies of the joint feature using higher-order polynomials of the adjacency matrix. Still, it suffers from a biased weighting problem, that closer nodes have more possible walks than the actual K-hop neighbors due to cyclic walks. Liu *et al.* [22] directly address this problem by removing redundant walks of MS-GCN. Otherwise, Zao *et al.* [45] introduce semGCN that learn semantic information between the joints using different weighting matrices for each joint feature channel.

Chapter 3

PRELIMINARIES

3.1 Attention Mechanism

Attention mechanisms [1] explicitly model interactions between inputs. This property of attention for finding interactions between joints is used for the proposed method. A scaled dot-product attention [40] can be formulated as:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{3.1}$$

where matrices Q, K and V represent query, key, and value respectively. d_k denotes dimension of key vector. Fig 3.1 (left) shows how the mechanism is calculated.

3.2 Transformer

As most of the recent architectures which show state-of-the-art performance in the NLP task are based on Transformer Architecture [40], it has been a de-facto standard in NLP. Transformer exploits Attention mechanism in both encoder and decoder so that it can represent contextual meaning in sentences and infer next word by considering contextual meaning. Transformer also takes advantage of parallelism by applying

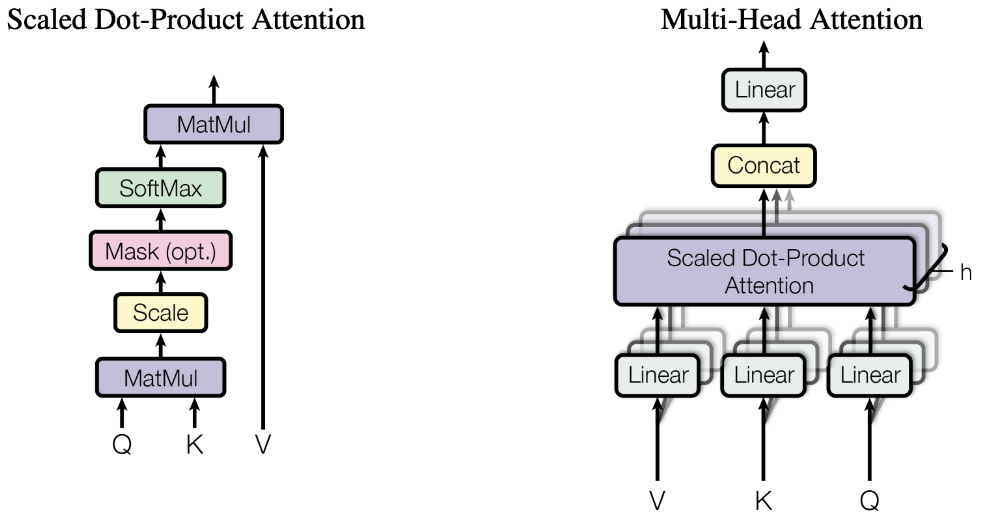

Figure 3.1: Attention Mechanisms, Figure from [40]

Multi-Head Attention mechanism. Fig 3.1 (right) illustrates how the Multi-Head Attention works and Fig 3.2 illustrates Transformer architecture.

3.3 Masked Language Model

This dissertation is mostly inspired by the word embedding method of BERT. It shows an impressive performance in NLP tasks such as general language understanding and question and answering. The BERT [3] is a stacked Transformer Encoder [40] which contains a Multi-Head Self-Attention module that captures connectivity between the input elements. To fully comprehend a sentence, the individual meaning of the words is not sufficient, but the model must understand how words correlate in the context of the sentence. The BERT learns the context of words through the mask inference task and provides the probability distribution of words. BERT is pre-trained on two unsupervised tasks: 1) Masked Language Model (MLM) and 2) Next Sentence Prediction. Especially, MLM is to predict randomly selected masked words in a sentence, and

BERT is trained better to comprehend the grammar and syntax of text while predicting masks.

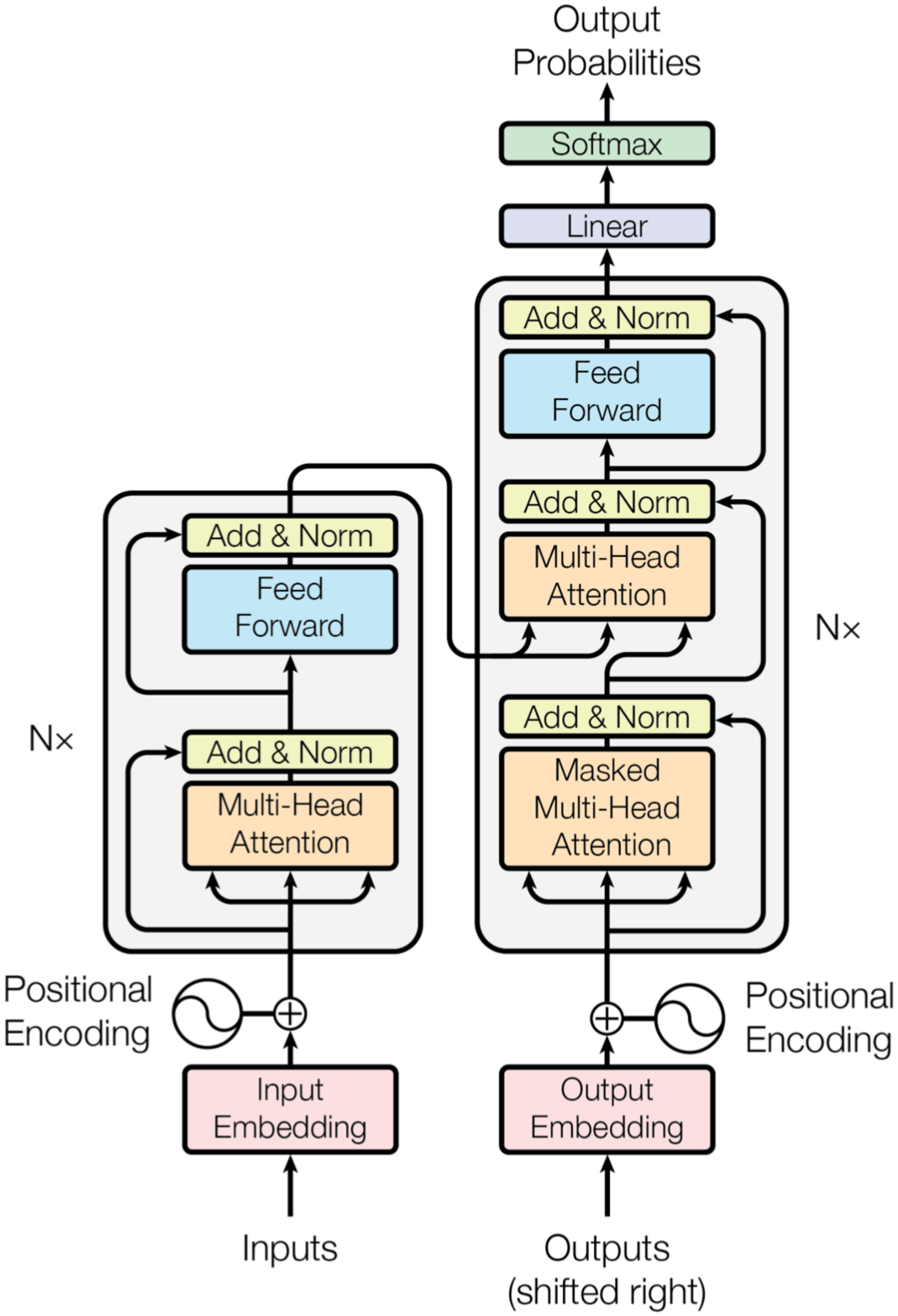

Figure 3.2: Transformer Architecture, Figure from [40]

Chapter 4

Method

4.1 Problem Definition

3D Hand Pose Estimation problem can be defined as inferring 3D joint locations given 2D images. Hence, the objective function is to minimize the Euclidean distance between the predicted value and the ground truth.

4.2 3D Hand Pose Estimation Framework

Instead of directly predicting 3D joints from images, I decompose the problem into 2D estimation from images and 3D regression from estimated 2D locations. Moreover, I add a refinement module in the final stage. Hence, my proposed framework consists of three modules; Dense Representation Module, 3D Regression Module, and Joint Refinement Module.

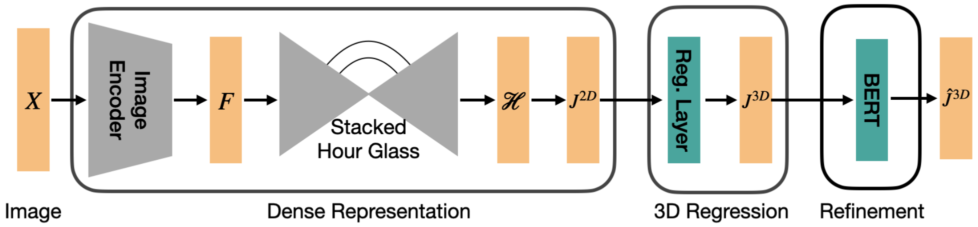

Figure 4.1: Proposed framework

4.2.1 Dense Representation Module

I first extract feature $F \in \mathbb{R}^{256\times64\times64}$ from input image X. ResNet50 [12] trained on ImageNet [31] is used as the image encoder. Then, image feature map F that comes from image encoder is fed to Stacked HourGlass network [24] which returns heatmap $H \in \mathbb{R}^{14\times64\times64}$. As heatmaps contains dense representation for 14 joints, I can get $J^{2D} \in \mathbb{R}^{14\times2}$ by simply applying argmax function to $\mathcal{H}$. Fig 4.2 shows an example of heatmap $\mathcal{H}$. In addition, I also get the confidence vector $C \in \mathbb{R}^{14\times1}$ of argmax points which has far lower value in case of occlusion. For training, I use a fixed sized Gaussian blob centered at ground truth 2D joint locations J^{2D}_{GT} as a ground truth heat maps $\mathcal{H}_{\mathcal{GT}}$. L2 loss is applied to minimize the loss between $\mathcal{H}$ and $\mathcal{H}_{GT}$.

$$\mathcal{L}_H = \|\mathcal{H} - \mathcal{H}_{GT}\|_2^2, \tag{4.1}$$

4.2.2 3D Regression Module

3D Regression Module outputs joint locations $J^{3D} \in \mathbb{R}^{14\times3}$ given H. A linear layer predicts value of offset $d \in \mathbb{R}^{14\times1}$ from H. Then, by combining J^{2D} and d, I finally get J^{3D}. L2 loss function is applied to reduce the 3D joint location error.

$$\mathcal{L}_{J^{3D}} = \|J^{3D} - J^{3D}_{GT}\|_2^2, \tag{4.2}$$

4.2.3 Joint Refinement Module

The Joint Refinement Module refines J^{3D} which comes from the 3D Regression Module and outputs $\hat{J}^{3D}$ which has exactly the same dimension with J^{3D}. Implementation of Joint Refinement Module follows that of BERT; however, the output is logit vector in Joint Refinement Module whereas the output is probability vector in BERT. Fig 4.3 visualizes the architecture of Joint Refinement Module. Before end-to-end training, this module is pre-trained with J^{3D}_{GT}. Pre-training details will be described in equation 4.3. For training, L1 loss is applied to reduce the error between $\hat{J}^{3D}$ and J^{3D}_{GT}. I also utilize confidence vector C to reflect confidence information while fine-tuning J^{2D}. C^{-1} is defined as a element-wise reciprocal vector of C which can be formulated as 4.3.

$$C^{-1} = (1/C_1, 1/C_2, ..., 1/C_N) \tag{4.3}$$

I use C^{-1} as a weight for refinement since lower confidence literally means that the Stacked HourGlass network is not confident for its output. Hence, the Hadamard product is used to reflect the confidence of the Stacked HourGlass network.

$$\mathcal{L}_R = \|C^{-1} \odot (\hat{J}^{3D} - J^{3D}_{GT})\|_1 \tag{4.4}$$

During end-to-end training, Joint Refinement Module is fine-tuned with other modules.

4.3 Pre-training

As my proposed framework consists of three modules, each module can be pre-trained. I choose the best pre-training option by experimenting with all possible combinatorial cases. Experiments show that fine-tuning the entire architecture with the pre-trained Stacked HourGlass network and pre-trained Joint Refinement Module acquires the best performance. The pre-training process also reduces convergence time during the end-to-end training.

4.3.1 Stacked HourGlass

Image feature F, the output of the freezed ResNet50 encoder, is invariant during the training. This means the Stacked HourGlass network has invariant input values during the training so that caching image feature F reduces calculation. Hence, I cached the image feature F as a file and fed it to the Stacked HourGlass network during the pre-training. The equation 4.1 is used to pre-train the Stacked HourGlass network. I stack only one HourGlass network in this framework due to the GPGPU memory limitation. Pre-training hyperparameters are written in Table 4.1.

Table 4.1: Pre-training hyperparmeters of Stacked HourGlass

Hyperparameter	Value
OPTIMIZER	Adam [17]
EPOCH	100
BATCH SIZE	32
LEARNING RATE	1e-4
LR DECAY	Exponential
LR DECAY FACTOR	0.99
WARM-UP STEP	1000
# OF STACK	1

4.3.2 Joint Refinement Module

Inspired by the method of BERT, I pre-train the Refinement Module using 3D joints position with noised input. During the pre-training, input J_{GT}^{3D} is perturbed by Gaussian noise $Z \sim \mathcal{N}(0, 1)$. The Joint Refinement Module learns the relation between the joints as it learns the contextual meaning of words in Neural Language Processing. L2 loss is used for the pre-training Refinement Module. Pre-training hyperparameters are written in Table 4.2.

Table 4.2: Pre-training hyperparmeters of Joint Refinement Module

Hyperparameter	Value
OPTIMIZER	Adam
EPOCH	500
BATCH SIZE	128
LEARNING RATE	1e-4
LR DECAY	Cosine
WARM-UP STEP	1000
# OF LAYER	12
# OF HEAD	8
BERT DIMENSION	128

4.4 Training

End-to-end training works as a fine-tuning the whole framework. The total loss $\mathcal{L}_{total}$ is as follows,

$$\mathcal{L}_{total} = \mathcal{L}_{\mathcal{H}} + \lambda_1 \mathcal{L}_{J^{3D}} + \lambda_2 \mathcal{L}_R \tag{4.5}$$

where λ_1 and λ_2 are weight values for loss. Hyperparamters are written in Table 4.3

Table 4.3: Hyperparmeters of end-to-end training

Hyperparameter	Value
OPTIMIZER	Adam
EPOCH	100
BATCH SIZE	32
LEARNING RATE	1e-4
LR DECAY	Exponential
WARM-UP STEP	1000

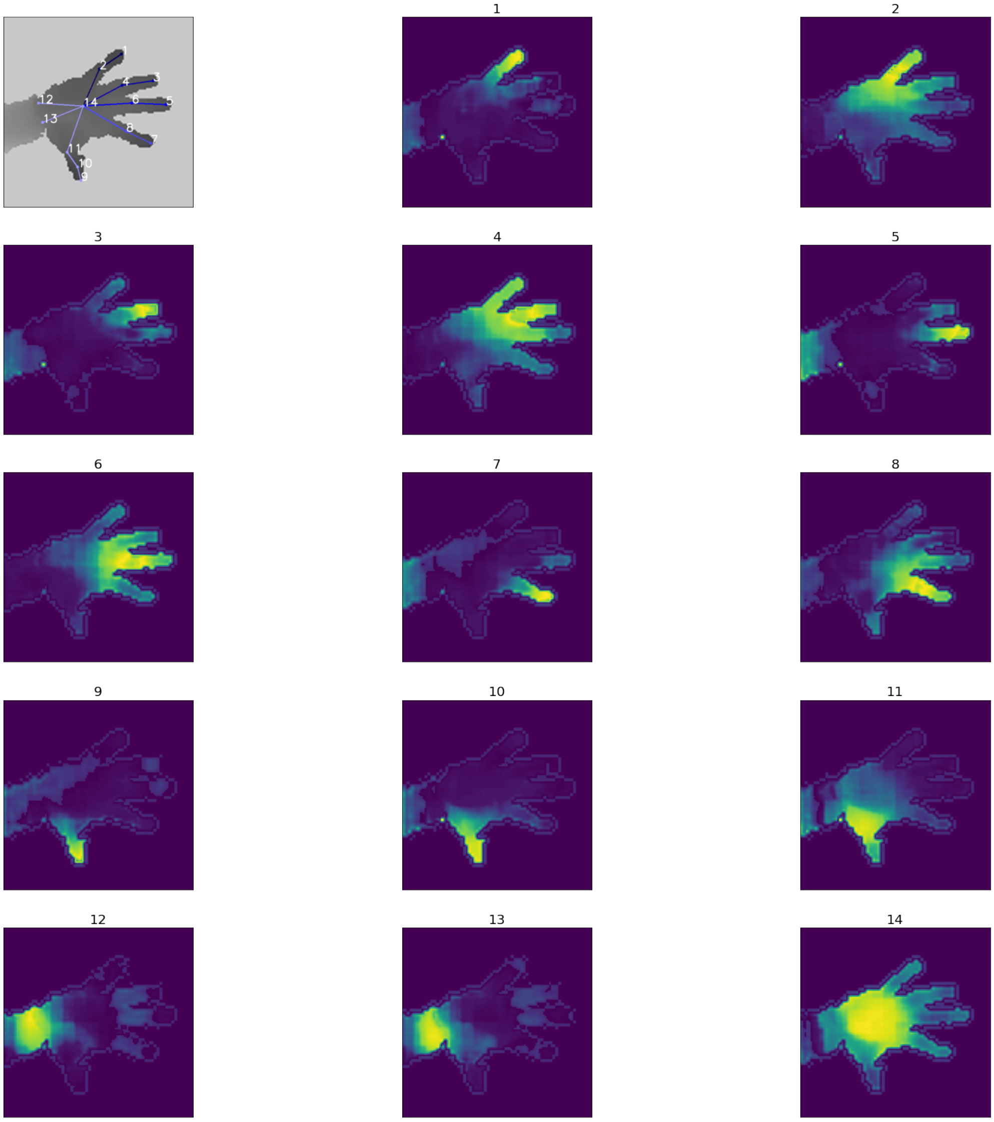

Figure 4.2: A sample of heatmap

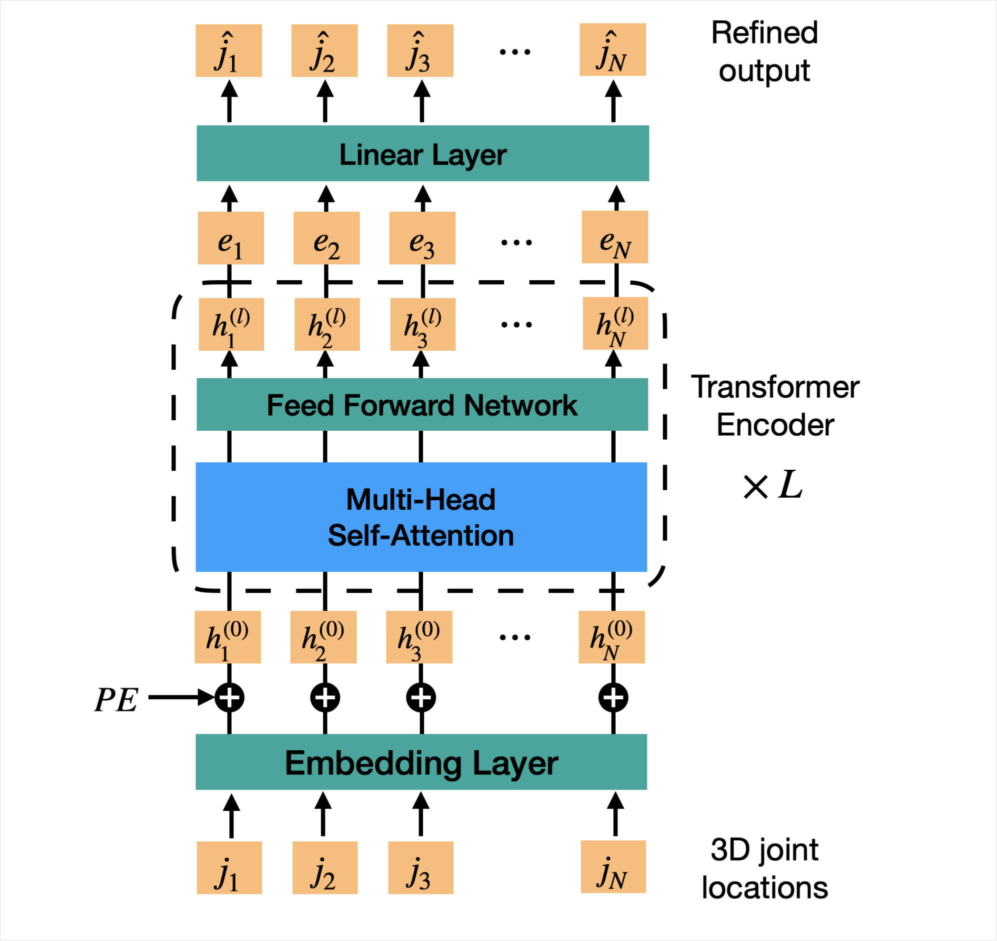

Figure 4.3: Joint Refinement Module

Chapter 5

EXPERIMENTS

I evaluate the Joint Refinement Module with the NYU dataset. The experiment is implemented by the open source machine learning framework PyTorch and performed on a single GPGPU, NVIDIA 1080 TI.

5.1 Dataset

I use NYU dataset for the experiments. The NYU Hand Pose dataset [38] collects comprehensive coverage of hand poses, containing 72,757 and 8,252 RGB-D frames for training and evaluation. Each frame includes ground truth annotation of 3D joint locations. The dataset presents a third-person view without the objects. I use a subset of 14 hand joints following [8], to compare our results with other methods. It contains 9% of occluded joints on average according to [44]. Fig 5.1 is a sample of NYU dataset and Fig 5.2 is an example of self-occluded case.

5.2 Experimental Results

5.2.1 Quantative Results

I compare the 3D joint error of my proposed framework to AWR [14] which also applies regression based method to solve Hand Pose Estimation problem. I report the result in Table 5.1. My proposed framework outperformed AWR [14] method by 0.15mm. The results also shows that pre-trained refinement module improves 0.27mm compared to the method without refinement module. Notably, refinement module didn't improve the accuracy of the framework when it had not been pre-trained. Applying refinement module without pre-training even deteriorates the accuracy of the framework. In addition, I reported joint-wise error in Fig 5.3. It shows that refinement module improves accuracy for all joints. From this results, I concluded that the refinement module learns how to correct the error by pre-training.

Table 5.1: Pose estimation accuracy comparison against AWR[14]

Method	3D Joint Error(mm)
AWR Hourglass-1 [14]	7.70
Ours Hourglass-1 w/o refinement	7.82
Ours Hourglass-1 w/o pre-training	7.86
Ours Hourglass-1	**7.55**

5.2.2 Qualitative Results

I compare the 3D joint error of my proposed framework to AWR [14] which also applies the regression-based method to solve the Hand Pose Estimation problem. I report the result in Table 5.1. My proposed framework outperformed AWR [14] method by 0.15mm. The results also show that the pre-trained refinement module improves 0.27mm compared to the method without the refinement module. Notably, the refinement module didn't improve the accuracy of the framework when it had not been pre-trained. Applying refinement modules without pre-training even deteriorates the accuracy of the framework. In addition, I reported joint-wise error in Fig 5.3. It shows that the refinement module improves accuracy for all joints. From these results, I concluded that the refinement module learns how to correct the error by pre-training.

5.2.3 Computational Complexity

BERT has a computational complexity of $O(n^2)$. However, unlike the variable input length in Natural Language Processing tasks, 3D Hand Pose Estimation problem has fixed length input. Hence, Joint Refinement Module doesn't allocate any redundant memory. It has 1.6M learnable parameters, which is 5.2% of the total parameters. This means Joints Refinement Module is computationally inexpensive while it improves performance.

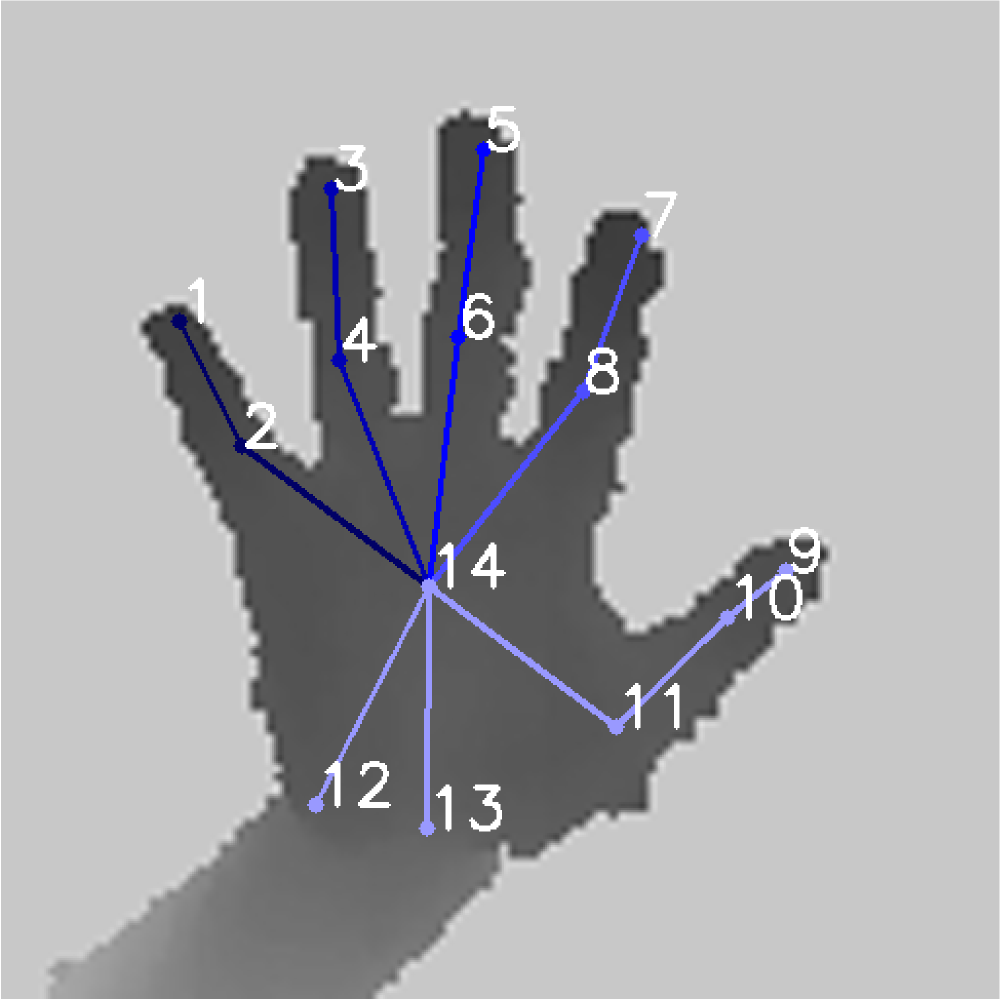

Figure 5.1: An example of NYU dataset

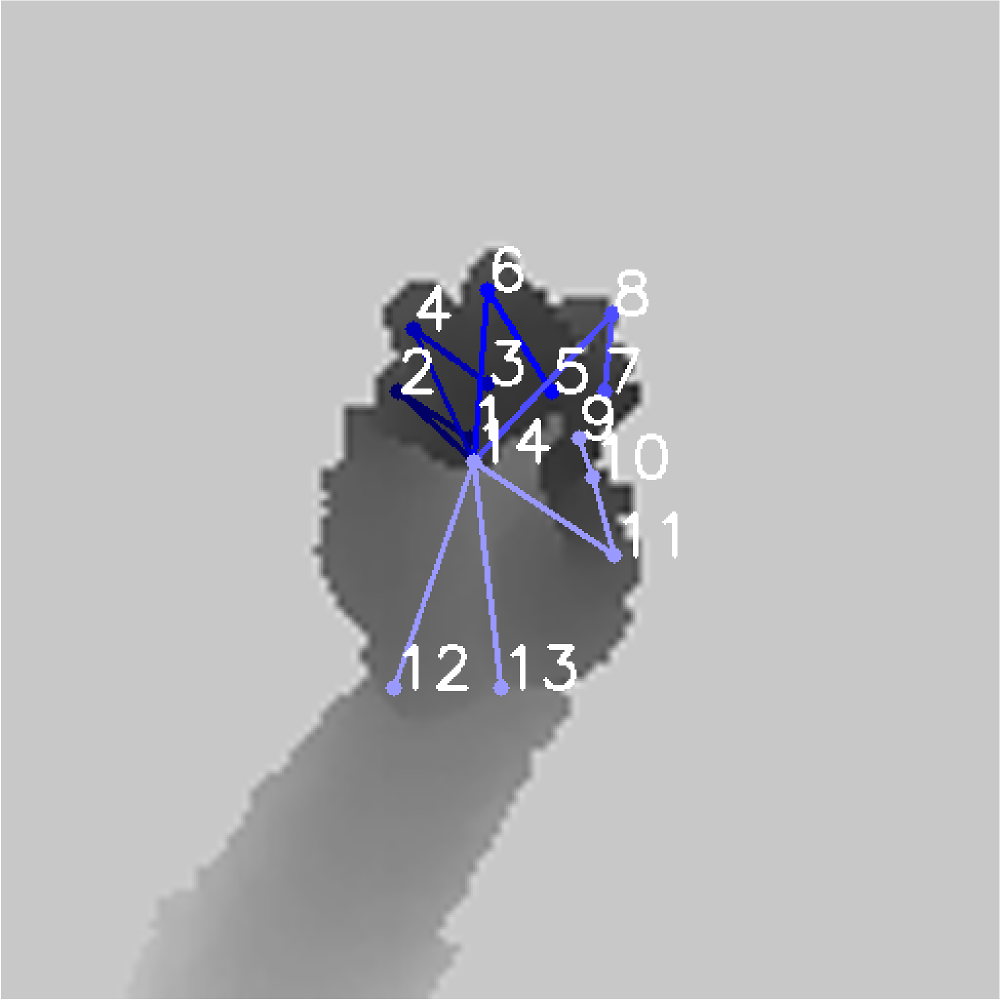

Figure 5.2: An example of occluded hand

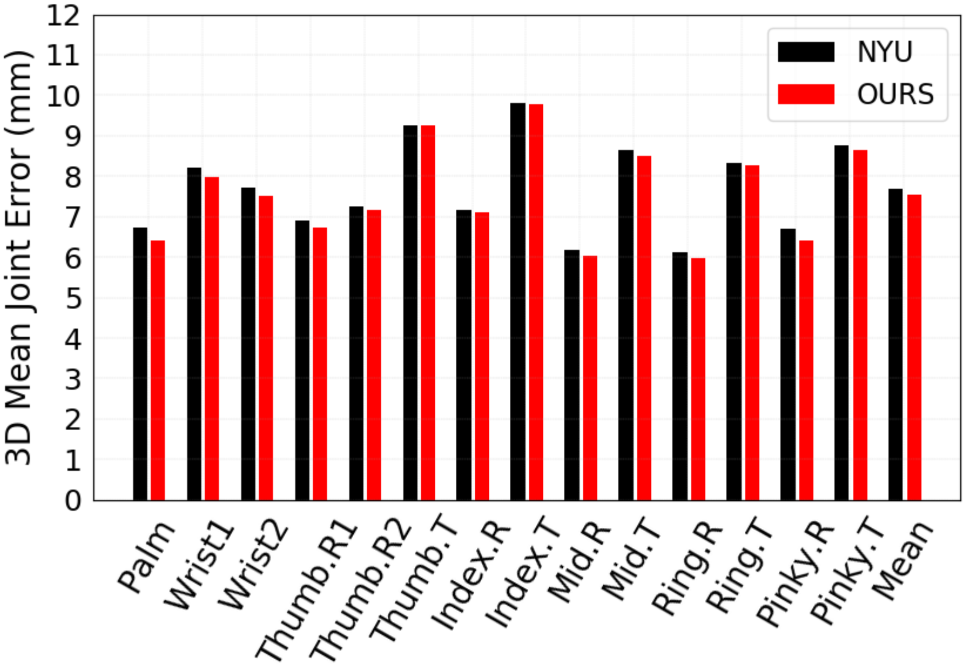

Figure 5.3: Error bar plot for NYU

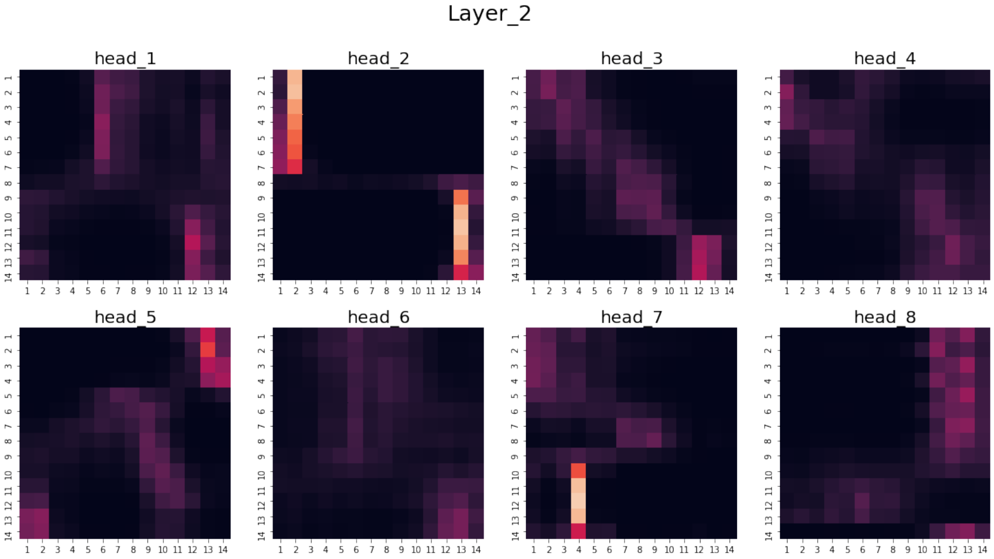

Figure 5.4: Visualization of multi-head attention matrices, second layer

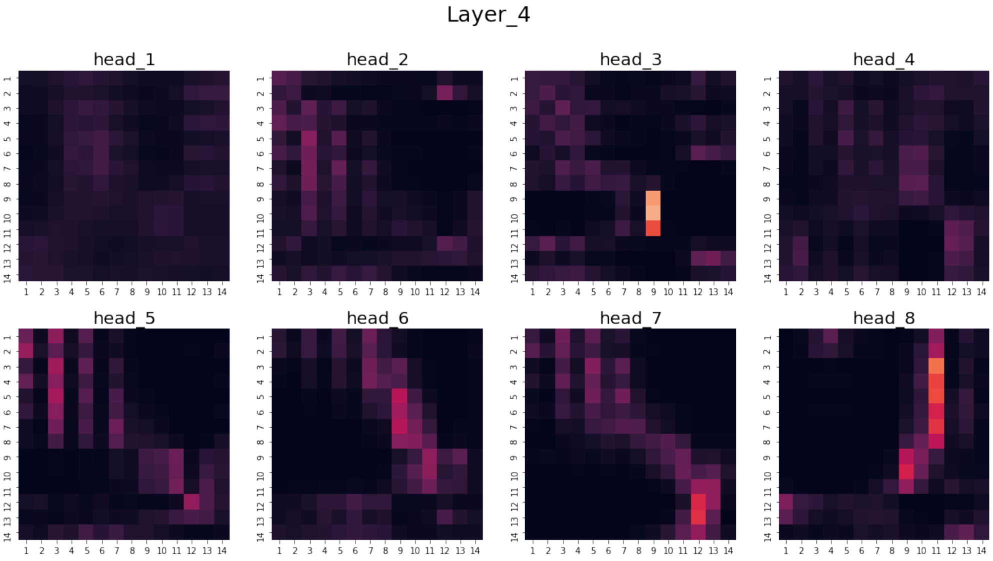

Figure 5.5: Visualization of multi-head attention matrices, 4th layer

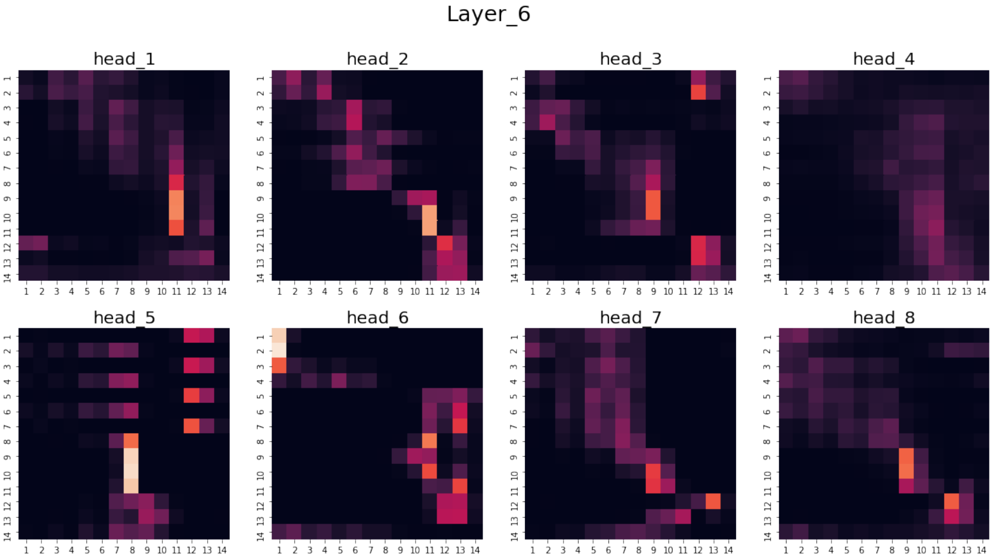

Figure 5.6: Visualization of multi-head attention matrices, 6th layer

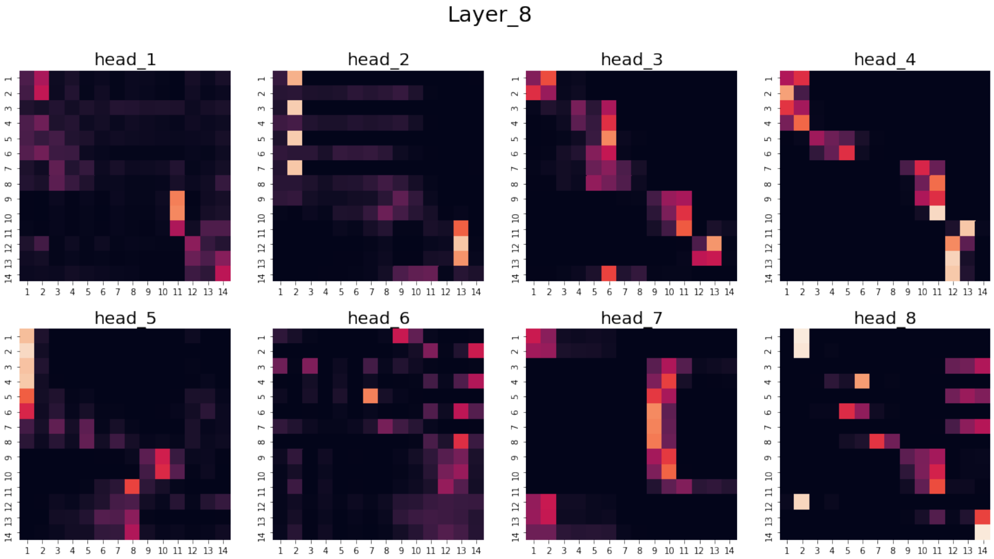

Figure 5.7: Visualization of multi-head attention matrices, 8th layer

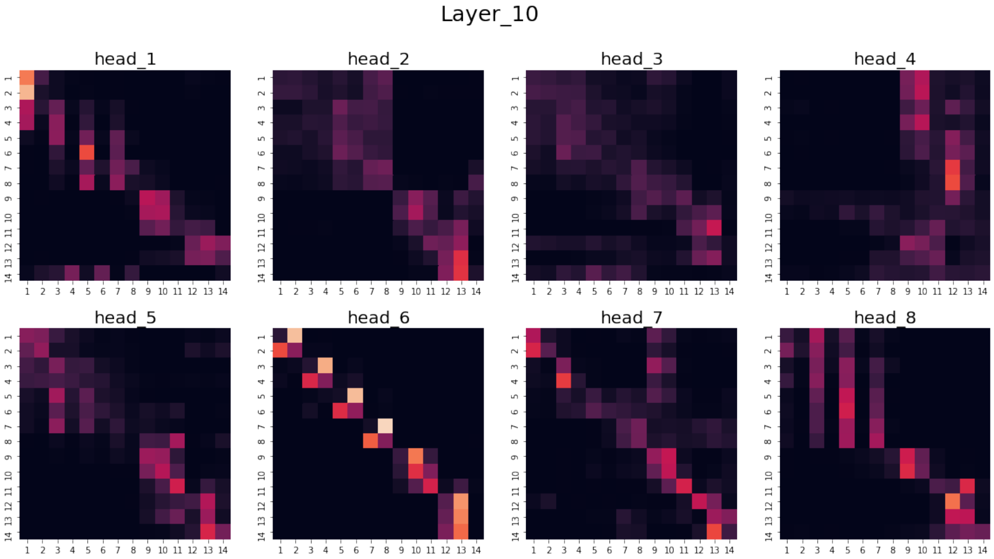

Figure 5.8: Visualization of multi-head attention matrices, 10th layer

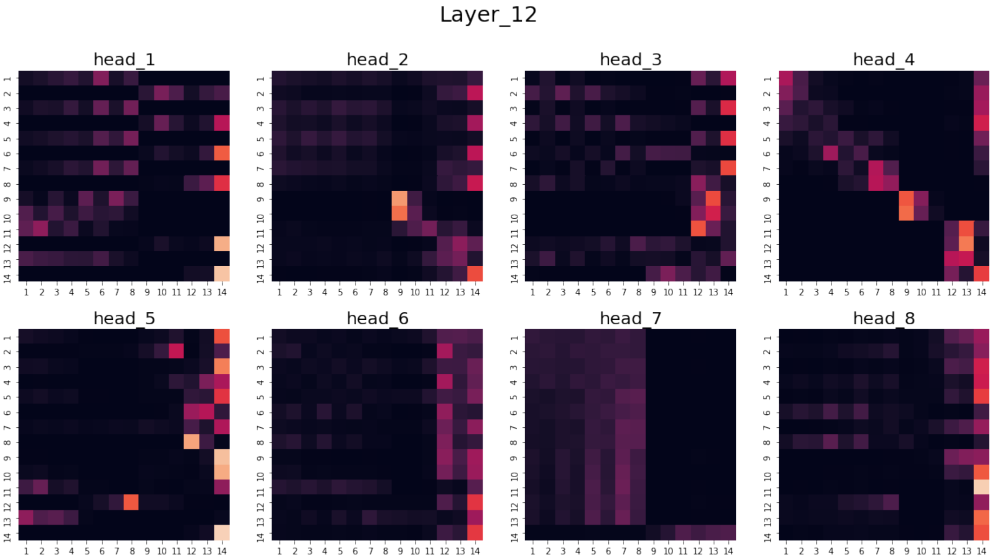

Figure 5.9: Visualization of multi-head attention matrices, last layer

Chapter 6

CONCLUSION

The experiment shows that Joint Refinement Module represents joint relation by learning attention weights between them. In lower layers, the module has a strong tendency to focus on adjacent indices; however, in higher layers, the module focuses on wider ranges. It is far different from the explicit connection between joints. Furthermore, my proposed framework outperformed the existing method by simply applying the refinement module. This means the previous assumption that each joint are only affected by physically adjacent joint hinders the model from properly predicting occluded information. Learning joint relation from the data and predicting occluded information based on joint relation is a far better approach because it doesn't assume anything. Hence, the unnecessary constraint doesn't exist.

The experiment also implies that pre-training needs to capture comprehensive connectivity of joints. My proposed model gets the best results with the pre-trained module. However, when it was trained without a pre-trained model, the refinement module exacerbated the performance. Due to this result, I concluded that it's hard for the model to understand the comprehensive connectivity of joints when it is directly trained in an end-to-end manner.

Bibliography

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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

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

[4] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6608–6617, 2020.

[5] Linpu Fang, Xingyan Liu, Li Liu, Hang Xu, and Wenxiong Kang. Jgr-p2o: Joint graph reasoning based pixel-to-offset prediction network for 3d hand pose estimation from a single depth image. In *European Conference on Computer Vision*, pages 120–137. Springer, 2020.

[6] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8417–8426, 2018.

[7] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1991–2000, 2017.

[8] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 475–491, 2018.

[9] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3196–3206, 2020.

[10] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 571–580, 2020.

[11] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11807–11816, 2019.

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

[13] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Hand-transformer: Non-autoregressive structured modeling for 3d hand pose estimation. In *European Conference on Computer Vision*, pages 17–33. Springer, 2020.

[14] Weiting Huang, Pengfei Ren, Jingyu Wang, Qi Qi, and Haifeng Sun. Awr: Adaptive weighting regression for 3d hand pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11061–11068, 2020.

[15] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018.

[16] Youngkyoon Jang, Seung-Tak Noh, Hyung Jin Chang, Tae-Kyun Kim, and Woontack Woo. 3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):501–510, 2015.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980 Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

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

[19] Bin Li, Xi Li, Zhongfei Zhang, and Fei Wu. Spatio-temporal graph routing for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8561–8568, 2019.

[20] Jia Li and Zengfu Wang. Local regression based hourglass network for hand pose estimation from a single depth image. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1767–1772. IEEE, 2018.

[21] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3595–3603, 2019.

[22] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 143–152, 2020.

[23] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1154–1163, 2017.

[24] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[25] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *Proceedings of the IEEE international conference on computer vision Workshops*, pages 585–594, 2017.

[26] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[27] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3316–3324, 2015.

[28] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical

constraints. In *2011 International Conference on Computer Vision*, pages 2088–2095. IEEE, 2011.

[29] Pengfei Ren, Haifeng Sun, Qi Qi, Jingyu Wang, and Weiting Huang. Srn: Stacked regression network for real-time 3d hand pose estimation. In *BMVC*, page 112, 2019.

[30] Grégory Rogez, Maryam Khademi, JS Supančič III, Jose Maria Martinez Montiel, and Deva Ramanan. 3d hand pose detection in egocentric rgb-d images. In *European Conference on Computer Vision*, pages 356–371. Springer, 2014.

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[32] Ayan Sinha, Chiho Choi, and Karthik Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4150–4158, 2016.

[33] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. *arXiv preprint arXiv:2003.09282*, 8, 2020.

[34] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision*, pages 294–310. Springer, 2016.

[35] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 824–832, 2015.

[36] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3786–3793, 2014.

[37] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+ o: Unified egocentric recognition of 3d hand-object poses and interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4511–4520, 2019.

[38] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):1–10, 2014.

[39] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118(2):172–193, 2016.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[41] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2018.

[42] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 793–802, 2019.

[43] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[44] Qi Ye and Tae-Kyun Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–817, 2018.

[45] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[46] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5346–5355, 2020.

초 록

3D Hand Pose Estimation 문제는 한 장의 2차원 image를 이용하여 3차원의 손 좌표를 추정하는 문제로, 2차원 image에서 손의 일부가 가려지는 경우들 때문에 현존하는 접근방식으로 풀기에 까다로운 문제이다. 최근에 연구자들은 많은 양의 데이터를 모으거나, 합성하여 이 문제를 해결하려 했다. 하지만, 이러한 데이터 기반 접근들은 각 관절들이 독립적이라 가정하고 문제를 풀거나, 물리적으로 드러나는 관절들의 연결 관계만 가지고서 문제를 해결하려 했기때문에 성능 향상에 한계가 있었다.

이러한 문제를 완화하기 위해, 손 관절의 3차원 좌표들 간의 관계를 학습시키고 이를 기반으로 미세조정을 하여 전반적인 성능을 끌어 올리는 방법을 이 논문에서 제안 한다. 자연어 처리 분야에서 가장 많이 쓰이는 BERT 모델에서 영감을 받았으며, BERT를 이용하여 보이지 않는 손에 대하여 잘 추정하도록 하는 모듈을 추가함으로써 기존에 있던 접근방식들 보다 더 좋은 결과를 실험에서 얻을 수 있었다. 또한, 물리적인 관절간의 연결 관계에 갇혀있지 않고, 모델이 데이터로부터 각 관절간의 영향력을 파악하여 위치를 세부 조정하게 하였다. 이렇게 학습한 연결 관계를 시각화 하여 이 논문에 일부 소개하였고, 이를 통해 눈에 보이는 물리적인 연결관계 뿐만 아니라 관계없어 보이는 관절들 간에도 영향을 주고받고 있다고 가정하는 것이 훨씬 더 좋은 접근방법임을 관찰할 수 있었다.

주요어: 컴퓨터 비전, 3차원 손 좌표 추정, 자연어 처리

학번: 2019-23476