Vision-based 3D human and hand pose analysis

Cai, Yujun

2021

Cai, Y. (2021). Vision-based 3D human and hand pose analysis. Doctoral thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/153319>

<https://hdl.handle.net/10356/153319>

<https://doi.org/10.32657/10356/153319>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 15:37:12 SGT

Vision-based 3D Human and Hand Pose Analysis

Yujun Cai

Interdisciplinary Graduate School

Institute for Media Innovation

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

4th May 2021

.....

Date

Yujun Cai

.....

Yujun Cai

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

9 April 2021

.....
Date

.....
A/P Cham Tat Jen

Authorship Attribution Statement

This thesis contains material from 5 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as the first author.

Chapter 3.2 is published as [Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan](#). "Weakly-supervised 3d hand pose estimation from monocular rgb images." In Proceedings of the European Conference on Computer Vision (ECCV), pp. 666-682. 2018.

The contributions of the co-authors are as follows:

- Yujun Cai designed the experiments, and prepared the manuscript.
- Prof. Junsong Yuan, Prof. Jianfei Cai and Liuhaog Ge provided the initial project direction and revised the manuscript drafts.

Chapter 3.3 is published as [Yujun Cai, Liuhao Ge, Jianfei Cai, Nadia Magnenat-Thalmann, and Junsong Yuan](#). "3D hand pose estimation using synthetic data and weakly labeled RGB images." IEEE transactions on pattern analysis and machine intelligence (2020).

The contributions of the co-authors are as follows:

- Yujun Cai proposed the idea, designed the experiments, and prepared the manuscript.
- Prof. Jianfei Cai, Prof. Junsong Yuan and Liuhaog Ge discussed the details of the experiments and revised the manuscript drafts.
- Prof. Nadia Magnenat-Thalmann joined the discussion of this project.

Chapter 4 is published as [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, pp. 2272-2281. 2019.

The contributions of the co-authors are as follows:

- Yujun Cai proposed the idea, designed the experiments, and prepared the manuscript.
- Prof. Jianfei Cai, Prof. Cham Tat Jen, Prof. Junsong Yuan, Liuhaog Ge, and Prof. Jun Liu discussed the details of the experiments and revised the manuscript drafts.
- Prof. Nadia Magnenat-Thalmann joined the discussion of this project.

Chapter 5 is published as [Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, Ding Liu, Jing Liu and Nadia magnenat Thalmann](#) . "Learning progressive joint propagation for human motion prediction." In European Conference on Computer Vision, pp. 226-242. Springer, Cham, 2020.

The contributions of the co-authors are as follows:

- Yujun Cai proposed the idea, designed the experiments, and prepared the manuscript.
- Lin Huang and Yiwei Wang helped prepare the manuscript and joined the experiment discussion
- Prof. Jianfei Cai, Prof. Cham Tat Jen, Prof. Junsong Yuan, Prof. Jun Liu, Yiheng Zhu, Xiaohui Shen, Ding Liu, Jing Liu discussed the details of the experiments and revised the manuscript drafts.
- Prof. Nadia Magnenat-Thalmann and Xu Yang joined the discussion of this project.

Chapter 6 is published as [Yujun Cai, Yiwei Wang, Yiheng Zhu, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Chuanxia Zheng, Sijie Yan, Henghui Ding, Xiaohui Shen, Ding Liu, Nadia Magnenat Thalmann](#) "A Unified 3D Human Motion Synthesis Model via Conditional Variational Auto-Encoder." In Proceedings of the IEEE/CVF International Conference on Computer Vision.

The contributions of the co-authors are as follows:

- Yujun Cai proposed the idea, designed the experiments, and prepared the manuscript.
- Yiwei Wang and Yiheng zhu helped prepare the manuscript and joined the experiment discussion
- Prof. Jianfei Cai, Prof. Cham Tat Jen, Prof. Junsong Yuan, Prof. Jun Liu, Xiaohui Shen, Ding Liu discussed the details of the experiments and revised the manuscript drafts.
- Prof. Nadia Magnenat-Thalmann, Chuanxia Zheng, Sijie Yan and Henghui Ding joined the discussion of this project.

4th May 2021

.....

Date

Yujun Cai

.....

Yujun Cai

Acknowledgements

First and foremost, I wish to express my greatest gratitude to my supervisors Prof. Junsong Yuan, Prof. Jianfei Cai, and Prof. Cham Tat Jen. Your kind support throughout my Ph.D. study has given me much encouragement to explore my own way of doing research. During my P.h.D study, I'm also highly impressed by your patient, cautious attitude toward research work. I would also like to say thanks to Prof. Nadia Magnenat Thalmann and Prof. Yiyu Cai for their valuable guidance for my work. The time of studying with you is enjoyable and helpful in my life.

I would also like to thank the fund and Ph.D. grant from the Institute for Media Innovation (IMI) and the Interdisciplinary Graduate School (IGS), Nanyang Technological University, Singapore. It's my great honor to study in IGS-IMI and get the valuable resources for my research work.

My thanks to my dear colleagues in NTU, Liuhao Ge, Junwu Weng, Zhenzhen Wang, Jun Liu, Tan Yu, Chuanxia Zheng, Guoxian Song, Zhijie Zhang, Yuedong Chen, Bo Hu, Zhiwen Fang, TianYi Zhang, Chi Zhang, Xu Yang. Thanks for your kind help in my work and life! Especially, I would like to thank Liuhao Ge for his patience and guidance when I'm junior student, which greatly helps my research study in the long-term.

My sincere appreciation also goes to my QE examiners, the anonymous reviewers of my thesis and the panel of my Ph.D oral defense. Thank you for your valuable suggestions to my research and presentation. I am thankful to my mentors at Bytedance, Xiaohui Shen, Yiheng Zhu and Ding Liu, and mentors at Facebook Reality Lab, Amy Zhao, Lingling Tao, Chengcheng Tang, Yijing Li and Christopher Twigg.

Finally, I would like to thank my dear parents and my husband. Thank you for your encouragement and warm support during my Ph.D study. Without your companions, the thesis would not be possible.

To my dear family for their encouragement and support.

Contents

Acknowledgements	vi
List of Figures	xiii
List of Tables	xvii
Summary	xix
1 Introduction	1
1.1 Background and Motivation	1
1.2 Research Scope	2
1.2.1 3D Pose Estimation	2
1.2.2 3D Motion Prediction	3
1.2.3 3D Motion Synthesis	3
1.3 Thesis Contribution	3
1.4 Outline of the Thesis	6
2 Literature Review	9
2.1 3D Pose Estimation	9
2.1.1 3D Hand Pose Estimation	9
2.1.2 3D Human Pose Estimation	10
2.1.3 3D Mesh Reconstruction	11
2.2 3D Human Motion Prediction	12
2.3 Motion Synthesis	13
2.4 3D Pose Datasets	14
2.4.1 Hand Pose Datasets	14
2.4.2 Human Pose Datasets	15
2.5 Evaluation Metrics	16
2.5.1 Metrics for 3D Hand Pose Estimation	16
2.5.2 Metrics for 3D Human Pose Estimation	16
2.5.3 Metrics for 3D Motion Prediction	17
2.5.4 Metrics for 3D Motion Synthesis	17
3 Weakly-supervised 3D Hand Pose Estimation	19

3.1	Motivation	19
3.2	Cascaded Framework	21
3.2.1	Introduction	21
3.2.2	Methodology	23
3.2.2.1	Overview	23
3.2.2.2	2D Pose Estimation Network	24
3.2.2.3	Regression Network	24
3.2.2.4	Depth Regularizer	25
3.2.2.5	Training	27
3.2.3	Experiments	27
3.2.3.1	Implementation Details	27
3.2.3.2	Datasets and Metrics	28
3.2.3.3	Quantitative Results	29
3.2.3.4	Qualitative Results	32
3.3	Conditional Variational Autoencoder (CVAE)-based Statistical Framework	35
3.3.1	Introduction	35
3.3.2	Methodology	36
3.3.2.1	Overview	36
3.3.2.2	2D Pose Estimation Network	38
3.3.2.3	CVAE-based Statistical Network	39
3.3.2.4	Depth Regularizer	43
3.3.2.5	Training	43
3.3.3	Experiments	45
3.3.3.1	Implementation Details	45
3.3.3.2	Datasets and Metrics	47
3.3.3.3	Self-Comparison	49
3.3.3.4	Cascaded Depth Regularizer Versus Paralleled one	52
3.3.3.5	Comparison with State-of-the-art Methods	53
3.3.3.6	Qualitative Results	55
3.4	Runtime Analysis	56
3.5	Conclusion	57
4	Exploiting Spatial-temporal Relationships for 3D Pose Estimation	59
4.1	Introduction	59
4.2	Methodology	62
4.2.1	Overview	62
4.2.2	Spatial-temporal Graph Construction	63
4.2.3	Revisiting Graph Convolutional NNs	63
4.2.4	Graph Convolution for Pose Estimation	64
4.2.5	GCN-based Local-to-global Prediction	65
4.2.6	Pose Refinement	66
4.2.7	Training	67

4.3	Experiments	68
4.3.1	Implementation Details	68
4.3.2	Datasets	69
4.3.3	Evaluation Metrics	71
4.3.4	Comparison with the State-of-the-art	71
4.3.5	Ablation Studies	72
4.3.6	Qualitative results.	74
4.4	Runtime Analysis	74
4.5	Conclusion	75
5	Progressive Joint Propagation for Human Motion Prediction	77
5.1	Introduction	77
5.2	Methodology	80
5.2.1	Overview	80
5.2.2	Revisiting Transformer	81
5.2.3	Transformer for Pose Prediction	81
5.2.4	Progressive Joint Propagation	83
5.2.5	Dictionary	84
5.2.6	Training Strategy	85
5.3	Experiments	87
5.3.1	Implementation details	87
5.3.2	Datasets and Evaluation Metrics	87
5.3.3	Comparison with the State-of-the-art Methods	89
5.3.4	Ablation Study	91
5.4	RunTime Analysis	93
5.5	Conclusion	93
6	Investigating Unified 3D Human Motion Synthesis Model	95
6.1	Introduction	95
6.2	Proposed Method	98
6.2.1	CVAE-based Statistical Framework	99
6.2.2	Action-Adaptive Modulation	101
6.2.3	Cross-Attention Mechanism	102
6.2.4	Training	103
6.3	Experiments	105
6.3.1	Implementation Details	105
6.3.2	Datasets	105
6.3.3	Evaluation Metrics	107
6.3.4	Results of Pose Series Generation	108
6.3.5	Ablation Study	110
6.4	Runtime Analysis	111
6.5	Conclusion	111
7	Conclusion and Future Work	113

7.1	Conclusion	113
7.2	Future Work	114
7.2.1	Understanding Hand-Object Interactions	114
7.2.2	Human Action Understanding	115
7.2.3	Self-supervised Learning	115
	Author's Publications	116
	Bibliography	118

List of Figures

1.1	(a) Example of 3D pose estimation. (b) Example of motion prediction	2
1.2	Example of 3D motion synthesis	2
1.3	Outline of this thesis.	7
2.1	Example of Motion prediction.	12
3.1	Illustration of the concept of weakly supervised 3D hand pose estimation. Different from conventional fully-supervised methods (a) that use 3D labels to guide joint predictions, our proposed weakly-supervised method (b) leverages the reference depth map.	21
3.2	We present a weakly-supervised approach for 3D hand pose estimation from monocular RGB-only input.	22
3.3	Overview of our proposed weakly-supervised 3D hand pose regression network, which is trained in an end-to-end manner. During testing, real images only go through the part of the network in the dashed line box.	23
3.4	Network architecture of our proposed depth regularizer.	26
3.5	Comparisons of 3D PCK results of different baselines with our method on STB [1].	28
3.6	Different annotation schemes on RHD [2] and STB [1] dataset	28
3.7	The effect of the proposed depth regularizer in fully-supervised setting on RHD [2] and STB datasets [1]. Left: 3D PCK on RHD dataset. Right: 3D PCK on STB dataset.	30
3.8	Mean joint error distances in z-dimension on RHD and STB datasets.	31
3.9	Comparisons with state-of-the-art methods on RHD [2] and STB [1]. Left: 3D PCK on RHD dataset. Right: 3D PCK on STB dataset. . .	32
3.10	Samples of the generated depth maps by the trained depth regularizer with the input of ground truth 3D hand joint locations.	33
3.11	Visual results of our proposed weakly-supervised approach (column 1,4) and other baselines (column 2,3), compared with ground truth (column 5).	34
3.12	Visual results of our fully-supervised method on RHD and STB datasets. First row: RHD dataset. Second row: STB dataset. Note that skeletons are shown at a novel viewpoint for easy comparison. .	34
3.13	Illustration of the concept of weakly supervised 3D hand pose estimation.	34

3.14	Overview of our proposed weakly-supervised 3D hand pose estimation network.	37
3.15	A pictorial representation of the implemented CVAE-based statistical network, which jointly learns the pose-specific distribution across the paired 3D pose and image feature, from which we sample the latent representation and then decode the predicted 3D hand pose.	38
3.16	Two architectures for using the depth regularizer, either (a) cascaded or (b) paralleled with the 3D hand pose decoder.	41
3.17	Network architecture of our proposed depth regularizer.	42
3.18	Comparisons of 3D PCK results of different baselines with our weakly-supervised method on STB [1].	44
3.19	Comparison of the weakly-supervised method with our conference paper[3] on STB [1] dataset, in scenarios with both depth regularization and 2D supervision (left) or with only depth regularization (right).	45
3.20	Comparisons of the 3D PCK results of different architectures of the depth regularizer for weakly-supervised method. Left: Weakly-supervised flow with only depth regularization. Right:Weakly-supervised flow with both depth regularization and 2D supervision.	45
3.21	Median EPE comparisons with the state-of-the-art methods on STB [1] dataset in supervised and semi-supervised manner as a function of the percentage of labeled data.	46
3.22	Comparisons with the state-of-the-art fully-supervised methods on RHD [2] and STB [1]. Left: 3D PCK on RHD dataset. Right: 3D PCK on STB dataset.	48
3.23	Comparisons with the state-of-the-art methods on Dexter Object [4]. Left: 2D PCK on Dexter Object dataset. Right: 3D PCK on Dexter Object dataset.	48
3.24	2D PCK comparisons with the state-of-the-art methods on Egodexter [5] dataset.	49
3.25	Visual results of our proposed weakly-supervised approach trained with only depth regularization (column 1, 4) and with both 2D labels and depth regularization (column 5), as well as other baselines (non-finetuned approach in column 2, and weakly-supervised method with only 2D supervision in column 3), compared with the ground truth 3D hand pose (column 6).	50
3.26	Visual results of our fully-supervised method on RHD and STB datasets. Row 1-2: RHD dataset. Row 3-4: STB dataset.	51
3.27	Qualitative Results for Dexter Object [4](row 1-2) and Egodexter [5] (row 3-4) datasets for weakly-supervised approach.	51
3.28	Visual results of our proposed CVAE-based statistical framework (column 1, 3) and the original regression framework (column 2) presented in our conference paper [3].	52

3.29	t-SNE visualization of 150 input samples of the encoded features in the latent space. 3D Hand poses are decoded samples drawn from the latent space.	53
3.30	Latent space traversal. The left most and right most 3D poses in each row are reconstruction from the latent space samples of two real RGB images. The figures in-between are decoded from interpolated latent space samples.	54
4.1	Graphical spatial-temporal dependencies between different joints of (a) full human body, and (b) human hand. For easy illustration, we only plot the whole spatial connections on the front frame of the spatial-temporal graph, including the direct physical connections (solid line) and the indirect “symmetrical” relations (dashed curve).	60
4.2	Schematic overview of our proposed network architecture for 3D pose estimation from consecutive 2D poses.	61
4.3	Visualization of different neighboring nodes for (a) human body and (b) human hand.	62
4.4	Illustration of the “local-to-global” network architecture, which is able to effectively process and consolidate features across scales.	65
4.5	The defined hierarchical graph pooling strategy for (a) human body and (b) human hand.	66
4.6	Comparisons of the 3D PCK results with the state-of-the-art methods on STB for 3D hand pose estimation.	70
4.7	The impact of the pose refinement on the mean error distances of different body parts on Human3.6M.	70
4.8	Visual results of our proposed method on Human3.6M and STB datasets. First and second rows: Examples from Human3.6M dataset[6]. Third row: Examples from STB dataset[7]. Forth row: Examples from MPII dataset[8].	74
5.1	Left: Overview of our proposed network architecture for 3D motion prediction. Right: Progressive decoding strategy.	78
5.2	The architecture of the conventional transformer [9], where the far left side is the transformer-encoder that encodes the input into context features $\mathbf{C}$, while the relatively right side is the decoder that recursively generates the output sequence with the encoded context features.	80
5.3	Query and reconstruction procedure in memory-based dictionary for joint j	84
5.4	(a) Qualitative comparison of long-term prediction on Human 3.6M dataset. From top to bottom, we show the ground truth, the results of Residual sup.[10], GCN [11] and our method. (b) Qualitative analysis for the impact of the dictionary. We see that adding the dictionary facilitates more descriptive future dynamics.	89
5.5	Average 3D position error of each predicted frame for all actions on Human 3.6M dataset.	90

6.1	Here we propose a unified framework for various 3D motion synthesis tasks, which can be considered as a generalization of several existing problems such as prediction, completion, interpolation, and spatial-temporal recovery.	96
6.2	Schematic overview of our proposed network architecture.	97
6.3	Illustration of the encoder and decoder architectures, which facilitate the effective process and consolidation of the features across scales. For clarity, we also depict a building block (bottom dashed box) showing the detailed architecture of each decoding stage.	99
6.4	Action-Adaptive Modulation (AAM) is applied to the normalization layers in the decoding stage.	100
6.5	Network Architecture of the AttenNet in Figure 6.3. We first perform self-attention to the decoder features to capture long-term dependencies. Then we apply cross attention to enhance the consistency between the encoder and decoder features.	102
6.6	Qualitative examples of our proposed model for different motion-related tasks on Human3.6M and CMU-Mocap datasets. Gray poses are the pre-defined input frames (or partial body of some frames), while the red & blue skeleton sequences are the synthesized poses. Note that the input constraints can be flexibly set to arbitrary positions with varying densities.	104
6.7	Diverse actions generated from the same initial pose. By setting different semantic guidance, our model generates a variety of semantically meaningful results given the same input frame.	106
6.8	Visualization example of comparisons on Human 3.6M dataset. From top to bottom, we show the result of HP-GAN [12], CSGN [13], our method and ground truth motions.	106
6.9	MPJPE comparison of motion prediction task among different methods.	110

List of Tables

3.1	Impact of our proposed depth regularizer on 3D hand pose estimation with different percentages of labeled data. The Mean EPE (mm) on STB dataset are listed in this table.	51
3.2	Effectiveness of our proposed CVAE-based framework compared with the original regression network in [3] on 3D hand pose estimation with different percentages of labeled data. The Mean EPE (mm) on STB dataset are listed in this table.	51
3.3	Run Time analysis for the proposed 3D Hand pose estimation methods.	57
4.1	Quantitative comparisons of Mean Per Joint Position Error (MPJPE) in millimeter between the estimated pose and the ground-truth on Human3.6M under Protocol #1 and Protocol #2, where T denotes the number of input frames used in each method. The best score is marked in bold	69
4.2	MPJPE Results (in mm) of our method with different input sequence lengths on Human3.6M and STB.	72
4.3	MPJPE Results (in mm) of our method with 3 input frames and different graph convolutional strategies on Human3.6M.	72
4.4	Ablation studies on different components of our network architecture. The evaluation is performed on Human3.6M with the MPJPE metric under Protocol #1.	73
4.5	Run Time analysis for our proposed 3D pose estimation method. . .	75
5.1	Short-term prediction results in Mean Angle Error (MAE) on Human3.6M for all actions. The best result is marked in bold.	88
5.2	Short-term prediction results in Mean Per Joint Position Error (MPJPE) on Human3.6M for all actions. The best result is marked in bold. A 3D suffix to a method indicates that the method was directly trained on 3D joint positions. Otherwise, the results were obtained by converting the joint angle to 3D positions. The best result is marked in bold.	88
5.3	Short and long-term prediction of 3D joint positions in MPJPE on CMU-Mocap dataset.	88
5.4	Long-term prediction of 3D joint positions in MPJPE on Human 3.6M dataset. Our method using 3D coordinates yields the best performance.	90

5.5	Influence of the spatial temporal explorations and the progressive-decoding strategy on 4 actions of Human3.6M. For fair comparison, we exclude the dictionary module for all models.	91
5.6	Impact of the propagating directions on Human3.6 M dataset under the MPJPE protocol. The outward direction performs in the proposed central-to-peripheral extension while the inward direction contrastly propagate the predictions from the outside to inside body.	91
5.7	The impact of the dictionary module on Human3.6M dataset under the MJMPE metric.	91
5.8	Run Time analysis for the proposed 3D pose prediction method.	93
6.1	Classification Accuracy (%) of different action classes on Human 3.6M using a classifier independently trained with ground truth 3D pose series. Our method with both action guidance and Action-Adaptive Modulation (AAM) considerably improves the classification accuracy by a large margin compared with the other two baselines.	107
6.2	Quantitative results on Human3.6M with FID / IS / Diversity metrics to evaluate the long-term generation and MPJPE to assess the completion accuracy. The best results are marked in bold.	108
6.3	Quantitative results on CMU-Mocap dataset. The best results are marked in bold.	108
6.4	Impact of the cross attention mechanism on Human3.6M. For fair comparison, action labels are leveraged for both methods.	108
6.5	Run Time analysis for the proposed 3D pose synthesis method.	111

Summary

Vision-based 3D human and hand pose analysis has been a fast-growing research area and has aroused long-standing research attention in the past decades, since it plays a significant role in numerous applications such as human-computer interactions, robotics, and gesture recognition. Despite the great progress in this field, it is still challenging to obtain accurate 3D pose estimation, predict future motions, and synthesize realistic human behaviors due to the physical complexity of human/hand motion and the lack of high-quality dataset. To address these issues, in this thesis, four chapters are proposed to investigate these tasks.

For 3D pose estimation, I mainly focus on two important aspects, how to alleviate the burden of 3D annotations, and how to better exploit the spatial-temporal correlations of human/hand structure. For the first aspect, different from existing learning-based monocular RGB-input approaches that require accurate 3D annotations for training, I propose to leverage the depth images that can be easily obtained from commodity RGB-D cameras during training, while during testing only RGB inputs are used for 3D joint predictions. In this way, the burden of costly 3D annotations is alleviated for real-world dataset. For the second aspect, motivated by the effectiveness of incorporating spatial dependencies and temporal consistencies, a novel graph-based method is proposed to tackle the problem of 3D human body and 3D hand pose estimation from a short sequence of 2D joint detections. Domain knowledge about the human hand (body) configurations is explicitly incorporated into the graph convolutional operations to meet the specific demand of the 3D pose estimation.

For 3D motion prediction, I aim to capture complicated structures and explore the motion patterns of human behaviors. Specifically, a transformer-based architecture is applied to simultaneously capture the long-range temporal correlations and spatial dependencies. To exploit the kinematic chains of body skeletons, a progressive strategy is deployed, which explicitly decomposes the future joint motion predictions into progressive steps, performed in a central-to-peripheral manner according

to the structural connectivity. To further enable a generalized full-spectrum human motion space across all videos in training data, a memory-based dictionary was proposed to provide auxiliary information to enhance the prediction quality.

For 3D motion synthesis, I aim to find a unified architecture for various 3D motion synthesis tasks, since most existing methods are either restricted to one type of motion synthesis or use different approaches to address various tasks. In particular, I propose a framework based on the Conditional Variational Auto-Encoder (CVAE), where any arbitrary input is treated as a masked motion series. To further allow the flexibility of manipulating the motion style of the generated series, an Action-Adaptive Modulation (AAM) is designed to propagate the given semantic guidance through the whole sequence.

To summarize, this thesis focuses on 3D human and hand pose analysis for images and videos. Novel neural networks are developed to improve the 3d pose estimation accuracy in an end-to-end manner. Meanwhile, motion prediction strategy and a unified motion synthesis model are proposed in this thesis, which significantly contribute to human motion tracking and complex human gesture animations.

Chapter 1

Introduction

1.1 Background and Motivation

Human activity analysis is an important area of computer vision research. It plays an important role in numerous applications such as human-computer interaction (HCI), virtual reality, autonomous driving and computer graphics. In all these applications, analysis of 3D hand and body poses has received increasing attention in the past decade, as it provides rich geometric and motion information for human behavior analysis, whereby computers and robots can identify, locate, and understand actions of people.

There are various tasks with 3D human and hand pose analysis. In this thesis, we focus on three basic ones: estimation, prediction, and synthesis. In particular, 3D pose estimation aims to determine the 3D kinematic configurations of human or hand body parts, which serves as the basis for the other two tasks (as seen in Figure 1.1(a)). Robust and accurate 3D pose estimation will strongly support classifying human actions, making predictions based on observations, and synthesizing the motions of natural human activities under different physical or semantic constraints. 3D human motion prediction (Figure 1.1 (b)) is targeted at forecasting a sequence of future dynamics from an observed series of human poses. Compared with motion prediction, 3D motion synthesis is a more generalized problem, which covers the tasks of motion prediction, completion, interpolation, and spatial-temporal recovery (see Figure 1.2 for more details). Next, a detailed description of each task and our research scope will be provided in this thesis.

FIGURE 1.1: (a) Example of 3D pose estimation. (b) Example of motion prediction

FIGURE 1.2: Example of 3D motion synthesis

1.2 Research Scope

1.2.1 3D Pose Estimation

Articulated pose estimation research has arisen in the past decades [14–16], since it plays a significant role in numerous applications such as human-computer interaction and virtual reality. Although 3D hand pose estimation with depth cameras [17–21] has gained tremendous success in recent years, the monocular RGB-based solution for 3D pose estimation is more favored than depth-based solutions in many

vision applications. In this thesis, we focus on 3D pose estimation from RGB images and videos, which is elaborated in chapter 3 and chapter 4. In particular, chapter 3 aims to alleviate the burden of 3D annotations in real-world datasets. We propose to leverage depth images that can be easily obtained from commodity RGB-D cameras during training, while during testing we will only use RGB inputs for 3D joint predictions. Chapter 4 targets at exploiting the spatial-temporal correlations of human/hand structures for videos. Motivated by the effectiveness of incorporating spatial dependencies and temporal consistencies, we propose a novel graph-based method to tackle the problem of 3D human body and 3D hand pose estimation from a short sequence of 2D joint detections.

1.2.2 3D Motion Prediction

3D motion prediction typically refers to the task of predicting future human motion given a short initial motion segment. It has been extensively studied in the past few years. Despite the great progress in human motion prediction, it remains a challenging task due to the complicated structural dynamics of human behavior. In chapter 5, we address this problem by exploiting the long-range spatial-temporal dependencies and incorporate a general motion space for high-quality prediction.

1.2.3 3D Motion Synthesis

We focus on the generalized problem of 3D motion synthesis that covers the tasks of motion prediction, completion, interpolation, and spatial-temporal recovery in chapter 6. Since these tasks have different input constraints and various fidelity and diversity requirements, most existing approaches only cater to a specific task or use different architectures to address various tasks. To better solve this problem, we propose a unified framework and allow the flexibility of manipulating the motion style of the generated series.

1.3 Thesis Contribution

The main contributions presented in this thesis can be stated as follows:

(i) 3D Hand Pose Estimation from RGB Images

- We innovatively introduce the weakly supervised problem of leveraging low-cost depth maps during training for 3D hand pose estimation from RGB images, which alleviates the burden of 3D joint labeling. To the best of our knowledge, we are the first to introduce such a weakly-supervised setting for 3D hand pose estimation from RGB images.
- We propose an end-to-end learning-based solution for weakly-supervised adaptation from fully-annotated synthetic images to real-world images without requiring complete 3D annotations. In particular, we introduce a depth regularizer supervised by easily captured depth images, which considerably enhances the estimation accuracy compared with other baselines.
- To further exploit the physical structure of 3D hand pose, we present a novel conditional variational auto-encoder(CVAE)-based statistical framework to embed RGB images into a pose-specific subspace, which can then be used to infer the 3D hand joint locations. Experimental results show that the CVAE-based framework is able to improve prediction accuracy and provide a physically valid structure for estimated 3D hand pose.

(ii) 3D Pose Estimation from Videos

- By treating a sequence of skeletons as a spatial-temporal graph, we propose to use GCN to effectively exploit spatial configurations and temporal consistencies for 3D pose estimation, both of which are significant for improving the 3D pose estimation accuracy
- We design a local-to-global network architecture, which is capable of learning multi-scale graph features via successive graph pooling and upsampling layers. Experimental results demonstrate the benefits of such a hierarchical architecture, which can effectively consolidate the local and global features in our network.
- We propose a non-uniform graph convolutional strategy based on generic graph convolutional operations, which learns different convolutional kernel weights for different neighboring nodes according to their semantic meanings.

Experiments show that the proposed graph convolutional strategy is crucial for performance improvement with the constructed sparse spatial-temporal graph for 3D pose estimation.

(iii) 3D Motion Prediction

- We propose to leverage the transformer-based architecture to simultaneously exploit the spatial correlations and the temporal smoothness of human motion, by treating the sequential joints with the encoded temporal features as the input of the network.
- To further exploit the structural connectivity of human skeletons, we deploy a progressive decoding strategy to predict the future joint dynamics in a central-to-peripheral manner in accordance with the kinematic chains of a human body.
- To incorporate the general motion space for high-quality results, we build a memory-based dictionary to guide the predictions, preserving the correspondences between the historical motion features and the representative future dynamics.

(iv) 3D Motion Synthesis

- We propose a unified CVAE-based framework to handle various motion synthesis tasks such as motion prediction, completion, interpolation, and spatial-temporal recovery that have different input constraints and different fidelity and diversity requirements.
- We introduce an Action-Adaptive Modulation (AAM) that is able to control semantic motion styles of the generated series.
- We design a cross-attention mechanism that exploits long-term context information to enhance the realism and global consistency of synthesized sequences.

1.4 Outline of the Thesis

This thesis is organized as follows:

In Chapter 2, we present a detailed literature review of existing works in 3D pose estimation, human motion prediction, and human motion synthesis, respectively.

In Chapter 3, we introduce the weakly-supervised problem for 3D hand pose estimation from RGB images, and then propose two different frameworks, one with a cascaded architecture and one with a Conditional Variational Autoencoder (CVAE) architecture to address this problem.

In Chapter 4, motivated by the effectiveness of incorporating spatial dependencies and temporal consistencies, we propose a graph-based method to tackle the problem of 3D human body and 3D hand pose estimation from a short sequence of 2D joint detections.

In Chapter 5, we present details of a novel approach for human motion prediction, which leverages the transformer-based architecture to simultaneously capture the long-range temporal correlations and spatial dependencies.

In Chapter 6, we investigate the more generalized problem of 3D motion synthesis that covers the tasks of motion prediction, completion, interpolation, and spatial-temporal recovery. Since these tasks have different input constraints and various fidelity and diversity requirements, most existing approaches only cater to a specific task or use different architectures to address various tasks. Here we propose a unified framework based on the Conditional Variational Auto-Encoder (CVAE), where we treat any arbitrary input as a masked motion series.

Chapter 7 summarizes the previous chapters and presents concluding remarks on this thesis. Future research directions are also discussed in this chapter.

To facilitate the reading, Figure 1.3 visualizes the structure of this thesis.

FIGURE 1.3: Outline of this thesis.

Chapter 2

Literature Review

In this chapter, we review existing work in 3D pose estimation, human motion prediction and human motion synthesis respectively.

2.1 3D Pose Estimation

2.1.1 3D Hand Pose Estimation

Articulated 3D hand pose estimation has been studied extensively for a long time, with vast theoretical innovations and important applications. Early work [14, 22, 23] on 3D hand pose estimation from monocular color input used complex model-fitting schemes which require strong prior knowledge on physics or dynamics and multiple hypotheses. These sophisticated methods, however, usually suffer from low estimation accuracy and restricted environments, which result in limited prospects in real-world applications. While multi-view approaches [24, 25] alleviate the occlusion problem and provide decent accuracy, they require sophisticated mesh models and optimization strategies that prohibit them from real-time tasks.

The emergence of low-cost consumer-grade depth sensors in the last few years greatly promotes the research on depth-based 3D hand pose estimation, since the captured depth images provide richer context that significantly reduces depth ambiguity. With the prevailing of deep learning technology[26], learning-based 3D hand pose estimation from single depth images has also been introduced, which can

achieve state-of-the-art 3D pose estimation performance in real-time. In general, they can be classified into generative approaches [27–29], discriminative approaches [17, 18, 21, 30–33] and hybrid approaches [34–37].

Inspired by the great improvement of CNN-based 3D hand pose estimation from depth images[38], deep learning has also been adopted in some recent works on monocular RGB-based applications [2, 3, 39–45]. In particular, Zimmermann and Brox [2] proposed a deep network that learns an implicit 3D articulation prior of joint locations in canonical coordinates, as well as constructing a synthetic dataset to tackle the problem of insufficient annotations. Spurr *et al.* [40] presented a “cross-modal variational model” based on VAE to learn a shared latent space between different modalities. Mueller *et al.* [39] embedded a “GANerated” network which transfers the synthetic images to “real” ones so as to reduce the domain shift between them. To reduce the limitation of depth ambiguities from single-view RGB images, Spurr [46] proposed a set of biomechanical constraints that ensures that the predicted 3D poses are always anatomically plausible during training.

For our proposed depth regularizer in Chapter 3, Dibra *et al.* [47] is the closest work in spirit, which presented an end-to-end network that enables the adaptation from synthetic dataset to unlabeled real-world dataset. However, we want to emphasize that our method is significantly different from [47] in several aspects. Firstly, our work is targeted at 3D hand pose estimation from single RGB input, whereas [47] focuses on depth-based predictions. Secondly, compared with [47] that leverages a rigged 3D hand model to synthesize depth images from 3D hand pose, we use a simple convolutional network to infer the corresponding depth maps.

2.1.2 3D Human Pose Estimation

3D human pose estimation aims to estimate locations of body joints in 3D space. Though multiple viewpoints [48, 49] or other sensors such as depth sensors [50, 51] and IMUs (Wearable Inertial Measurement Units) [52, 53] can achieve accurate results by incorporating rich 3D information, most recent research work [54, 55] focus on the challenging 3D human pose estimation from monocular images or videos, as it is more favored in real world applications.

Thanks to the fast development of deep learning approaches, different aspects of learning-based human pose estimation have been explored in the past few years, which can be roughly classified into two categories: i) directly regressing the 3D locations of each joint from 2D images; ii) decoupling 3d pose estimation into the 2D pose estimation and 3D pose estimation from 2D joint detections.

For the first category, Li and Chan [54] designed a multi-task framework that jointly learns pose regression and body part detectors. Park *et al.* [56] introduced an end-to-end framework with simultaneous training of both 2D joint classification and 3D joint regression. Pavlakos *et al.* [55] introduced a deep convolutional neural network based on the stacked hourglass architecture, with a fine discretization of the 3D space to predict per voxel likelihoods for each joint.

For the second category, Martinez *et al.* [57] directly regressed 3D keypoints from extracted 2D poses via a simple network composed of several fully-connected layers. Zhou *et al.* [58] augmented a state-of-the-art 2D pose estimation sub-network with a 3D depth regression sub-network to predict 3D body pose. To incorporate spatial priors into the framework, Fang [59] developed a deep grammar network to explicitly encode the human body dependencies and relations. Moreover, to deal with the depth ambiguity and visual jitters in static images, Hossain *et al.* [60] utilized the temporal information by propagating joint position information across frames based on a sequence-to-sequence model. The performance gain achieved by these methods motivates us to take a follow-up exploration towards the incorporation of both spatial and temporal dependencies, instead of only focusing on one aspect. In chapter 4, we learn the spatial-temporal information implicitly by combining graph convolutional operations with the domain-specific knowledge for 3D pose estimation.

2.1.3 3D Mesh Reconstruction

To further evaluate the deformable and articulated architecture of 3D human bodies and hands, recently some deep learning-based approaches [61–68] proposed to reconstruct 3D hand mesh from RGB inputs. For hand mesh reconstruction, [63–67] attempted to regress MANO parameters using deep neural networks with the supervisions of silhouette and/or 2D keypoints. Hasson *et al.* [61] presented a learnable model for reconstructing hand and object meshes during manipulations and

exploited a novel contact loss that favors physically plausible hand-object constellations. Ge *et al.* [62] and Kulon *et al.* [69] leveraged graph-based methods to directly operate on the 3D mesh vertices and estimate 3D pose from the generated mesh. For body mesh recovery, the SMPL model [70] has been widely used since it is compatible with existing rendering engines. For instance, [71–73] regressed SMPL parameters to reconstruct 3D human mesh. Instead of predicting SMPL parameters, Zhu *et al.* [74] combined the SMPL model with a hierarchical mesh deformation framework to facilitate the flexibility of 3D body deformation. Kocabas *et al.* [75] included the large-scale motion capture dataset AMASS [76] to discriminate between real human motions and predicted pose. These approaches show good directions on joint 3D pose and shape estimation.

2.2 3D Human Motion Prediction

Human motion prediction aims to forecast a sequence of future dynamics based on an observed series of human poses. It has extensive applications in robotics, computer graphics, healthcare and public safety [77–81], such as human-robot interaction [82], autonomous driving [83] and human tracking [84]. Early approaches tackled this problem with Hidden Markov Model [85], linear dynamics system [86], and Gaussian Process latent variable models [87], *etc.*, which commonly suffer from the computational resources and can be easily stuck in non-periodical actions.

FIGURE 2.1: Example of Motion prediction.

Recently, due to the success of the sequence-to-sequence inference, RNN-based architectures have been widely used in state-of-the-art approaches [12, 88–91]. For instance, Fragkiadaki *et al.* [89] proposed a Encoder- Recurrent-Decoder (ERD) framework, which maps pose data into a latent space and propagates it across the temporal domain through LSTM cells. Martinez *et al.* [10] developed a residual encoder-decoder, which exploits the first-order motion derivatives to alleviate

the jitters between adjacent frames. Interestingly, it shows that a simple zero-velocity baseline, i.e., constantly predicting the last observed pose, led to better performance than directly predicting absolute future poses [89]. To facilitate more realistic human motions, Gui *et al.* [92] introduced an adversarial training and Wang *et al.* [93] employed imitation learning into the sequential modeling. While pushing the boundaries of the motion predictions, many of these RNN-based models directly use a fully-connected layer to learn the representation of human pose, which to some extent overlook the inherent spatial configurations of the human body.

Several recent works [11, 94–99] tried to embed the spatial articulations of human body to enhance the validity of the 3D structures. For example, Jain *et al.* [94] proposed to encode the spatial and temporal structure of the pose via a manually designed spatial-temporal graph. With more structural information incorporated into the network, this approach produces plausible long-term motion for more challenging, locally-periodic actions such as eating and smoking, and does not collapse to unrealistic poses in aperiodic “discussion” sequences. To further resolve the limitation of the fixed graphs and discover long-range interactions between different joints, Mao *et al.* [11] leveraged GNN-based architectures, where all joints are linked together for full explorations.

2.3 Motion Synthesis

Motion synthesis is a more general term that includes not only the motion prediction task but also other motion generation tasks such as motion completion and interpolation. Due to the great demand of producing realistic and plausible human body animations in robotics, computer graphics and computer vision fields, it has been extensively studied in recent years. To synthesize smooth and natural motions, traditional methods [100–102] rely heavily on dense pose specifications and complex optimization, which are time-consuming and expensive to obtain.

Recently, researchers have investigated to synthesize plausible motions with deep learning-based approaches. For short-term motion prediction, some approaches [11, 88, 94–96, 103] deterministically predicted pose sequences by minimizing the MSE loss between ground truth the generated pose series. To produce long-term

robust and dynamic motion synthesis results, generative models [12, 13, 91, 104–108] have been introduced in this task. Barsoum *et al.* [12] combined the Seq2seq framework with a GAN for motion prediction, which is able to generate multiple results by using different latent vectors drawn from a random distribution. Harvey [109] presented a transition generation technique that can serve as a new tool for 3D animators, based on adversarial recurrent neural networks. Henter [110] proposed an autoregressive architecture for generating motion-data sequences based on normalizing flows. Cai *et al.* [111] proposed a two-stage GAN for skeleton motion synthesis, where the first stage initializes the best latent space for the input constraints, and the second stage generates temporal signals represented as latent vector sequences. To allow for bidirectional transforms between the latent and the skeleton spaces, Yan *et al.* [13] proposed a Convolutional Sequence Generation Network (CSGN), which transforms the skeleton sequence from a series of latent vectors sampled from a Gaussian process (GP). Though achieving local coherence in temporal space, we notice that random sampling from a Gaussian process does not fully exploit the global context information, and cannot apply to the spatial-temporal recovery task. To address these issues, in Chapter 6, we introduce a novel CVAE-based framework, which is capable of synthesizing meaningful pose series based on the global context, and can generalize to different tasks, such as motion prediction, completion, and spatial-temporal recovery.

2.4 3D Pose Datasets

2.4.1 Hand Pose Datasets

RHD [2] is a synthetic dataset of rendered hand images with a resolution of 320×320 , which is built upon 20 different characters performing 39 actions and is composed of 41,258 images for training and 2,728 images for testing. All samples are annotated with 2D and 3D keypoint locations. For each RGB image, the corresponding depth image is also provided. This dataset is considerably challenging due to the large variations in viewpoints and hand shapes, as well as the large visual diversity induced by random noise and different illuminations. With all the labels provided, the dataset can be used to train 3D hand pose estimation from monocular RGB images, as we will show in Chapter 3.

STB [1] is a real-world dataset containing two subsets with an image resolution of 640×480 : the stereo subset STB-BB captured from a Point Grey Bumblebee2 stereo camera and the color-depth subset STB-SK captured from an active depth camera. Note that the two types of images are captured simultaneously with the same resolution, identical camera pose, and similar viewpoints. Both STB-BB and STB-SK provide 2D and 3D annotations of 21 keypoints. The dataset is used for 3D hand pose estimation from monocular RGB images and RGB videos, as we will explain in Chapter 3 and Chapter 4.

Dexter+Object [4] & **EgoDexter** [5]. Dexter+Object, in total, provides 6 test video sequences with 3145 frames. All sequences are recorded using a static camera with a single person interacting with an object, where 2D and 3D pose annotations of visible finger-tips are provided for the dataset. Similarly, EgoDexter[5] contains 4 testing video sequences with 3190 frames, which are recorded from egocentric viewpoints with cluttered backgrounds and complex hand-object interactions. 2D and 3D pose annotations of visible fingertips are provided for most frames of this dataset. The two datasets are mainly used to evaluate the robustness of the 3D pose estimation method. (See Section 3.3 for more details.)

2.4.2 Human Pose Datasets

Human3.6M [6] is a large-scale and commonly used dataset for 3D human pose estimation, which consists of 3.6 million images captured from 4 different cameras, with 11 subjects performing a variety of actions, such as “Walking”, “Sitting” and “Smoking”. The 3D pose ground truth and all camera parameters (including intrinsic and extrinsic parameters) are provided in this dataset. In this thesis, we use this dataset for human pose estimation (Chapter 4), motion prediction (Chapter 5) and motion synthesis (Chapter 6) tasks.

CMU-Mocap dataset¹ is a motion capture data of various activities, such as “basketball”, “baseball”, “soccer”, etc. It contains 3D joint locations and we commonly take several motion subjects from the dataset for motion prediction (Chapter 5) and motion synthesis (Chapter 6) tasks.

¹Available at <http://mocap.cs.cmu.edu/>

MPII Dataset [8] is a large scale in-the-wild human pose dataset. The images are collected from on-line videos and annotated by human for 16 2D joints. Since the 3D pose annotations are not provided in MPII dataset, we only use its validation subset for qualitative evaluation for 3D pose estimation. More details can be found in Chapter 4.

2.5 Evaluation Metrics

In this section, we briefly introduce the evaluation metrics that will be used in different tasks.

2.5.1 Metrics for 3D Hand Pose Estimation

We evaluate the 3D hand pose estimation performance with the following metrics. The first metric is the area under the curve (**AUC**) on the percentage of correct keypoints (**PCK**) score, which is a popular criterion to evaluate the pose estimation accuracy with different thresholds, as proposed in [2, 39]. The second and third metrics are the **mean and median end-point-error (EPE)** over all testing frames, as utilized in [40]. Following the same condition used in [2, 39], we assume that the global hand scale and the root depth are known in the experimental evaluations so that we can report PCK curve based on 3D hand joint locations in the global domain, which are computed from the output root-relative articulations. For Dexter Object and Egodexter datasets, we follow [41] to calculate the absolute 3D pose with global scale using the predicted normalized root-relative 3D pose and the intrinsic camera parameters.

2.5.2 Metrics for 3D Human Pose Estimation

We report the mean per joint position error (**MPJPE**) as the evaluation metric, which calculates the average Euclidean distance of the estimated joints to ground truth after the alignment of the root joint (central hip). This protocol is referred to as protocol #1. In some work, an alternative metric is adopted, where the

estimated 3D pose is aligned to the ground truth via a rigid transformation, which is referred as protocol #2.

2.5.3 Metrics for 3D Motion Prediction

The evaluation of 3D motion prediction was performed under two metrics. Following [10, 11, 92, 93, 112], we first report the Euclidean distance between the predicted and the ground-truth joint angles in Euler angle representation, which can be referred to as Mean Angle Error (**MAE**). In [11], an alternative metric of Mean Per Joint Position Error (**MPJPE**) in millimeters is adopted, which is also widely used in 3D pose estimation field [3, 31, 45, 57, 59, 95, 113–115]. Compared with MAE, MPJPE has been noted to be more effective in measuring the predicted human poses due to the inherent ambiguity in angle space, where two different sets of angles can yield the same 3D pose. To show this, we measured the MPJPE in two ways in chapter 5: directly using 3D coordinates to train the network, and converting Euler angles into 3D joint locations.

2.5.4 Metrics for 3D Motion Synthesis

Evaluating the quality of synthesized pose sequences is a difficult problem due to the diverse reasonable possibilities for one masked pose sequence. We would expect the diversity to be smaller with more pre-defined input constraints, and conversely larger when it comes to generating long-range missing parts.

Motivated by this, based on the generation uncertainty of the motion series, we propose two ways to quantitatively evaluate the synthesized samples. Specifically, for completing sequences with pre-defined past and future frames, we assume that one of our 50 synthesized samples will be close to the original ground truth, and select the single sample with the minimum mean per joint position error (**MPJPE**) of the masked places. Practically, we apply a center mask of 30 frames to each sequence for the completion assessment.

Conversely, for long-range generations with very few initial input frames, since the sequences tend to have diverse possible futures in totally different trends, it is not appropriate to directly compare the synthesized results against the one ground

truth. Instead, we mainly focus on whether the results are adequately realistic compared with real data, as well as diverse enough to generate multiple plausible solutions. In particular, extending from the evaluation protocols of the image/video generation problem [111, 116–118], we use three evaluation metrics: the Frechet Inception Distance (**FID**) [119] that measures the statistical distance between the real and synthesized data in the feature space, the Inception Score (**IS**) [120] that analyzes class probabilities for each generated sequence over all classes, and a **Diversity** measure that estimates the feature-based standard deviation of the multiple generated outputs with the same input condition. For fair and complete comparison, FID is further divided into two metrics: **FID_g** that generally assesses the distribution distances between the real and synthesized sequences over the whole dataset, and **FID_a** that calculates the average class-based statistical discrepancies. To get the classification scores and intermediate features, we trained a ST-GCN-based classifier [121] with the ground truth training data. The intermediate feature used for the FID and Diversity scoring is the feature map extracted from the last layer. To precisely analyze the motion patterns within a long-duration skeleton sequence, we cut the synthesized sequence into several short snippets, each of which has a length of 30 frames, for the purpose of quantitative evaluation.

Apart from the above metrics to evaluate the generation quality, we also want to examine whether the synthesized samples are actually residing in the same motion styles with the given action labels. To this end, we employed the pre-trained ST-GCN network [121] to compute the classification accuracy of each action class. Note that the higher the recognition accuracy, the better the generated samples follow the class-specific motion patterns in the real data.

Chapter 3

Weakly-supervised 3D Hand Pose Estimation

3.1 Motivation

Hands are of central importance to humans, since they provide a natural way for humans to manipulate objects and communicate with each other. Articulated hand pose estimation has aroused a long-standing research attention in the past decades [14–16, 34, 122–125], as it serves as an essential component in numerous applications such as human-computer interaction, virtual reality, robotics and rehabilitation. Additionally, similar to human body pose estimation used for action recognition [126–128], 3D hand pose estimation can be further applied to gesture recognition and sign language recognition [17, 129].

Many recent works [17–21, 130–136] on 3D hand pose estimation have gained tremendous success due to the availability of low-cost depth cameras and the large public 3D hand pose datasets [122, 123, 137, 138] with depth images. The advance in monocular RGB-based 3D hand pose estimation[2, 39, 40, 139], however, still remains limited. Due to low cost and low power of RGB cameras, RGB-based solutions for 3D hand pose estimation are more favored than depth-based solutions in many vision applications.

Compared with depth images, single-view RGB images exhibit inherent depth ambiguity and lighting-sensitive appearance, which makes 3D hand pose estimation

from single RGB images a challenging problem. Most recent works on RGB-based 3D hand pose estimation heavily rely on large amount of labeled data for training, while comprehensive real-world dataset with complete 3D annotations is often difficult to obtain, thus limiting the performance. Specifically, compared with 2D annotations, providing 3D annotations for real-world RGB images is typically more difficult since 2D locations can be directly defined in the RGB images while 3D locations cannot be easily labeled by human annotators. To address this problem, Zimmermann *et al.* [2] turned to render low-cost synthetic hands with 3D models, from which the ground truth of 3D joints can be easily obtained. Although achieving good performance on the synthetic dataset, this method does not generalize well to real image dataset due to the domain gap between image features. To tackle this issue, Mueller *et al.* [39] leveraged CycleGANs [140] to generate a “real” dataset transferred from synthetic dataset and combined a CNN with a kinematic 3D hand model for 3D pose estimation. However, its limited performance shows that there still exists gap between generated “real” images and real-world images. Mahdi *et al.* [141] proposed a domain transfer method which learns the feature mapping from color images to depth images. Recently, Spurr *et al.* [40] employed a cross-modal deep variational hand pose estimation, which learns a cross-modal latent representation that are estimated from different modalities. This method, still relying on 3D annotations of real hands, is thus different from our weakly-supervised setting where no 3D annotations of real images are provided.

Most of the previous works [2, 39, 40] for hand pose estimation from real-world single-view RGB images focus on training with complete 3D annotations, which are expensive and time-consuming to obtain, while ignoring the depth images that can be easily captured by commodity RGB-D cameras. In addition, such depth images contain rich cues for 3D hand pose labels, as depth-based methods achieve state-of-the-art performance on 3D pose estimation. Based on these observations, we propose to leverage the easily captured depth images to compensate the scarcity of entire 3D annotations during training, while during testing we take only RGB inputs for 3D hand pose estimation. Specifically, in Section 3.2, we propose a cascaded network architecture including a 2D pose estimation network and a 3D regression network. To to better constrain the physical structure of 3D hand pose, in Section 3.3 we deploy a statistical framework based on Conditional Variational Autoencoder (CVAE) to encode the pose-specific latent subspace, from which we can further decode the 3D hand joint locations.

FIGURE 3.1: Illustration of the concept of weakly supervised 3D hand pose estimation. Different from conventional fully-supervised methods (a) that use 3D labels to guide joint predictions, our proposed weakly-supervised method (b) leverages the reference depth map.

3.2 Cascaded Framework

3.2.1 Introduction

Fig. 3.1 illustrates the concept of our proposed weakly supervised 3D hand pose estimation method, which alleviates the burden of the costly 3D annotations in real-world datasets. In particular, similar to the previous works [58, 142–145] in body pose estimation, we apply a cascaded network architecture including a 2D pose estimation network and a 3D regression network. We note that directly transferring the network trained on synthetic dataset to real-world dataset usually produces poor estimation accuracy, due to the domain gap between them. To address this problem, inspired by [47, 146], we innovate the structure with a depth regularizer, which generates depth images from predicted 3D hand pose and regularizes the predicted 3D regression by supervising the rendered depth map, as shown in Figure 3.13 (b). This network essentially learns the mapping from 3D pose to its corresponding depth map, which can be used for the knowledge transfer from the fully-annotated synthetic images to weakly-labeled real-world images without entire 3D annotations. Additionally, we apply the depth regularizer to the fully-supervised setting. The effectiveness of the depth regularizer is experimentally verified for both our weakly-supervised and fully-supervised methods on two

FIGURE 3.2: We present a weakly-supervised approach for 3D hand pose estimation from monocular RGB-only input.

benchmark datasets: RHD[2] and STB datasets[1].

To summarize, this section makes the following contributions:

- We innovatively introduce the weakly supervised problem of leveraging low-cost depth maps during training for 3D hand pose estimation from RGB images, which releases the burden of 3D joint labeling.
- We propose an end-to-end learning based 3D hand pose estimation model for weakly-supervised adaptation from fully-annotated synthetic images to weakly-labeled real-world images. Particularly, we introduce a depth regularizer supervised by the easily captured depth images, which considerably enhances the estimation accuracy compared with weakly-supervised baselines (see Figure 3.2).
- We conduct experiments on the two benchmark datasets, which show that our weakly-supervised approach compares favorably with existing works and our proposed fully-supervised method outperforms all the state-of-the-art methods.

FIGURE 3.3: Overview of our proposed weakly-supervised 3D hand pose regression network, which is trained in an end-to-end manner. During testing, real images only go through the part of the network in the dashed line box.

3.2.2 Methodology

3.2.2.1 Overview

Our target is to infer 3D hand pose from a monocular RGB image, where the 3D hand pose is represented by a set of 3D joint coordinates $\mathbf{x} = \{x_i\}_{i=1}^J \in \Lambda_{3D}$. Here Λ_{3D} is the $J \times 3$ dimensional hand joint space with $J = 21$ in our case. Figure 3.3 depicts the proposed network architecture, which utilizes a cascaded architecture inspired from [58]. It consists of a 2D pose estimation network (convolutional pose machines - CPM), a 3D regression network, and a depth regularizer. Given a cropped single RGB image containing human hand with certain gesture, we aim to get the 2D heatmap and the corresponding depth of each joint from the proposed end-to-end network. The 2D joint locations are denoted as $\mathbf{x}_{2D} \in \Lambda_{2D}$, where $\Lambda_{2D} \in \mathcal{R}^{K \times 2}$ and the depth values are denoted as $\mathbf{x}_z \in \Lambda_z$, where $\Lambda_z \in \mathcal{R}^{K \times 1}$. The final output 3D joint locations are represented in the camera coordinate system, where the first two coordinates are converted from the image plane coordinates using the camera intrinsic matrix, and the third coordinate is the joint depth. Note that our depth regularizer is only utilized during training. During testing, only 2D estimation network and regression network are used to predict joint locations.

The depth regularizer is the key part to facilitate the proposed weakly supervised training, *i.e.*, relieve the painful joint depth annotations for real-world dataset by

making use of the rough depth maps, which can be easily captured by consumer-grade depth cameras. In addition, our experiments show that the introduced depth regularizer can slightly improve 3D hand pose prediction of fully-supervised methods as well, since it serves as an additional constraint for the 3D hand pose space.

The entire network is trained with a Rendered Hand Pose Dataset (RHD) created by [2] and a real-world dataset from Stereo Hand Pose Tracking Benchmark [1]. For ease of representation, the synthesized dataset and the real-world dataset are denoted as I_{RHD} and I_{STB} , respectively. Note that for weakly-supervised learning, our model is pretrained on I_{RHD} and then adapted to I_{STB} by fusing the training of both datasets. For fully-supervised learning, the two datasets are used independently in the training and evaluation processes.

3.2.2.2 2D Pose Estimation Network

For 2D pose estimation, we adopt the encoder-decoder architecture similar to the Convolutional Pose Machines by Wei *et al.* [147] and [2], which is fully convolutional with successively refined heatmaps in resolution. The network outputs K low-resolution heatmaps. The intensity on each heat-map indicates the confidence of a joint locating in the 2D position. Here we predict each joint by applying the MMSE (Minimum mean square error given a posterior) estimator, which can be viewed as taking the integration of all locations weighed by their probabilities in the heat map, as proposed in [148]. We initialize the network with weights adapted from human pose prediction to I_{RHD} , tuned by Zimmermann *et al.* [2].

To train this module, we employ mean square error (or L2 loss) between the predicted heat map $\hat{\mathbf{x}}_{HM} \in \mathcal{R}^{H \times W}$ and the ground-truth Gaussian heat map $G(\mathbf{x}_{2D}^{gt})$ generated from ground truth 2D labels $\mathbf{x}_{2D}^{gt}$ with standard deviation $\sigma = 1$. The loss function is

$$L_{2D}(\hat{\mathbf{x}}_{HM}, \mathbf{x}_{2D}^{gt}) = \sum_h^H \sum_w^W (\hat{\mathbf{x}}_{HM}^{(h,w)} - G(\mathbf{x}_{2D}^{gt})^{(h,w)})^2. \quad (3.1)$$

3.2.2.3 Regression Network

The objective of the regression network is to infer the depth of each joint from the obtained 2D heatmap. Most previous work [2, 142, 149] in 3D human pose and

hand pose estimation based on single image attempt to lift the set of 2D heatmaps into 3D space directly, while a key issue for this strategy is how to distinguish between the multiple 3D poses inferred from a single 2D skeleton. Inspired from [58], our method exploits contextual information to reduce the ambiguity of lifting 2D heatmaps to 3D locations, by extracting the intermediate image evidence in 2D pose estimation network concatenated with the predicted 2D heatmaps as the input to the regression network. We employ a simple yet effective depth regression network structure with only two convolutional layers and three fully-connected layers. Note that here we infer a scale-invariant and translation-invariant representation of joint depth, by subtracting each hand joint with the location of root keypoint and then normalizing it by the distance between a certain pair of keypoints, as done in [2, 39].

For fully-supervised learning, we simply apply smooth L1 loss introduced in [150] between our predicted joint depth $\hat{\mathbf{x}}_z$ and the ground truth label $\mathbf{x}_z^{gt}$. For weakly-supervised learning, no penalty is enforced because of the absence of 3D annotations. To address this issue, we introduce a novel depth regularizer as weak supervision for joint depth regression, which will be elaborated in Section 3.2.2.4.

Overall, the loss function of the regression network is defined as

$$L_z(\hat{\mathbf{x}}_z, \mathbf{x}_z^{gt}) = \begin{cases} \text{smooth}_{L1}(\hat{\mathbf{x}}_z, \mathbf{x}_z^{gt}) & , if \text{ full supervision} \\ 0 & , if \text{ weak supervision} \end{cases} \quad (3.2)$$

in which

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2, & if |x| < 1 \\ |x| - 0.5, & otherwise. \end{cases} \quad (3.3)$$

3.2.2.4 Depth Regularizer

The purpose of the depth regularizer is to take the easily-captured depth images as an implicit constraint of physical structures that can be applied to both weakly-supervised and fully-supervised situations. Figure 3.4 shows the architecture of the proposed depth regularizer, which is fully-convolutional with six layers, inspired by [146, 151]. Each layer contains a transposed convolution followed by a Relu, after which the feature map is expanded along both image dimensions. In the first five layers, batch normalization [152] and drop out [153] are introduced before

FIGURE 3.4: Network architecture of our proposed depth regularizer.

Relu in order to reduce the dependency on the initialization and alleviate from overfitting the training data. The final layer combines all feature maps to generate the corresponding depth image from 3D hand pose.

Let $(\hat{\mathbf{x}}_{3D}, D)$ denote a training sample, where $\hat{\mathbf{x}}_{3D}$ is the input of the depth regularizer containing a set of 3D hand joint locations, and $\mathbf{D}$ is the corresponding depth image. We normalize $\mathbf{D}$ into $\mathbf{D}_n$:

$$\mathbf{D}_n = \sum_{i,j} \frac{d_{max} - d_{ij}}{d_{range}} \quad (3.4)$$

where d_{ij} is the depth value at the image location (i, j) , and d_{max} and d_{range} represent the maximum depth value and the depth range, respectively. Note that the normalized depth value tends to be larger when located closer to the camera and background is set to 0 in this process.

The input of the network contains two parts: the ground truth 2D labels $\mathbf{x}_{2D}^{gt}$ in the image coordinate system and the joint depth $\mathbf{Z}$. Note that the reason we use ground truth 2D locations rather than our predicted 2D results is to simplify the training process since no back-propagation from the depth regularizer is fed back into the 2D pose estimation network. For the joint depth $\mathbf{Z}$, we apply the same normalization:

$$\mathbf{Z} = \frac{d_{max} - \hat{\mathbf{x}}_z \cdot L_{scale} - d_{root}}{d_{range}} \quad (3.5)$$

where $\hat{\mathbf{x}}_z$ denotes the predicted joint depth from the regression network, which is a set of root-relative and normalized values and can be recovered to global coordinates by multiplying with hand scale L_{scale} and shifting to root depth d_{root} .

To train the depth regularizer, we adopt L1 norm to minimize the difference between the generated depth image $\hat{\mathbf{D}}_n$ and the corresponding ground truth $\mathbf{D}_n$:

$$L_{dep}(\hat{\mathbf{D}}_n, \mathbf{D}_n) = |\hat{\mathbf{D}}_n - \mathbf{D}_n| \quad (3.6)$$

3.2.2.5 Training

Combining the losses in Eq. (3.8), (3.2), and (3.19), we obtain the overall loss function as

$$L = \lambda_{2D} L_{2D}(\hat{\mathbf{x}}_{HM}, \mathbf{x}_{2D}^{gt}) + \lambda_z L_z(\hat{\mathbf{x}}_z, \mathbf{x}_z^{gt}) + \lambda_{dep} L_{dep}(\hat{\mathbf{D}}_n, \mathbf{D}_n). \quad (3.7)$$

Adam optimization [154] is used for training. For weakly-supervised learning, similar to [58] and [155], we adopt fused training where each mini-batch contains both the synthetic and the real training examples (half-half), shuffled randomly during the training process. In our experiments, we adopt a three-stage training process, which is more effective in practice compared with direct end-to-end training. In particular, *Stage 1* initializes the regression network and fine-tunes the 2D pose estimation network with weights from Zimmermann *et al.* [2], which are adapted from the Convolutional Pose Machines [147]. *Stage 2* initializes the depth regularizer, as described in Section 3.2.2.4. *Stage 3* fine-tunes the whole network with all the training data, which is an end-to-end training.

3.2.3 Experiments

3.2.3.1 Implementation Details

Our method is implemented with Pytorch. For the first training stage described in Section 3.2.2.5, we take 60 epochs with an initial learning rate of 10^{-7} , a batch size of 8 and a regularization strength of 5×10^{-4} . For Stage 2 and Stage 3, we spend 40 and 20 epochs, respectively. During the fine-tuning process of the whole network, we set $\lambda_{2D} = 1$, $\lambda_z = 0.1$ and $\lambda_{dep} = 1$. All experiments are conducted on one GeForce GTX 1080 GPU with CUDA 8.0.

FIGURE 3.5: Comparisons of 3D PCK results of different baselines with our method on STB [1].

FIGURE 3.6: Different annotation schemes on RHD [2] and STB [1] dataset

3.2.3.2 Datasets and Metrics

We evaluate our method on two publicly available datasets: Rendered Hand Pose Dataset (RHD) [2] and a real-world dataset from Stereo Hand Pose Tracking Benchmark (STB) [1].

RHD: With all the labels provided, we train the entire proposed network, including the 2D pose estimation network, the regression network and the depth regularizer.

STB: For weakly-supervised experiments, we use color-depth pairs in STB-SK with 2D annotations, as well as root depth (*i.e.*, wrist in the experiments) and hand scale (the distance between a certain pair of keypoints). For fully-supervised experiments, both color-depth pairs (STB-BB) and stereo pairs (STB-SK) with 2D and 3D annotations are utilized to train the whole network. Note that all experiments conducted on STB dataset follow the same training and evaluation protocol used in [2, 39], which trains on 10 sequences and tests on the other two.

We evaluate the 3D hand pose estimation performance with two metrics. The first metric is the area under the curve (AUC) on the percentage of correct keypoints (PCK) score, which is a popular criterion to evaluate the pose estimation accuracy with different thresholds, as proposed in [2, 39]. The second metric is the mean error distance in z-dimension over all testing frames, which is used to further analyse the impact of the proposed depth regularizer. Following the same condition used in [2, 39], we assume that the global hand scale and the root depth are known in the experimental evaluations so that we can report PCK curve based on 3D hand joint locations in the global domain, which are computed from the output root-relative articulations.

3.2.3.3 Quantitative Results

Weak supervision. We first evaluate the impact of weak label constraints on STB dataset compared with fully-supervised methods with complete 2D and 3D annotations. Specifically, we compare our proposed weakly-supervised approach (**w/ 2D + w/ depth regularizer**) with three baselines: a) **w/o 2D + w/o depth regularizer**: directly using pretrained model based on RHD dataset; b) **w/ 2D + w/o depth regularizer**: tuning the pretrained network with 2D labels in STB dataset and c) **w/ 2D + w/ 3D**: fully-supervised method with complete 2D and 3D annotations.

As illustrated in Figure 3.5, the fully-supervised method achieves the best performance while directly transferring the model trained on synthetic data with no adaptation (baseline-a) yields the worst estimation results. This is not surprising, since the fully-supervised method provides the most effective constraint in the 3D

FIGURE 3.7: The effect of the proposed depth regularizer in fully-supervised setting on RHD [2] and STB datasets [1]. Left: 3D PCK on RHD dataset. Right: 3D PCK on STB dataset.

hand pose estimation task and real-world images have considerable domain shift from synthetic ones. Note that these two baselines serve as upper bound and lower bound for our weakly-supervised method. Compared with baseline-a, by fine-tuning the pretrained model with the 2D labels of the real images, baseline-b significantly improves the AUC value from 0.667 to 0.807. Moreover, adding our proposed depth regularizer further increases AUC to 0.889, which demonstrates the effectiveness of the depth regularizer.

We note that STB and RHD datasets adopt different schemes for 2D and 3D annotations, as shown in Figure 3.6. In particular, STB dataset annotates palm position as root joint, which is different from RHD dataset that uses wrist position as root keypoint. Thus, we move the palm joint in STB to wrist point so as to make the annotations consistent for fused training. We extend the line from MCP (Metacarpophalangeal joint) towards palm with a fixed ratio (1:2) to make the distance from palm to wrist equal to that from MCP to palm, as done in [2]. To

FIGURE 3.8: Mean joint error distances in z-dimension on RHD and STB datasets.

evaluate the introduced noise of moving root joint, we compare our results of fully-supervised method on STB dataset with palm-relative and wrist-relative representations. Original palm-relative representation performs slightly better, reducing the mean error by about 0.6mm. Besides, it is also noted that MCP (Metacarpophalangeal joints) positions are closer to wrist joint in STB dataset and labels for STB dataset are relatively noisy compared with synthetic dataset RHD (*e.g.*, thumb dip is annotated in the background). Due to these differences, we argue that there exists a bias between our pose predictions and the ground truth provided by STB dataset, which might decrease the reported estimation accuracy of our proposed weakly-supervised approach. Furthermore, these inconsistencies, on the other hand, suggest the necessity of the introduced depth regularizer, since it provides certain prior knowledge of hand pose and shapes.

Fully-supervised 3D Hand Pose Estimation. We also evaluate the effectiveness of the depth regularizer in the fully-supervised setting on both RHD and STB datasets. Note that the two datasets are trained independently in this case. As presented in Figure 3.7 (left) and Figure 3.7 (right), our fully-supervised method with depth regularizer outperforms that without depth regularizer on both RHD and STB dataset, with improvement of 0.031 and 0.001 in AUC, respectively. Figure 3.8 shows the mean joint error in z-dimension, indicating that adding depth regularizer

FIGURE 3.9: Comparisons with state-of-the-art methods on RHD [2] and STB [1]. Left: 3D PCK on RHD dataset. Right: 3D PCK on STB dataset.

is able to slightly improve the fully-supervised results in the joint depth estimation.

Comparisons with State-of-the-arts. Fig 3.9 shows the comparisons with state-of-the-art methods [1, 2, 39, 40, 42] on both RHD and STB datasets. It can be seen that on RHD dataset, even without the depth regularizer, our fully-supervised method significantly outperforms the state-of-the-art method [2], improving the AUC value from 0.675 to 0.887. On STB dataset, our fully-supervised method achieves the best results compared with all existing methods. Note that our weakly-supervised method is also superior to some of the existing works, which demonstrates the potential values for the weakly-supervised exploration when complete 3D annotations are difficult to obtain in real-world dataset. It is also noted that the AUC values of our proposed methods in Figure 3.9 are slightly different from their counterparts in Section 3.2.3.3. This is because here we test on the stereo pair subset STB-BB rather than the color-depth subset STB-SK.

3.2.3.4 Qualitative Results

Figure 3.11 shows some visual results of our proposed weakly-supervised approach and baselines. For a better comparison, we show the 3D skeleton reconstructions at a novel view and the skeleton reconstructions of our method at the original view are overlaid with the input images. It can be seen that, after additionally imposing the depth regularizer with the reference depth images, our weakly-supervised

FIGURE 3.10: Samples of the generated depth maps by the trained depth regularizer with the input of ground truth 3D hand joint locations.

approach on real-world dataset yields considerably better estimation accuracy, especially in terms of global orientation, which is consistent with our aforementioned quantitative analysis.

Figure 3.12 shows some visual results of our fully-supervised methods on RHD and STB datasets. We exhibit samples captured from various viewpoints with serious self-occlusions. It can be seen that our fully-supervised approach with the depth regularizer is robust to various hand orientations and complicated pose articulations.

Although the depth regularizer is only used in training but not in testing, it is interesting to see whether it has learned a manifold of hand poses. Thus, we collect some samples of the depth images generated by our well trained depth regularizer, given ground truth 3D hand joint locations, as shown in Figure 3.10. We can see that our depth regularizer is able to render smooth and convincing depth images for hand poses in large variations and self-occlusions.

FIGURE 3.11: Visual results of our proposed weakly-supervised approach (column 1,4) and other baselines (column 2,3), compared with ground truth (column 5).

FIGURE 3.12: Visual results of our fully-supervised method on RHD and STB datasets. First row: RHD dataset. Second row: STB dataset. Note that skeletons are shown at a novel viewpoint for easy comparison.

FIGURE 3.13: Illustration of the concept of weakly supervised 3D hand pose estimation.

3.3 Conditional Variational Autoencoder (CVAE)-based Statistical Framework

3.3.1 Introduction

In Section 3.2, we introduce a cascaded framework for weakly-supervised 3D hand pose estimation, which directly takes a regression framework to map 2D to 3D output. To further exploit the physical structure of 3D hand pose, in this section, we propose to learn a pose-specific latent distribution via an extension of Conditional Variational Autoencoder (CVAE) framework, which aims to maximize the posterior probability of 3D hand pose estimation given the RGB image input. Specifically, we provide a framework that allows the training of a latent subspace encoded from the image feature domain and the 3D hand pose domain separately, and force the two embedded latent distributions close to each other. By doing this, the fully trained network model is able to capture the pose-specific latent subspace efficiently and robustly from the RGB images during testing, which can then be used to sample the pose-specific feature and infer the 3D hand joint locations.

For unlabeled real images, directly transferring the network trained on synthetic dataset to real-world dataset usually produces poor estimation accuracy, due to the domain gap between them. To address this problem, inspired by [47, 146], we innovate the structure with a depth regularizer, which generates depth images from the above mentioned pose-specific latent feature and regularizes the latent representation by supervising the generated depth map, as shown in Figure 3.13 (b). This network essentially learns the mapping from the pose-specific latent feature to its corresponding depth map, which can be used for the knowledge transfer from the fully-annotated synthetic images to unlabeled real-world images without entire 3D annotations. The effectiveness of the depth regularizer is experimentally verified by our weakly-supervised, semi-supervised and fully-supervised methods on benchmark datasets.

Compared with the cascaded framework (in Section 3.2) for 3D hand pose estimation from monocular RGB images, we have the following new contributions in this section:

- We replace the original regression network with a novel CVAE-based statistical framework, which encodes the distribution of the pose-specific latent subspace that can be sampled to infer the 3D hand joint locations. Experimental results show that, the CVAE-based framework is able to improve the performance of prediction accuracy and provide a physically-valid structure for 3D hand pose compared with our original regression network.
- To further alleviate the burden of 2D annotation, we investigate our weakly-supervised method trained with only depth regularization, which removes the 2D supervision in the original weakly-supervised approach trained with both 2D labels and depth regularizer, as presented in our conference paper. Moreover, experimental results show that our proposed weakly-supervised method trained with only depth regularizer performs comparably with the original version constrained by both 2D labels and depth regularization.
- We modify the depth regularizer with the input of low-dimensional pose-specific feature instead of the predicted 3D hand pose. Moreover, we experimentally validate that the network can benefit more from constraining the latent representation by the proposed depth regularizer.
- We conduct more extensive experiments including in the semi-supervised setting and self-comparisons. Additionally, we also compare with more state-of-the-art methods on RHD [2], STB [1], Dexter Object[4] and Egodexter[5] datasets. Experimental results show that our method can achieve good performance and has plausible generalization ability in all settings.

3.3.2 Methodology

3.3.2.1 Overview

Figure 3.14 depicts the overview of our proposed network architecture. It consists of a 2D pose estimation network (convolutional pose machines - CPM), a CVAE-based statistic network which jointly learns a pose specific embedding from the 3D pose domain and the image feature domain, from which the 3D joint locations are predicted, as well as a novel depth regularizer to provide a weak supervision. Specifically, given a cropped single RGB image containing human hand with certain gesture, we aim to get the 2D heatmaps and the corresponding depth of each joint

FIGURE 3.14: Overview of our proposed weakly-supervised 3D hand pose estimation network.

from the proposed end-to-end network. We note that although the CVAE-based network is able to decode the 3D hand joint locations and we only choose the z -dimensional result $\mathbf{x}_z$ for our final output, since the 2D projections estimated from the 2D heatmaps slightly outperform those from the CVAE-based network. Moreover, we adopt the implementation of decoding 3D hand joint locations instead of directly regressing the depth of each joint, due to the fact that decoding 3D hand pose in practice brings more robust and accurate estimation of z -dimensional results.

Same as section 3.2, the depth regularizer is the key part to facilitate the proposed weakly supervised training and it is only utilized during training. The entire network is trained with a Rendered Hand Pose Dataset (RHD) created by [2] and a real-world dataset from Stereo Hand Pose Tracking Benchmark (STB) [1]. For ease of representation, the synthesized dataset and the real-world dataset are denoted as I_{RHD} and I_{STB} , respectively. Note that for weakly-supervised learning, our model is firstly pretrained on I_{RHD} and then adapted to I_{STB} by fusing the training of both datasets. For fully-supervised and semi-supervised learning, the two datasets are used independently in the training and evaluation process.

FIGURE 3.15: A pictorial representation of the implemented CVAE-based statistical network, which jointly learns the pose-specific distribution across the paired 3D pose and image feature, from which we sample the latent representation and then decode the predicted 3D hand pose.

3.3.2.2 2D Pose Estimation Network

For 2D pose estimation, we adopt the encoder-decoder architecture similar to the Convolutional Pose Machines proposed by Wei *et al.* [147], which is fully convolutional with successively refined heatmaps in resolution. The network outputs J low-resolution heatmaps. The intensity on each heat-map indicates the confidence of a joint locating in the 2D position. Here we predict the 2D position of each joint by applying the MMSE (Minimum mean square error given a posterior) estimator, which can be viewed as taking the integration of all locations weighed by their probabilities in the heat map, as proposed in [148]. We initialize the network with weights pretrained by Zimmermann *et al.* [2] and finetune them on I_{RHD} .

To train this module, we employ mean square error (or L2 loss) between the predicted heat map $\hat{\mathbf{x}}_{HM} \in \mathcal{R}^{H \times W}$ and the ground-truth Gaussian heat map $G(\mathbf{x}_{2D}^{gt})$ generated from ground truth 2D labels $\mathbf{x}_{2D}^{gt}$ with standard deviation $\sigma = 1$. The loss function is

$$L_{2D}(\hat{\mathbf{x}}_{HM}, \mathbf{x}_{2D}^{gt}) = \sum_h \sum_w^W (\hat{\mathbf{x}}_{HM}^{(h,w)} - G(\mathbf{x}_{2D}^{gt})^{(h,w)})^2. \quad (3.8)$$

3.3.2.3 CVAE-based Statistical Network

After obtaining the 2D joint predictions in the form of heatmaps, we aim to infer the 3D hand pose. Most previous work [2, 142, 149] in 3D human pose and hand pose estimation from a single image attempt to lift the set of 2D heatmaps into 3D space directly, while a key issue for this strategy is how to distinguish multiple 3D poses inferred from a single 2D skeleton. Inspired by [58], our method exploits contextual information to reduce the ambiguity of lifting 2D heatmaps to 3D locations. More specifically, we concatenate the intermediate image evidence extracted from the 2D pose estimation network with the predicted 2D heatmaps, and input the concatenated image features to our CVAE-based statistical network.

Conditional Variational Auto-encoder Given the image features $\mathbf{I}$ containing the 2D heatmaps and the intermediate feature representations, we attempt to estimate the corresponding 3D hand pose $\hat{\mathbf{x}}_p$. Mathematically, our goal is to maximize the posterior probability $p(\hat{\mathbf{x}}_p|\mathbf{I})$.

To address this problem, we resort to a CVAE (conditional variational auto-encoder)[156] based framework, whereby a low-dimensional latent representation $\mathbf{z}_p$ is drawn from the prior distribution $p(\mathbf{z}_p|\mathbf{I})$ and the predicted 3D pose $\hat{\mathbf{x}}_p$ is then generated from the distribution $p(\hat{\mathbf{x}}_p|\mathbf{z}_p, \mathbf{I})$. Similar to the original derivation, we start with the posterior formulation that we wish to maximize:

$$p(\hat{\mathbf{x}}_p|\mathbf{I}) = \int p_\phi(\hat{\mathbf{x}}_p|\mathbf{I}, \mathbf{z}_p) p_\phi(\mathbf{z}_p|\mathbf{I}) d\mathbf{z}_p, \quad (3.9)$$

which can be re-written as follows:

$$\begin{aligned} p(\hat{\mathbf{x}}_p|\mathbf{I}) &= \int p_\phi(\hat{\mathbf{x}}_p|\mathbf{I}, \mathbf{z}_p) \frac{p_\phi(\mathbf{z}_p|\mathbf{I})}{q_\psi(\mathbf{z}_p|\mathbf{x}_p)} q_\psi(\mathbf{z}_p|\mathbf{x}_p) d\mathbf{z}_p \\ &= E_{\mathbf{z}_p \sim q_\psi(\mathbf{z}_p|\mathbf{x}_p)} \left\{ p_\phi(\hat{\mathbf{x}}_p|\mathbf{I}, \mathbf{z}_p) \frac{p_\phi(\mathbf{z}_p|\mathbf{I})}{q_\psi(\mathbf{z}_p|\mathbf{x}_p)} \right\}. \end{aligned} \quad (3.10)$$

After taking logs and applying Jensen's inequality, we obtain the variational lower bound of the conditional log-likelihood of the observation:

$$\begin{aligned}
 \log(p(\hat{\mathbf{x}}_p|\mathbf{I})) &\geq E_{\mathbf{z}_p \sim q_\psi(\mathbf{z}_p|\mathbf{x}_p)}[\log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{I}, \mathbf{z}_p)) \\
 &\quad + \log\left(\frac{p_\phi(\mathbf{z}_p|\mathbf{I})}{q_\psi(\mathbf{z}_p|\mathbf{x}_p)}\right)] \\
 &= E_{\mathbf{z}_p \sim q_\psi(\mathbf{z}_p|\mathbf{x}_p)}\log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{I}, \mathbf{z}_p)) \\
 &\quad - \mathbf{KL}(q_\psi(\mathbf{z}_p|\mathbf{x}_p)||p_\phi(\mathbf{z}_p|\mathbf{I})).
 \end{aligned} \tag{3.11}$$

Here $\mathbf{KL}$ is the Kullback-Leibler divergence, $\mathbf{z}_p$ the pose-specific latent embedding, $\mathbf{x}_p$ the ground truth 3D label, $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ the posterior sampling function, $p_\phi(\mathbf{z}_p|\mathbf{I})$ the conditional prior, and $p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p, \mathbf{I})$ is the likelihood. Note that the conditional probability distributions ψ, ϕ can be parameterized by deep neural networks.

For our purposes, we assume that the low dimensional subspace serves as a controller for the dependency of the high-dimensional 3D hand configurations, which indicates that the extracted pose-specific latent feature $\mathbf{z}_p$ encoded from the 3D annotations $\mathbf{x}_p$ is sufficient to infer the 3D hand pose $\hat{\mathbf{x}}_p$. Therefore, $p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p, \mathbf{I})$ can be replaced by $p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p)$ and Eq. (3.11) can be updated as:

$$\begin{aligned}
 \log(p(\hat{\mathbf{x}}_p|\mathbf{I})) &\geq -\mathbf{KL}(q_\psi(\mathbf{z}_p|\mathbf{x}_p)||p_\phi(\mathbf{z}_p|\mathbf{I})) \\
 &\quad + E_{q_\psi(\mathbf{z}_p|\mathbf{x}_p)}[\log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p))].
 \end{aligned} \tag{3.12}$$

However, the second term in Eq. (3.12) only takes the sampling from $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ during training, which is encoded by the 3D annotations. This may not be optimal to make a prediction during testing, as we wish to reconstruct 3D hand pose typically from the extracted image features $\mathbf{I}$. In order to mitigate the gap between training and testing processes, inspired from [156], we modify Eq. (3.12) by setting the reconstruction with sampling from both $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ and $p_\phi(\mathbf{z}_p|\mathbf{I})$, and we have the following notation:

$$\begin{aligned}
 \log(p(\hat{\mathbf{x}}_p|\mathbf{I})) &\geq -\mathbf{KL}(q_\psi(\mathbf{z}_p|\mathbf{x}_p)||p_\phi(\mathbf{z}_p|\mathbf{I})) \\
 &\quad + \mu E_{q_\psi(\mathbf{z}_p|\mathbf{x}_p)}[\log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p))] \\
 &\quad + (1 - \mu) E_{p_\phi(\mathbf{z}_p|\mathbf{I})}[\log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p))].
 \end{aligned} \tag{3.13}$$

where $0 \leq \mu \leq 1$, which balances the two objectives during training stage. Here we set $\mu = 0.5$ in our experiment.

(a) Cascaded Network Architecture for Pose Estimation and Depth Regularization

(b) Paralleled Network Architecture for Pose Estimation and Depth Regularization

FIGURE 3.16: Two architectures for using the depth regularizer, either (a) cascaded or (b) paralleled with the 3D hand pose decoder.

Network Architecture and Loss function A pictorial representation of the implemented CVAE-based statistical network is provided in Figure 3.15, which consists of two paths. For the upper path, the ground truth 3D label $\mathbf{x}_p$ is leveraged to encode the distribution $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ during training, while for the lower path, which is also the test path, the latent distribution $p_\phi(\mathbf{z}_p|\mathbf{I})$ is inferred based on the extracted image features $\mathbf{I}$. Both $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ and $p_\phi(\mathbf{z}_p|\mathbf{I})$ can be sampled to reconstruct the 3D hand pose.

To train the CVAE-based framework, we propose the loss function with the combination of terms in Eq. (3.13) to maximize the variational lower bound and minimize the total loss:

$$L_{cvae} = L_{KL} + L_{pose}, \quad (3.14)$$

where L_{KL} represents the KL divergence between pairs of latent distributions from image feature domain and 3D pose domain, and L_{pose} denotes the reconstruction loss of the predicted 3D hand pose. Next, we will provide more details on each term, with intuitive interpretations separately.

The distribution regularization L_{KL} aims to regularize consistency between pairs of

FIGURE 3.17: Network architecture of our proposed depth regularizer.

the pose-specific distributions $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ and $p_\phi(\mathbf{z}_p|\mathbf{I})$, in terms of KL divergences, which ensures the latent distribution encoded from image feature domain close to that from the 3D hand pose domain. Practically, to simplify the computation, we define both $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ and $p_\phi(\mathbf{z}_p|\mathbf{I})$ as Gaussian distribution.

$$L_{KL} = \mathbf{KL}(q_\psi(\mathbf{z}_p|\mathbf{x}_p)||p_\phi(\mathbf{z}_p|\mathbf{I})). \quad (3.15)$$

L_{pose} can be interpreted as the likelihood terms in Eq. (3.13), encouraging accurate 3D hand pose reconstructions from 3D pose and RGB image domains. Particularly, for a collection of training datasets, the likelihood terms can be approximated by drawing samples $\mathbf{z}_p^{(l)} (l = 1, 2, \dots, L)$, $\mathbf{z}_p^{(m)} (m = 1, 2, \dots, M)$ from the pose specific distributions $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ and $p_\phi(\mathbf{z}_p|\mathbf{I})$, respectively:

$$E_{q_\psi(\mathbf{z}_p|\mathbf{x}_p)}[\log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p))] = \frac{1}{L} \sum_{l=1}^L \log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p^{(l)})) \quad (3.16)$$

$$E_{p_\phi(\mathbf{z}_p|\mathbf{I})}[\log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p))] = \frac{1}{M} \sum_{m=1}^M \log(p_\phi(\hat{\mathbf{x}}_p|\mathbf{z}_p^{(m)})). \quad (3.17)$$

Then, L_{pose} is defined as:

$$L_{pose} = \mu \text{smooth}_{L1}(\hat{\mathbf{x}}_p^p - \mathbf{x}_p) + (1 - \mu) \text{smooth}_{L1}(\hat{\mathbf{x}}_p^i - \mathbf{x}_p), \quad (3.18)$$

where smooth_{L1} denotes the smooth L1 loss introduced in [150], $\hat{\mathbf{x}}_p^p = \text{Decoder}(\mathbf{z}_p^p)$, $\hat{\mathbf{x}}_p^i = \text{Decoder}(\mathbf{z}_p^i)$, $\mathbf{z}_p^p$ and $\mathbf{z}_p^i$ are the latent features sampled from $q_\psi(\mathbf{z}_p|\mathbf{x}_p)$ and $p_\phi(\mathbf{z}_p|\mathbf{I})$, as shown in Figure 3.15, and $\mathbf{x}_p$ is the ground truth 3D hand pose.

3.3.2.4 Depth Regularizer

For fully-supervised learning with 3D annotations, the above mentioned CVAE-based statistical network is sufficient for both training and testing process, since we use 3D labels to guide the shared latent space and 3D pose predictions. However, for weakly-supervised learning with unlabeled real-world images, no penalty can be enforced because of the absence of 3D annotations.

To address this issue, given the fact that the easily-captured depth images can be applied as an implicit constraint of physical structures, we introduce a novel depth regularizer leveraging low-cost reference depth maps to provide a weak supervision for 3D hand pose. More specifically, inspired by [146, 157] that iteratively synthesize depth images from 3D pose, we deploy a deep neural network to render depth images directly from the corresponding 3D hand pose, which is referred as cascaded depth regularizer, as shown in Figure 3.16 (a). Since 3D hand pose can be decoded from a pose-specific latent feature, we assume that when tuning the pose-specific feature through the depth regularizer, the decoded 3D hand pose can be adapted as well. Moreover, we present another strategy for the network layout, which facilitates the depth regularizer during training by constraining the depth maps generated from the low-dimensional pose-specific features, *i.e.*, the proposed depth regularizer is in parallel with the 3D hand pose decoder, as illustrated in Figure 3.16 (b). In Sec. 3.3.3.4, we experimentally show that our proposed paralleled architecture outperforms the cascaded one, which suggests the benefits of constraining the low-dimensional latent feature.

To train the depth regularizer, we adopt L1 norm which is more robust for image generation task compared with L2 norm, to minimize the difference between the ground truth depth map $\mathbf{D}$ and the depth images $\hat{\mathbf{D}}^i, \hat{\mathbf{D}}^p$ generated from the latent feature $\mathbf{z}_p^i, \mathbf{z}_p^p$, respectively:

$$L_{dep}(\hat{\mathbf{D}}^i, \hat{\mathbf{D}}^p, \mathbf{D}) = \|\hat{\mathbf{D}}^p - \mathbf{D}\|_1 + \|\hat{\mathbf{D}}^i - \mathbf{D}\|_1. \quad (3.19)$$

3.3.2.5 Training

For weakly-supervised learning, similar to [58] and [155], we adopt fused training where each mini-batch contains both the synthetic and the real-world training

FIGURE 3.18: Comparisons of 3D PCK results of different baselines with our weakly-supervised method on STB [1].

examples (half-half), shuffled randomly during the training process.

To train the whole network, for well-labeled synthetic data, we obtain the overall loss function by combining terms in Eq. (3.8), (3.14) and (3.19):

$$L_s = \lambda_{2D}L_{2D} + \lambda_{cvae}L_{cvae} + \lambda_{dep}L_{dep}. \quad (3.20)$$

where L_{2D} is responsible for the prediction of 2D heatmap, L_{cvae} arises from the CVAE-based framework, and L_{dep} is the generation loss from our proposed depth regularizer. In contrast, for real-world data without 3D labels, we only utilize the depth regularizer to provide weak supervision. Therefore, the overall loss function is simplified as:

$$L_r = \lambda_{dep}L_{dep}. \quad (3.21)$$

FIGURE 3.19: Comparison of the weakly-supervised method with our conference paper[3] on STB [1] dataset, in scenarios with both depth regularization and 2D supervision (left) or with only depth regularization (right).

FIGURE 3.20: Comparisons of the 3D PCK results of different architectures of the depth regularizer for weakly-supervised method. Left: Weakly-supervised flow with only depth regularization. Right: Weakly-supervised flow with both depth regularization and 2D supervision.

3.3.3 Experiments

3.3.3.1 Implementation Details

In this section, we describe the details of the network architecture and implementation.

FIGURE 3.21: Median EPE comparisons with the state-of-the-art methods on STB [1] dataset in supervised and semi-supervised manner as a function of the percentage of labeled data.

We apply a simple network in our proposed CVAE-based framework and depth regularizer. More precisely, the encoder for the image features contains two convolutional layers with the kernel size 3, stride 2 and padding 1, followed by a fully connected layer. Moreover, similar to [40], we design the same architecture for the encoder/decoder of 3D hand pose, containing four fully connected layers with 512 hidden units. As for the layout of our proposed depth regularizer, inspired by [146, 151], we deploy a network which passes through a fully-connected layer connected with six convolutional layers, as shown in Figure 3.17. Each convolutional layer contains a transposed convolution followed by a ReLU, after which the feature map is expanded along both image dimensions. In the first five convolutional layers, batch normalization [152] is introduced before ReLU in order to reduce the dependency on the initialization. After that, the final layer combines all feature maps to generate the corresponding depth map.

Our method is implemented within PyTorch. Adam optimizer [154] is used for training. In our experiments, we adopt a three-stage training process, which is more effective in practice compared with direct end-to-end training. In particular, *Stage 1* fine-tunes the 2D pose estimation network with weights from Zimmermann *et al.* [2], which are adapted from the Convolutional Pose Machines [147]. *Stage 2* trains the CVAE-based statistical framework and depth regularizer from scratch with synthetic data. *Stage 3* fine-tunes the whole network with all the training

data, which is an end-to-end training. During optimization, the weights of different losses are set to $\lambda_{2D} = 1$, $\mu = 0.5$, $\lambda_{cvae} = 0.1$ and $\lambda_{dep} = 0.1$, with a batch size of 8 and a regularization strength of 5×10^{-4} . All experiments are conducted on one GeForce GTX 1080 GPU with CUDA 8.0.

3.3.3.2 Datasets and Metrics

We evaluate our method mainly on four publicly available datasets: Rendered Hand Pose Dataset (RHD) [2], real-world dataset from Stereo Hand Pose Tracking Benchmark (STB) [1], Egodexter [5] and Dexter+Object [4] datasets, where Egodexter and Dexter+Object are simply used for validation. Note that here we infer a scale-invariant and translation-invariant representation of 3D hand pose, by subtracting from each hand joint the location of root keypoint and then normalizing it by the distance between a certain pair of keypoints, as done in [2, 39].

RHD: With all the labels provided, we train the entire proposed network, including the 2D pose estimation network, the CVAE-based statistical network and the depth regularizer.

STB: For weakly-supervised experiments, we use color-depth pairs in STB-SK dataset, as well as root depth (*i.e.*, palm in the experiments) and hand scale (the distance between a certain pair of keypoints). For fully-supervised experiments, both color-depth pairs (STB-BB) and stereo pairs (STB-SK) with 2D and 3D annotations are utilized to train the whole network. Note that all experiments conducted on STB dataset follow the same training and evaluation protocol used in [2, 39], which train on 10 sequences and test on the other two.

Dexter+Object [4] & EgoDexter [5]: We note that the two validation datasets contain rich examples of hand-object interactions and egocentric viewpoints, which are hard to find in our training datasets (RHD and STB). To mitigate the gap between training and validation scenarios and improve the robustness of our proposed weakly-supervised model, an extra real-world dataset (CMU Multiview Bootstrapping [158], referred as CMUMB) with only 2D supervision is added to the training process when measuring the estimation performance of the two datasets.

FIGURE 3.22: Comparisons with the state-of-the-art fully-supervised methods on RHD [2] and STB [1]. Left: 3D PCK on RHD dataset. Right: 3D PCK on STB dataset.

FIGURE 3.23: Comparisons with the state-of-the-art methods on Dexter Object [4]. Left: 2D PCK on Dexter Object dataset. Right: 3D PCK on Dexter Object dataset.

We evaluate the 3D hand pose estimation performance with the curve (AUC) on the percentage of correct keypoints (PCK) score, the mean and median end-point-error (EPE) over all testing frames using root-relative articulations. For Dexter Object and Egodexter datasets, we follow [41] to calculate the absolute 3D pose with global scale using the predicted normalized root-relative 3D pose and the intrinsic camera parameters.

FIGURE 3.24: 2D PCK comparisons with the state-of-the-art methods on EgoDexter [5] dataset.

3.3.3.3 Self-Comparison

Weak Supervision. We first evaluate the impact of weak label constraints on STB dataset compared with fully-supervised methods with complete 2D and 3D annotations. Specifically, we compare our proposed weakly-supervised approach (**w/o 2D + w/ depth regularizer**) with four baselines: a) **w/o 2D + w/o depth regularizer**: directly using the network model pretrained on RHD dataset; b) **w/ 2D + w/o depth regularizer**: tuning the pretrained network with 2D labels on STB dataset; c) **w/ 2D + w/ depth regularizer**: tuning the pretrained network with both 2D labels and depth regularizer on STB dataset and d) **w/ 2D + w/ 3D**: fully-supervised method with complete 2D and 3D annotations.

As illustrated in Figure 3.18, the fully-supervised method (baseline-d) achieves the best performance while directly transferring the model trained on synthetic data with no adaptation (baseline-a) yields the worst estimation results. This is not surprising, since the fully-supervised method provides the most effective constraint in the 3D hand pose estimation task and real-world images have considerable domain shift from synthetic ones. Note that these two baselines serve as upper bound and lower bound for our weakly-supervised method, respectively. Compared with baseline-a, by fine-tuning the pretrained model with the 2D labels of the real images (baseline-b), the AUC value significantly increases from 0.249 to 0.467. Moreover,

FIGURE 3.25: Visual results of our proposed weakly-supervised approach trained with only depth regularization (column 1, 4) and with both 2D labels and depth regularization (column 5), as well as other baselines (non-finetuned approach in column 2, and weakly-supervised method with only 2D supervision in column 3), compared with the ground truth 3D hand pose (column 6).

as shown in Figure 3.18, adding our proposed depth regularizer without 2D supervision (our proposed weakly-supervised approach) achieves better performance than baseline-b, increasing AUC to 0.475, which suggests that the depth regularizer provides even stronger supervision than 2D labels. Furthermore, baseline-c with both 2D labels and depth regularizer further improves the AUC value to 0.558, demonstrating that the network can benefit from the complementarity between depth regularizer and 2D supervision.

We also compare the weakly-supervised method with our conference paper[3]. As shown in Figure 3.19, our proposed model surpasses the accuracy of [3], which improves the AUC value from 0.796 to 0.881 for weak-supervised method with only depth regularizer. Moreover, in scenarios with both 2D labels and depth regularization, our proposed method outperforms [3] as well, increasing the AUC value from 0.876 to 0.926. It is worth noting that our proposed weakly-supervised method with only depth regularization (AUC value: 0.881) even performs comparably with the original method [3] with both 2D labels and depth regularization (AUC value: 0.876), which further indicates the advanced generalization ability of the proposed pipeline.

Semi-supervision. To further assess the effectiveness of our proposed depth regularizer, we also explore the semi-supervised situation, where both labeled data with 3D annotations and unlabeled data with depth maps are available during training process. This is a common scenario, as unlabeled RGB images with depth maps can be easily captured by commodity RGB-D cameras.

FIGURE 3.26: Visual results of our fully-supervised method on RHD and STB datasets. Row 1-2: RHD dataset. Row 3-4: STB dataset.

FIGURE 3.27: Qualitative Results for Dexter Object [4](row 1-2) and Egodexter [5] (row 3-4) datasets for weakly-supervised approach.

TABLE 3.1: Impact of our proposed depth regularizer on 3D hand pose estimation with different percentages of labeled data. The Mean EPE (mm) on STB dataset are listed in this table.

	0%	5%	10%	50%	75%	100%
w/o depth reg.	30.18	13.95	11.61	8.11	7.58	7.12
w/ depth reg.	17.61	13.33	11.36	7.80	7.30	7.10

TABLE 3.2: Effectiveness of our proposed CVAE-based framework compared with the original regression network in [3] on 3D hand pose estimation with different percentages of labeled data. The Mean EPE (mm) on STB dataset are listed in this table.

	5%	10%	50%	75%	100%
[3] regression network	14.25	12.22	8.37	8.13	7.73
proposed CVAE framework	13.95	11.61	8.11	7.58	7.12

Table 3.1 compares the results of the experiments with different percentages of 3D joint labels in STB dataset. We see that with more unlabeled data used in the dataset, our proposed depth regularizer leads to larger gain in 3D hand pose estimation, which indicates the regularizing effect of the additional depth maps. Specifically, we note that for fully-supervised method (100% in Table 3.1), the individual performance difference brought by our proposed depth regularizer is minor, while on weakly-supervised setting (0% in Table 3.1), the margin is much

FIGURE 3.28: Visual results of our proposed CVAE-based statistical framework (column 1, 3) and the original regression framework (column 2) presented in our conference paper [3].

more significant, up to 12.57mm. This can be expected, as we already leverage the strong supervision of 3D annotations in fully-supervised method, which represents much more accurate 3D hand pose information than depth maps. In contrast, for weakly-supervised task, the complementary depth information becomes significant for closing the large domain shift between synthetic and real-world dataset, so as to considerably improve the performance in a relatively large margin.

CVAE-based framework Versus Simple Regression Network. We also compare the influence of our proposed CVAE-based statistical framework with the simple regression network presented in our conference paper [3], which is illustrated in Table 3.2. For fair comparison, we do not use depth regularizer in our experiment. Therefore, Mean EPE of the weakly-supervised method (0% of labeled data) is not reported in the table. It is worth noting that, when using CVAE-based framework, our method outperforms the original pipeline in both semi-supervised and fully-supervised method (100% of labeled data), as shown in Table 3.2. In Section 3.3.3.6, we show extensive qualitative comparisons for the weakly-supervised flow with only 2D supervision, which demonstrates the advantages of the CVAE-based framework in producing plausible physical structures of 3D hand pose.

3.3.3.4 Cascaded Depth Regularizer Versus Paralleled one

In this section, we investigate the prediction accuracy of our proposed depth regularizer operated on pose-specific latent representation (paralleled network architecture) or directly on 3D pose estimation (cascaded network architecture), as described in Sec. 3.3.2.4. For fair comparison, as shown in Figure 3.16, we experiment with similar network architecture, where the only difference is the layout of

FIGURE 3.29: t-SNE visualization of 150 input samples of the encoded features in the latent space. 3D Hand poses are decoded samples drawn from the latent space.

the depth regularizer, either cascaded or paralleled with the 3D hand pose decoder. Figure 3.20 reports the comparative results for weakly-supervised method on STB dataset, which shows that the paralleled depth regularizer is superior to the cascaded one, indicating that the network can benefit more from directly constraining the low-dimensional pose-specific latent feature, compared with constraining the 3D hand pose.

3.3.3.5 Comparison with State-of-the-art Methods

Fully-supervised and Weakly-supervised Methods. Figure 3.22 shows the comparisons with state-of-the-art methods, including PSO [27], ICPPSO [159], Panteleris *et al.* [42], Zimmermann *et al.* [2], Mueller *et al.* [39], Spurr *et al.* [40], Iqbal *et al.* [41], Yang *et al.* [160] and our conference paper [3] on both RHD and STB datasets. Note that here we report the performance of our fully-supervised method without depth regularizer.

As shown in Figure 3.22 (left), on RHD dataset, our fully-supervised method is superior to the state-of-the-art methods. The AUC value of our proposed method is 0.24, 0.066, 0.059 and 0.059 higher than those of the methods in [2, 3, 40, 160], respectively. On STB dataset, our fully-supervised method achieves the best results

FIGURE 3.30: Latent space traversal. The left most and right most 3D poses in each row are reconstruction from the latent space samples of two real RGB images. The figures in-between are decoded from interpolated latent space samples.

compared with all existing methods, improving the AUC value to 0.996 in joint error range between 20mm and 50mm. Note that our weakly-supervised method with only depth regularizer also outperforms some of the existing fully-supervised methods, which demonstrates the potential values for the weakly-supervised exploration when complete 3D annotations are difficult to obtain in real-world dataset. We would like to point out that the gap between the weakly-supervised and fully-supervised approaches is partially from the skeleton differences between synthetic and real data, as different annotation schemes are adopted by the synthetic RHD [2] and real-world STB[1] datasets.

To further assess the generalization quality of our proposed weakly-supervised approach, we additionally provide quantitative analysis on Dexter Object[4] and Egodexter[5] datasets, as shown in Fig 3.23 and Fig 3.24. Note that different from other state-of-the-art approaches [2, 39, 41, 63, 67] mentioned in Fig 3.23 and Fig 3.24, our proposed weakly-supervised method didn't take real-world 3D labels of other datasets as supervision during training. We can see that our weakly-supervised approach is comparable with other fully-supervised approaches with real-world 3D annotations, which indicates the generalization ability of our approach to real-world scenarios.

Semi-supervised Methods. We also perform a quantitative analysis of our semi-supervised method on STB dataset with the state-of-the-art semi-supervised

methods[3, 40] for RGB-based 3D hand pose estimation. As presented in Figure 3.21, our proposed semi-supervised approach without depth regularizer outperforms [40] by a relatively large margin. Additionally, the proposed CVAE-based model consistently surpasses our conference paper[3], which validates that our proposed CVAE-based statistical framework can better capture the 3D pose information and provide more accurate estimation, compared with the original regression network. Furthermore, Figure 3.21 also shows that adding depth regularizer can further improve the estimation accuracy of the semi-supervised method, as discussed in Section 3.3.3.3.

3.3.3.6 Qualitative Results

Figure 3.25 shows some visual results of our proposed weakly-supervised approach and baselines. For a better comparison, we show the 3D skeleton reconstructions at a novel view and the skeleton reconstructions of our method at the original view are overlaid with the input images. It can be seen that, directly applying the model pretrained on synthetic dataset to real dataset (column 2) fails to capture the global orientation of 3D pose, while finetuning with 2D labels improve this situation to some extent, as shown in column 3. It is worth noting that after simply imposing the depth regularizer without 2D supervision (column 1, 4), both the 2D estimation results and global orientations considerably perform better, compared with the non-finetuned method in column 2. Finally, with both 2D constraint and depth regularizer, the weakly-supervised approach on real-world dataset yields the best estimation accuracy, which is consistent with our aforementioned quantitative analysis.

Figure 3.26 shows some visual results of our fully-supervised methods on RHD and STB datasets. We exhibit samples captured from various viewpoints with serious self-occlusions. It can be seen that our fully-supervised approach is robust to various hand orientations and complicated pose articulations.

To evaluate the generalization ability of our proposed weakly-supervised method, we also provide qualitative examples on another two challenging datasets (Egodexter[5] and Dexter Object[4]), as shown in Figure 3.27. As can be seen in Figure 3.27, our method is able to provide valid pose estimation for challenging scenarios in daily life.

We also conduct qualitative comparison between our proposed CVAE-based statistical framework and the regression network in our original paper[3]. For a fair comparison, we train the network on STB dataset with only 2D supervision. Note that no depth regularizer is introduced in this setting. Qualitative results are provided in Figure 3.28. As can be seen, although our proposed method did not accommodate well with the global orientation due to the absence of depth supervision, it still performs better in exploiting plausible and valid 3D pose structures than the original simple regression network.

The main benefit of modelling uncertainty in VAE-type model is to utilize a smooth, continuous, and consistent low-dimensional sub-space directly for posterior estimation in 3D hand-pose estimation. To show the distribution of the latent space, we use t-SNE to plot the latent space features, which is shown in Figure 3.29. As we can see, the latent space is relatively smooth and similar poses can be clustered together in the latent space.

To better understand the smoothness and consistency of the latent space, we also perform a walk on the latent space samples. Specifically, given the latent features from two RGB images, we interpolate the latent space between them and decode the interpolated latent features into 3D poses. As we can see in Figure 3.30, the generated 3D poses move smoothly and consistently across the interpolated latent space. This demonstrates that the learned latent space is smooth and represents a valid statistical manifold of 3D hand poses.

3.4 Runtime Analysis

We compare the time latency for the two proposed weakly-supervised 3D hand pose estimation with a single 1080TI in the following table. STB dataset is used in this experiment. We see that our method is capable of running in real-time at over 62 frames per second (fps) for the cascaded framework, and 55 frames per second (fps) for the CVAE Framework.

Method	Train Run Time (ms)	Inference Run Time (ms)	MPJPE (mm)
Cascaded Framework	73	16	16.89
CVAE Framework	81	18	15.13

TABLE 3.3: Run Time analysis for the proposed 3D Hand pose estimation methods.

3.5 Conclusion

Providing full 3D annotations for a large real-world hand dataset is a major bottleneck for learning-based 3D hand pose estimation. To address this problem, we have presented a weakly supervised solution, which can leverage unlabeled real-world dataset by adapting fully-annotated synthetic dataset with the aid of low-cost depth images, which, to our knowledge, is the first exploration of leveraging depth maps to compensate the absence of entire 3D annotations for 3D hand pose estimation from RGB images. To be specific, we have introduced a novel and effective end-to-end architecture consisting of a 2D estimation network, a CVAE-based statistical network, and a novel depth regularizer. The results show that applying depth regularization significantly improves the estimation accuracy of global orientation of the 3D poses (as can be seen in Figure 3.25), as it provides additional depth information for the 3D surface. Compared with the cascaded framework, the proposed CVAE-based network is able to capture the pose-specific latent subspace efficiently and robustly from the RGB images, which further improves the accuracy of the 3D pose estimation. For the weakness of this work, we note that the proposed model may fail for heavy occlusion cases, especially for object occlusions, since the network has no prior knowledge about specific objects and the contact surface of hand-object interactions. For wider usage of the weakly-supervised 3D pose estimation, this will be a worth-exploring direction for future study.

Chapter 4

Exploiting Spatial-temporal Relationships for 3D Pose Estimation

4.1 Introduction

In Chapter 3, we introduce a weakly supervised problem of leveraging low-cost depth maps during training for 3D hand pose estimation from RGB images. The proposed 3D estimation pipeline relies on effective 2D pose estimation frameworks to first localize the 2D keypoints on the image plane, and then lift 3D poses from the estimated 2D joint positions. However, it is worth noting that estimating 3D poses from 2D representations is inherently an ill-posed problem, since there may exist multiple valid 3D interpretations for a single 2D skeleton. This makes it difficult to infer a unique valid solution, especially for cases with severe occlusions.

To overcome this ambiguity, several methods [59, 161, 162] attempted to embed kinematic correlations to ensure the spatial validity of the 3D structures. For instance, Fang *et al.* [59] explicitly incorporated geometric dependencies among different body parts by enforcing spatial consistency over the estimated 3D human poses. Moreover, to deal with the incoherent and jittery predictions, some work [60, 163, 164] turned to exploit the temporal information across sequences. For example, Hossain *et al.* [60] designed a sequence-to-sequence network to predict

FIGURE 4.1: Graphical spatial-temporal dependencies between different joints of (a) full human body, and (b) human hand. For easy illustration, we only plot the whole spatial connections on the front frame of the spatial-temporal graph, including the direct physical connections (solid line) and the indirect “symmetrical” relations (dashed curve).

3D joint locations and imposed temporal smoothness constraints during training to ensure the temporal consistency over a sequence.

Despite their promising results, we observe that most of the existing work only focus on incorporating either spatial configuration constraints or temporal correlations, while ignoring the complementarity between these two types of information. More precisely, we note that having priors on the spatial dependencies can reduce the possibility of generating physically impossible 3D structures and alleviate the problem of self-occlusions, while utilizing temporal inference helps resolve the challenging issues such as depth ambiguity and visible jitters. These observations encourage us to develop a method that can effectively embed both spatial and temporal relationships into a learning-based framework, and leverage it for 3D pose estimation.

Motivated by the natural graph-based representation for a series of skeletal forms and inspired by recent advances in graph convolution networks (GCNs) [121, 165–167], in this chapter, we propose to utilize GCNs to exploit spatial and temporal relationships for 3D pose estimation. Note that different from the two recent papers [62, 97] that either uses uniform GCN for dense hand mesh reconstruction or considers spatial graph-lstm, our work uses GCN for spatial-temporal graph with semantic grouping for sequential 3D pose estimation. Specifically, as depicted in Figure 4.1, we define the sequence of skeletal joints as a spatial-temporal graph.

FIGURE 4.2: Schematic overview of our proposed network architecture for 3D pose estimation from consecutive 2D poses.

The graph topology is formed with joints as the graph nodes, linked by two types of connections: spatial edges that represent spatial dependencies among different joints, and temporal edges that connect the same joint across neighboring frames. To deal with sparse connections and functionally-variant graph edges for 3D pose estimation, we propose to learn different convolutional kernel weights for different neighborhood types, while the generic graph convolutional operations uniformly treat the neighboring nodes at the same degree with shared kernel weights.

Moreover, inspired by the previous 2D pose estimation approach [168] that processed and consolidated information at multiple resolutions, we analogously propose a graph-convolutional “local-to-global” hierarchical network architecture that captures multi-scale features, where our graph pooling and upsampling layers are designed based on the interpretable human body (or hand) configurations. Finally, a pose refinement step is introduced to further improve the estimation accuracy (see Figure 4.2 for system overview). The contributions of this work are threefold:

- By treating a sequence of skeletons as a spatial-temporal graph, we propose to use GCN to effectively exploit the spatial configurations and temporal consistencies for 3D pose estimation, both of which are significant for improving the 3D pose estimation accuracy.
- We design a local-to-global network architecture, which is capable of learning multi-scale graph features via successive graph pooling and upsampling layers. Experimental results demonstrate the benefits of such hierarchical architecture that can effectively consolidate the local and global features in our network.

FIGURE 4.3: Visualization of different neighboring nodes for (a) human body and (b) human hand.

- We propose a non-uniform graph convolutional strategy based on the generic graph convolutional operations, which learns different convolutional kernel weights for different neighboring nodes according to their semantic meanings. Experiments show that the proposed graph convolutional strategy is crucial for performance improvement with the constructed sparse spatial-temporal graph for 3D pose estimation.

We conduct comprehensive experiments on two widely-used benchmarks: the Human3.6M dataset [6] for 3D human body pose estimation and the STB dataset [7] for 3D hand pose estimation. Experimental results show that our proposed method achieves state-of-the-art performance on both tasks.

4.2 Methodology

4.2.1 Overview

Figure 6.2 depicts an overview of our proposed network architecture. Given a small number of adjacent 2D joint locations of a hand (or body) estimated from video frames as input, we aim at predicting a target frame’s 3D joint locations $\mathbf{x} = \{x_i\}_{i=1}^J \in \Lambda_{3D}$ in the camera coordinate system, where J is the number of

joints, and $\mathbf{A}_{3D}$ is the $J \times 3$ dimensional hand joint space. In particular, we construct a spatial-temporal graph with the joints as graph nodes and the local connectivities in the spatial (skeleton structure) and temporal domains as graph edges. To effectively learn the multi-scale features of the graph-based representation, a hierarchical “local-to-global” scheme is introduced into the framework, which takes successive steps of pooling and upsampling before generating the 3D predictions. Lastly, a pose-refinement process is added to further refine the 3D pose estimation. The whole model is trained in an end-to-end manner with backpropagation. Next, we will describe the individual components in details.

4.2.2 Spatial-temporal Graph Construction

A skeleton sequence can be naturally organized as a spatial-temporal graph representation. Specifically, we define a pose sequence as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{V} = \{v_{ti} | t = 1, \dots, T; i = 1, \dots, J\}$ denotes a set of vertices, corresponding to T frames and M body joints per frame, $\mathcal{E} = \{e_{ij}\}$ is the set of edges, indicating the connections between nodes, and $W = (w_{ij})_{N \times N}$ with $N = JT$ is the adjacency matrix, with $w_{ij} = 0$ if $(i, j) \notin \mathcal{E}$, and $w_{ij} = 1$ if $(i, j) \in \mathcal{E}$. The normalized graph Laplacian [169] is computed as $L = I_N - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, where $D^{ii} = \sum_j W^{ij}$. The edge set consists of two parts: temporal connections that link each joint with its counterpart in the neighboring frames, and spatial connections that include both direct and indirect kinematic dependencies in each frame (see Figure 4.1).

4.2.3 Revisiting Graph Convolutional NNs

In this work, we adopt a spectral-based GCN, since it works well with structured graphs with predefined topology. In particular, the spectral convolutions on graphs [167] can be considered as the multiplication of a signal $h \in \mathbb{R}^N$ with a filter $g_\theta = \text{diag}(\theta)$ in Fourier domain:

$$g_\theta * h = U g_\theta U^T h, \quad (4.1)$$

where graph Fourier basis U is the matrix of the eigenvectors of the normalized graph Laplacian L , and $U^T h$ denotes the graph Fourier transform of h .

To reduce the computational complexity, Kipf and Welling [166] introduced a layer-wise linear formulation defined by stacking multiple localized graph convolutional layers with the first-order approximation of graph Laplacian:

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{W} \tilde{D}^{-\frac{1}{2}} H \Theta, \quad (4.2)$$

where the input signal $H \in \mathbb{R}^{N \times C}$ is a generalized one, representing the C -dim features of N vertices on the graph, $\Theta \in \mathbb{R}^{C \times F}$ is the matrix of filter parameters, $\tilde{W}$ and $\tilde{D}$ are the normalized versions with $\tilde{W} = W + I_N$ and $\tilde{D}^{ii} = \sum_j \tilde{W}^{ij}$, and $Z \in \mathbb{R}^{N \times F}$ is the convolved signal matrix.

4.2.4 Graph Convolution for Pose Estimation

In the existing graph convolution (Eq. (4.2)), essentially each kernel Θ is shared by all the 1-hop neighboring nodes. This works fine for dense graph. However, our spatial-temporal graph for 3D pose estimation is sparse with functionally-variant graph edges (*e.g.*, spatial edges and temporal edges representing different correlations), for which a uniform treatment of neighboring nodes is not suitable.

To tackle this issue, inspired from the previous studies [121, 170] that take the convolutional operator with a larger kernel size, we made modifications to the generic graph convolutional operations. In particular, we classify neighboring nodes according to their semantic meanings and use different kernels for different neighboring nodes. As presented in Figure 4.3, the neighboring nodes are divided into six classes based on intuitive interpretations: 1) the center node itself; 2) a physically-connected neighboring node that is closer to the root node than the center node; 3) a physically-connected neighboring node that is farther from the root node than the center node; 4) an indirect “symmetrically-related” neighboring node; 5) a time-forward neighboring node; and 6) a time-backward neighboring node. Based on the classification, the graph convolution in (4.2) is updated to:

$$Z = \sum_k D_k^{-\frac{1}{2}} W_k D_k^{-\frac{1}{2}} H \Theta_k, \quad (4.3)$$

where k is the index of the neighbor types, and Θ_k is the filter matrix for the k -th type 1-hop neighboring nodes. Note that here $\tilde{W}$ is dismantled into k sub-matrices with $\tilde{W} = \sum_k W_k$, and $D_k^{ii} = \sum_j W_k^{ij}$.

FIGURE 4.4: Illustration of the “local-to-global” network architecture, which is able to effectively process and consolidate features across scales.

4.2.5 GCN-based Local-to-global Prediction

A design choice that has been particularly effective for pose estimation is capturing visual patterns or semantics at different resolutions in a feed-forward fashion. Bottom-up processing is first performed by subsampling the feature maps, and then top-down processing is conducted by upsampling the feature maps with the combination of higher resolution features from bottom layers, as proposed in the Stacked Hourglass network [168] for 2D pose estimation. Inspired by the success of such hierarchical architectures, we propose a conceptually similar “local-to-global” scheme, which aims at learning multi-scale features but from the graph-based representations.

Graph Pooling and Upsampling For graph-based representations, the pooling operation requires meaningful neighborhoods on graphs, where similar vertices are clustered together. In this work, we propose to gradually cluster the whole skeleton per frame based on interpretable human body (or hand) configurations, as specified in Figure 4.5. We use maxpooling during implementation.

For the top-down process, the upsampling procedure simply takes a reverse step of the graph pooling procedure, where the features of vertices in the coarser graph are duplicated to the corresponding child vertices in the finer scale. In addition, the temporal links remain the same throughout the different abstraction levels, connecting each node with its counterparts in neighboring frames.

Hierarchical Architecture: Figure 4.4 shows the proposed hierarchical “local-to-global” network, which can effectively process and consolidate features across

FIGURE 4.5: The defined hierarchical graph pooling strategy for (a) human body and (b) human hand.

scales. In the earlier stage, we gradually perform the graph convolution and pooling operations from the original scale to a very low resolution. Thereafter, the network conducts a top-down process with a sequence of upsampling and combining of features across scales. To utilize both bottom-up and top-down features, we perform an element-wise concatenation for features with the same scale, followed by a per-node FC layer to update the combined features. Furthermore, a non-local block [171] is introduced before generating the 3D pose sequences to facilitate a holistic processing of the full body.

4.2.6 Pose Refinement

For the 3D pose estimation task, there are two types of widely-used 3D pose representations. The first uses root-relative 3D coordinates of the joints in the camera coordinate system, while the second involves concatenating the predicted depths of each joint and the UV coordinates extracted from 2D detectors. These two representations can be easily converted from one to the other using the camera intrinsic matrix.

For relatively accurate 2D pose, the second representation is preferred since it guarantees the consistency between the predicted 3D pose and the 2D projections on the image plane. However, for poor 2D pose, maintaining the consistency between

the projections and the 3D pose often leads to a physically invalid 3D pose structure; here the first representation is better as it is more capable of generating a valid 3D pose structure. To strike a balance between the two circumstances, we design a simple two-layer fully-connected network for pose refinement, which takes the 3D pose estimation results in both representations (where the depth values in the second representation are directly computed from the first) as the input, and output the confidence values for the two sets of results. Finally, the refined 3D joint locations are computed as the confidence-weighted sum of the two sets of estimation results. The confidence score is not supervised during training, which is regularized by minimizing the differences between the refined 3D joint locations and ground truth 3D poses.

4.2.7 Training

We use the following losses in training.

3D Pose Loss. $L_p = \sum_{t=1}^T \sum_{i=1}^J \|\hat{x}_{t,i} - x_{t,i}\|_2^2$, where $\hat{x}_{t,i}$ and $x_{t,i}$ represent the estimated and ground truth 3D joint locations of joint i at time t , respectively

Derivative Loss. Similar to [60], we adopt a derivative loss L_d to enforce temporal smoothness. Considering that joints located at limb terminals commonly move faster than other joints, we divide the joints of a human body into three sets: torso head, limb mid and limb terminal, while for the human hand we divide the 21 joints into: palm root, finger mid and finger terminal. Mathematically, the derivative loss L_d is defined as

$$L_d = \sum_{t=2}^T \sum_{i=1}^J \sum_{s \in S} \eta_s \|\hat{x}_{t,i}^s - \hat{x}_{t-1,i}^s\|_2^2, \quad (4.4)$$

where $\hat{x}_{t,i}^s$ denotes the predicted 3D locations of joints belonging to the set s , and η_s is the scalar hyper-parameter controlling the significance of each set, where a higher value is assigned to the set of joints that are generally more stable than others.

Symmetry Loss L_s . It is defined for penalizing the differences in the lengths of left and right bone pairs, as is typically employed in 3D body pose estimation.

Mathematically, L_s can be written as

$$L_s = \sum_{t=1}^T \sum_b \left\| \hat{B}_{t,b} - \hat{B}_{t,C(b)} \right\|_2^2 \quad (4.5)$$

where $\hat{B}_{t,b}$ is the estimated bone length for a right-side bone b and $C(b)$ is the corresponding left-side bone.

Training strategy. In our implementation, we first train the network prior to the pose refinement layers using the 3D pose loss L_p , which generates consecutive 3D joint locations from input 2D pose sequences. We then train the entire network in an end-to-end manner with the combined loss:

$$L = \lambda_p L_p + \lambda_d L_d + \lambda_s L_s \quad (4.6)$$

where $\lambda_p = 1$, $\lambda_d = 1$ and $\lambda_s = 0.01$. Note that the pose loss L_p and the symmetry loss L_s are applied on all of the 3D pose estimation results, including all intermediate 3D pose predictions and the final refined 3D joint locations. The derivative loss is only applied on the consecutive 3D joint estimation before pose refinement.

4.3 Experiments

4.3.1 Implementation Details

In our experiments, we first feed the input 2D skeletons into a batch normalization layer to keep the consistency of the input data, which are then passed to our proposed hierarchical “local-to-global” network. Specifically, we employ six graph convolutional layers during the bottom-up process, with 3, 2 and 2 layers for the three graph resolutions. For the top-down process, we deploy a per-node fully-connected operation for each stage of the feature concatenation to get the consecutive 3D joint locations and then choose the target frame 3D pose estimation. Finally, we feed the estimation results into a pose refinement network which is composed of two fully-connected layers with 1024 hidden units followed by a ReLU function. For better understanding, detailed diagrams of our network architecture can be found in our supplementary materials.

Protocol #1	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Mehta, 3DV'17 [172] ($T = 1$)	57.5	68.6	59.6	67.3	78.1	82.4	56.9	69.1	100.0	117.5	69.4	68.0	55.2	76.5	61.4	72.9
Pavlakos, CVPR'17 [55] ($T = 1$)	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Zhou, ICCV'17 [58] ($T = 1$)	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.1	66.0	51.4	63.2	55.3	64.9
Martinez, ICCV'17 [173] ($T = 1$)	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Sun, ICCV'17 [173] ($T = 1$)	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
Fang, AAAI'18 [59] ($T = 1$)	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Pavlakos, CVPR'18 [174] ($T = 1$)	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Hossain, ECCV'18 [60] ($T = 5$)	48.4	50.7	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
Lee, ECCV'18 [95] ($T = 3$)	40.2	49.2	47.8	52.6	50.1	75.0	50.2	43.0	55.8	73.9	54.1	55.6	58.2	43.3	43.3	52.8
Liu, TPAMI'19 [97] ($T = 1$)	50.7	60.0	51.1	63.6	59.7	69.3	48.8	52.0	72.7	105.3	58.6	61.0	62.2	45.9	48.7	61.1
Pavlo, arxiv'18 [163] ($T = 9$)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	49.8
Pavlo, arxiv'18 [163] ($T = 1$)	47.1	50.6	49.0	51.8	53.6	61.4	49.4	47.4	59.3	67.4	52.4	49.5	55.3	39.5	42.7	51.8
Ours, ($T = 1$)	46.5	48.8	47.6	50.9	52.9	61.3	48.3	45.8	59.2	64.4	51.2	48.4	53.5	39.2	41.2	50.6
Ours, ($T = 3$)	44.9	48.1	46.1	49.4	50.6	58.4	47.2	44.4	57.1	62.2	49.7	47.2	52.2	38.2	40.8	49.1
Ours, ($T = 7$)	44.6	47.4	45.6	48.8	50.8	59.0	47.2	43.9	57.9	61.9	49.7	46.6	51.3	37.1	39.4	48.8
Protocol #2	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez, ICCV'17 [57] ($T = 1$)	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Sun, ICCV'17 [173] ($T = 1$)	42.1	44.3	45.0	45.4	51.5	53.0	43.2	41.3	59.3	73.3	51.0	44.0	48.0	38.3	44.8	48.3
Fang, AAAI'18 [59] ($T = 1$)	38.2	41.7	43.7	44.9	48.5	55.3	40.2	38.2	54.5	64.4	47.2	44.3	47.3	36.7	41.7	45.7
Pavlakos, CVPR'18 [174] ($T = 1$)	34.7	39.8	41.8	38.6	42.5	47.5	38.0	36.6	50.7	56.8	42.6	39.6	43.9	32.1	36.5	41.8
Hossain, ECCV'18 [60] ($T = 5$)	35.7	39.3	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
Lee, ECCV'18 [95] ($T = 3$)	34.9	35.2	43.2	42.6	46.2	55.0	37.6	38.8	50.9	67.3	48.9	35.2	50.7	31.0	34.6	43.4
Pavlo, arxiv'18 [163] ($T = 1$)	36.0	38.7	38.0	41.7	40.1	45.9	37.1	35.4	46.8	53.4	41.4	36.9	43.1	30.3	34.8	40.0
Ours, ($T = 1$)	36.8	38.7	38.2	41.7	40.7	46.8	37.9	35.6	47.6	51.7	41.3	36.8	42.7	31.0	34.7	40.2
Ours, ($T = 3$)	36.0	38.4	37.6	40.8	39.9	45.2	37.0	35.0	46.0	50.5	40.6	36.5	42.2	30.6	34.5	39.4
Ours, ($T = 7$)	35.7	37.8	36.9	40.7	39.6	45.2	37.4	34.5	46.9	50.1	40.5	36.1	41.0	29.6	33.2	39.0

TABLE 4.1: Quantitative comparisons of Mean Per Joint Position Error (MPJPE) in millimeter between the estimated pose and the ground-truth on Human3.6M under Protocol #1 and Protocol #2, where T denotes the number of input frames used in each method. The best score is marked in **bold**.

We implement our method within the PyTorch framework. For the first training stage described in Section 4.2.7, we train for 60 epochs with a mini-batch size of 256 using the Amsgrad optimizer. The learning rate starts from 0.001, with a shrink factor of 0.95 applied after each epoch and 0.5 after every 10 epochs. For the second stage, we set $\lambda_p = 1$, $\lambda_d = 1$ and $\lambda_s = 0.01$ and train for 20 epochs with the learning rate of 5×10^{-6} . All experiments were conducted on one GeForce GTX 1080 GPU with CUDA 8.0.

4.3.2 Datasets

We evaluate our method on two publicly available datasets: the Human3.6M dataset [6] for 3D human body pose estimation, and STB [7] for 3D hand pose estimation.

Human3.6M. In this research, we follow the evaluation protocols in prior work [57, 60, 95, 97, 163, 174], in which 5 subjects (S1, S5, S6, S7, S8) are used for training and 2 subjects (S9 and S11) are adopted for testing. All camera views are trained with a single model for all actions. We perform 2D pose detections using the cascaded pyramid network (CPN) [175] which is an extension of FPN, as proposed in [163].

FIGURE 4.6: Comparisons of the 3D PCK results with the state-of-the-art methods on STB for 3D hand pose estimation.

FIGURE 4.7: The impact of the pose refinement on the mean error distances of different body parts on Human3.6M.

STB Dataset. We follow the same training and evaluation protocol used in [2, 3], training on 10 sequences and testing on the other two, with the Convolutional Pose Machine [147] used for detecting the 2D joint locations.

MPPII Dataset. Since the 3D pose annotations are not provided in MPPII dataset, we only use its validation subset for qualitative evaluation. The network trained

for Human3.6M is used for evaluation on the MPII validation subset. We visualize some of the examples in Figure 4.8.

4.3.3 Evaluation Metrics

For Human3.6M, we report the mean per joint position error (MPJPE) as the evaluation metric, which calculates the average Euclidean distance of the estimated joints to ground truth after the alignment of the root joint (central hip). This protocol is referred to as protocol #1. In some work, an alternative metric is adopted, where the estimated 3D pose is aligned to the ground truth via a rigid transformation, which is referred as protocol #2. For the STB dataset, we evaluate the 3D hand pose estimation performance with two metrics. The first metric is the area under the curve (AUC) on the percentage of correct keypoints (PCK) score. The second metric is MPJPE, identical to that for 3D body pose estimation.

4.3.4 Comparison with the State-of-the-art

Results on Human3.6M. As shown in Table 4.1, we compare the performance of our approach with previously reported results on Human3.6M, where T represents the number of input frames. For fair comparison, previous methods with different input sequence lengths are listed in this table. Note that [163] reported better results for 3D pose estimation using 243 frames. However, this is not suitable for the online scenarios we focus on, wherein it is not viable to have long sequences of frames as input. From the table, we can see that compared with the state-of-the-art methods with a similar number of input frames, our approach achieves the best performance under all protocols.

Results on STB Dataset. Figure 4.6 shows the comparison with the state-of-the-art methods [2, 3, 27, 40–42, 159, 176] on STB for 3D hand pose estimation. It can be seen that our approach outperforms the state-of-the-art methods over most error thresholds, improving the AUC value to 0.995 in the joint error range between 20mm and 50mm. Note that here we measure the 3D PCK curve of our proposed method with a single-frame model for fair comparison, since most of the previous works focus on estimating 3D pose from a single image.

	1-frame	3-frames	5-frames	7-frames
Human3.6M	50.62	49.08	48.86	48.78
STB	6.95	6.70	6.65	6.61

TABLE 4.2: MPJPE Results (in mm) of our method with different input sequence lengths on Human3.6M and STB.

Method	Error (mm)
Uniform GCN	69.8
Split Temporal Connect.	54.8
Split Temporal & Symmetrical Connect.	54.0
Split Symmetrical Connect. & Physical Connect. w/o temporal connections	51.1
CVAE w/o temporal connections	52.5
Split Temporal & Symmetrical & Physical Connect. (proposed)	49.1

TABLE 4.3: MPJPE Results (in mm) of our method with 3 input frames and different graph convolutional strategies on Human3.6M.

4.3.5 Ablation Studies

Impact of input sequence length. Table 4.2 shows the MPJPE results of our method with different input sequence lengths on the Human3.6M and STB datasets. We can see that with more input frames used for predictions, our proposed method obtained larger gains in both 3D human and hand pose estimation. This is expected since temporal correlations help resolve issues such as depth ambiguity and self-occlusions, which are typically challenging for single-frame 3D pose estimation task. Noticing that the estimation error with $T = 3$ (49.08 mm) is only slightly higher than those with $T = 5$ (48.86 mm) and $T = 7$ (48.78 mm), we fix $T = 3$ in the following experiments to balance between the estimation accuracy and the computational complexity.

Effect of modified graph convolution. To assess the effectiveness of our modified graph convolution for 3D pose estimation, we carry out experiments on Human3.6M with four variants of our method. a) **Uniform GCN**: all nodes in a neighborhood are uniformly treated with a shared filter matrix. b) **Split Temporal Connect.**: neighboring nodes are divided into three classes: time-forward node, time-backward node and other nodes. c) **Split Temporal & Symmetrical Connect.**: neighboring nodes are divided into four classes: time-forward node, time-backward node, symmetrical node and other nodes. d) **Split Symmetrical Connect. & Physical Connect. w/o temporal connections**: We do not link

Method	Error (mm)	Δ
Ours, proposed	49.1	-
w/o last pooling & 1st upsampling layers	52.3	3.2
w/o all pooling & upsampling layers	53.9	4.8

TABLE 4.4: Ablation studies on different components of our network architecture. The evaluation is performed on Human3.6M with the MPJPE metric under Protocol #1.

the temporal edges between frames. The neighboring nodes are divided into symmetrical node and three different physically-connected nodes. (See Section 4.2.4 for detailed definitions.) All the models are with 3 input frames and consistent graph topology for fair comparisons. The results are presented in Table 4.3. It can be seen that the strategy of separating neighboring nodes into three classes (the first variant) with individual kernel weights considerably improves the performance by a large margin (from 69.8 mm to 54.8 mm). Among the multiple ways of partitioning neighboring nodes, our proposed implementation (Split Temporal & Symmetrical & Physical Connect.) achieves the best result (49.1 mm), which indicates the effectiveness of our proposed non-uniform graph convolution that precisely classifies neighboring nodes based on the semantics of the sparse spatial-temporal graph for 3D pose estimation.

We may also be curious about how the CVAE method proposed in the previous chapter performs in this task. For fair comparison, we adopt CVAE based method (CVAE w/o temporal connections) in the ablation study. As we can see in Table 4.3, the CVAE based method can also achieve good results without temporal smoothness (52.5mm), while GCN may slightly performs better as it explicitly takes human prior (skeleton connections) into considerations.

Effect of local-to-global prediction. We examine the advantage of using our proposed local-to-global architecture by successively removing the graph pooling and upsampling layers from our model. As presented in Table 4.4, removing the pooling and upsampling layers leads to 3 mm to 5 mm increase in error, which demonstrates the benefit of leveraging multi-scale features in our proposed framework.

Impact of pose refinement. We also evaluate the impact of the proposed pose refinement. As presented in Figure 4.7, with the pose refinement, the average estimation errors of different body parts as well as the overall mean errors consistently

FIGURE 4.8: Visual results of our proposed method on Human3.6M and STB datasets. First and second rows: Examples from Human3.6M dataset[6]. Third row: Examples from STB dataset[7]. Forth row: Examples from MPII dataset[8].

decrease on Human3.6M [6], which indicates that our proposed pose refinement can further improve the estimation accuracy of 3D joint locations.

4.3.6 Qualitative results.

Figure 4.8 shows some visual results of our method on Human3.6M [6], STB [7] datasets and MPII dataset [8]. We exhibit samples captured from various view-points with serious self-occlusions. The results show that our proposed model can reliably handle the challenging poses with various orientations and complicated pose articulation.

4.4 Runtime Analysis

We estimate the run time during training and testing for different numbers of input video frames with a single 1080TI. The results are shown in the Table 4.5 using Human3.6M dataset. We see that using more video frames can achieve better evaluation accuracy while increasing the computational complexity. Thus,

we suggest to use 3-frame input in our paper to balance between the estimation accuracy and the computational complexity

Method	Train Run time (ms)	Inference Run time (ms)	MPJPE (mm)
Ours (1 frame)	66	15	50.62
Ours (3 frames)	89	20	49.08
Ours (7 frames)	102	24	48.86

TABLE 4.5: Run Time analysis for our proposed 3D pose estimation method.

4.5 Conclusion

In this chapter, we have presented a novel graph-based method for 3D pose estimation from a short sequence of extracted 2D joint locations. To incorporate the domain-specific knowledge of the constructed spatial-temporal graph, we have introduced a non-uniform graph convolutional operation by learning individual kernel weights for functionally-variant neighbors. Moreover, a local-to-global network architecture has also been proposed to effectively capture the representative features at different scales. Experimental results on two benchmark datasets have demonstrated the superior performance of our method for both 3D hand pose estimation and 3D human body pose estimation tasks. We note that the accuracy of 2D pose estimation is significant for the 3D output results, as we mainly focus on 2D to 3D mapping without further correcting the 2D estimation based on the input images. To enhance the 3D estimation performance, we suggest to train a robust 2D estimator with rich real-world images, which is a commonly effective component for most of the 3D pose estimation methods.

Chapter 5

Progressive Joint Propagation for Human Motion Prediction

In Chapter 4, we propose a graph-based method for 3D pose estimation from a sequence of images/2D joint locations. In this chapter, we wish to steer towards a highly-related and interesting task: can we predict future motions given an observed 3D pose sequence?

5.1 Introduction

Human motion prediction aims to forecast a sequence of future dynamics based on an observed series of human poses. It has extensive applications in robotics, computer graphics, healthcare and public safety [77–81], such as human robot interaction [82], autonomous driving [83] and human tracking [84].

Due to the inherent temporal nature of this task, many existing methods [10, 89, 93] resort to recurrent neural networks (RNN) and their variants for temporal modeling. However, simply relying on the temporal coherence is not enough, since the bio-mechanical dynamics of human behavior are extremely complicated, which not only correspond to temporal smoothness, but also highly relate to spatial joint dependencies. To address this issue, previous work attempted to embed the spatial configurations into the modeling space, so as to enhance the validity of the 3D structures. For instance, Li *et al.* [112] relied on a convolutional filter to capture the

FIGURE 5.1: Left: Overview of our proposed network architecture for 3D motion prediction. Right: Progressive decoding strategy.

dependencies across the spatial and temporal domains. The range of such dependencies, however, is strongly limited by the size of the convolutional kernel. Mao *et al.* [11] applied Discrete Cosine Transform (DCT) to encode temporal information and designed a Graph Neural Network (GNN) to model spatial correlations. Although achieving good results, it forgoes the prevailing sequential decoding architecture and thus cannot explicitly leverage context features that may lead to further improvement.

Based on these observations, we aim to efficiently capture long-range spatial-temporal dependencies while also incorporating the advantage of sequential modeling. In particular, motivated by substantial performance gains achieved by transformer-based networks [9, 177–179] in Natural Language Processing (NLP), we propose to apply the transformer architecture to simultaneously model the spatial and temporal dependencies of human motion. A key benefit of the transformer is that it can capture the global dependencies among the input and output sequences with the help of the attention mechanism. Note that instead of directly feeding sequential poses into the network, following [11] we encode the temporal trajectories of each joint into the frequency domain, before transferring these embedded temporal features to the network. In this way, the model essentially works in the trajectory domain while simultaneously drawing global attention among different joints, as well as between the input historical trajectories and the output predictions.

Moreover, we would like to point out that simply using the transformer for motion prediction does not fully exploit the kinematic chains of body skeletons, yet these

are important since they underlie the motions in human behavior. For instance, absolute displacement of a wrist is often mediated by initial movement of the shoulder, followed by the elbow. Inspired by spatial explorations in 3D human pose estimation [95, 113], we propose to exploit the structural configurations by predicting the joint trajectories progressively in a central-to-peripheral manner. More precisely, as depicted in Figure 5.1 (bottom left), we first estimate the future dynamics of the central body as seed points, and then sequentially propagate the joint predictions based on the kinematic connections.

In addition, the typical approach for most encoder-decoder frameworks, when decoding the motion predictions, is to mainly focus on the single source video that is being processed. This may not be the optimal, since partial motions of many actions follow certain types of general patterns (*e.g.* walking feet, waving hands and bending knees), which may appear in multiple videos with similar but not identical context. Thus, we further propose to incorporate a general motion space into the predictions. Specially, inspired by the memory scheme that is widely utilized in Question Answering (QA) [180, 181], we design a memory-based dictionary to store the common actions across different videos in training data. From the dictionary, we can query the historical motions $\mathbf{C}$ and construct the future dynamics $\mathbf{F}$ to guide the predictions, as shown in Figure 5.1 (left).

In summary, our contributions of this chapter are threefold:

- We propose to leverage the transformer-based architecture to simultaneously exploit the spatial correlations and the temporal smoothness of human motion, by treating the sequential joints with the encoded temporal features as the input of the network.
- To further exploit the structural connectivity of human skeletons, we deploy a progressive decoding strategy to predict the future joint dynamics in a central-to-peripheral manner in accordance with the kinematic chains of a human body.
- To incorporate the general motion space for high quality results, we build a memory-based dictionary to guide the predictions, which preserves the correspondences between the historical motion features and the representative future dynamics.

FIGURE 5.2: The architecture of the conventional transformer [9], where the far left side is the transformer-encoder that encodes the input into context features $\mathbf{C}$, while the relatively right side is the decoder that recursively generates the output sequence with the encoded context features.

We conducted comprehensive experiments on two widely-used benchmarks for human motion prediction: the Human3.6M dataset and the CMU-Mocap dataset, and our proposed method improves state-of-the-art performance in both datasets.

5.2 Methodology

5.2.1 Overview

Figure 5.1 gives an overview of our proposed network architecture. Given a series of human motion poses $\mathbf{X}_{1:T} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T]$, where $\mathbf{x}_t$ denotes the pose at frame t , our target is to generate a corresponding prediction $\mathbf{X}_{T+1:T+T_f}$ for the future T_f frames. To achieve this goal, different from most existing work that employ RNN-based architectures to model the temporal information of human motions, we leverage the transformer network to capture the long-range spatial and temporal correlations of human motions with the help of the attention mechanism. Specifically, we apply Discrete Cosine Transform (DCT) to encode the temporal

information of each joint into frequency space and feed the network with the sequential joints with encoded temporal patterns. Additionally, motivated by the inherent structural connectivity of body skeletons, we explicitly stagger the decoding into predefined progressive steps. This is performed in a central-to-peripheral manner according to the kinematic chains, with a total update at the final stage (see Figure 5.1 bottom-left). To create a generalized and full-spectrum human motion space across the different videos in training data, we further introduce a dictionary as an auxiliary decoder to enhance the prediction quality. The whole model is trained in an end-to-end manner with backpropagation. Next, we describe the individual components in detail.

5.2.2 Revisiting Transformer

The transformer architecture is a core pillar underpinning of many state-of-the-art methods in Natural Language Processing (NLP) since [177], showing superior performance compared to conventional RNN-based structures. This is mainly because RNNs have difficulties in modelling long-term dependencies, while the transformer overcomes this limitation by leveraging the global attention mechanism to draw the dependencies between the entire input and output sequences, without regard to their distances. In particular, as shown in Figure 5.2, the transformer employs an attention-based encoder-decoder framework, where the encoder applies self-attention to extract the useful context from the input sequence, and the decoder consecutively produces the prediction based on the global dependencies between the context features and the previous output sequences. To make use of the sequential order, the transformer additionally inserts a “positional encoding” module to the embeddings at the bottom of the encoder and decoder stacks, assigning each dimension of each token with a unique encoded value.

5.2.3 Transformer for Pose Prediction

Motivated by the substantial performance gain induced by the transform architecture in NLP, we propose to solve the pose prediction problem with the help of a transformer-based network. A straightforward way is to take the human pose at each time step as corresponding to a “word” in the machine translation task, and

then predict the pose at the next time step as akin to predicting the next word. However, doing so blindly ignores the spatial dependencies between joints, which have proven to be highly effective in state-of-the-art methods for pose estimation [31, 59, 62, 95, 97, 113] and pose prediction [11, 112].

To tackle this issue and leverage both the spatial and temporal dependencies of human poses, following [11] we encode the temporal information of each joint into frequency space. The advantage of applying DCT is that, by discarding the high-frequencies, the DCT can provide a more compact representation to capture the smoothness of human motion. In our implementation, we first replicate the last pose $\mathbf{x}_T$ for T_f times to generate a temporal sequence of length $T + T_f$, and then compute the DCT coefficients of each joint. In this way, the task becomes that of generating an output sequence $\hat{\mathbf{X}}_{1:T+T_f}$ from an input sequence $\mathbf{X}_{1:T+T_f}$, with our true objective to predict $\mathbf{X}_{T+1:T+T_f}$. We take 18 DCT coefficients for short-term (400 ms) prediction and 30 DCT coefficients for long-term (1s) prediction. Specifically, given the trajectory $\tilde{\mathbf{x}}_j = (\mathbf{x}_{j,1}, \mathbf{x}_{j,2}, \dots, \mathbf{x}_{j,t})$, where j denotes the index of joint ($j = 1, 2, \dots, J$), the corresponding l^{th} DCT coefficient can be computed as:

$$C_{j,l} = \sqrt{\frac{2}{T}} \sum_{t=1}^T x_{j,t} \sqrt{\frac{1}{1 + \delta_{l,1}}} \cos\left(\frac{\pi}{2T}\right)(2t-1)(l-1) \quad (5.1)$$

where $\sigma_{m,n}$ denotes the *Kronecker* delta function :

$$\delta_{m,n} = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases} \quad (5.2)$$

Similarly, the final output can be transformed to temporal domain via the Inverse Discrete Cosine Transform (IDCT).

$$x_{j,t} = \sqrt{\frac{2}{T}} \sum_{l=1}^T C_{j,l} \sqrt{\frac{1}{1 + \delta_{l,1}}} \cos\left(\frac{\pi}{2T}\right)(2t-1)(l-1) \quad (5.3)$$

We then feed the obtained DCT coefficients into the transformer-based network, so as to capture the spatial dependencies with the help of the attention mechanism. A key benefit of this design is that the network essentially works in the trajectory space while simultaneously modeling the spatial dependencies among the input and the output joint sequences. Moreover, thanks to the positional encoding, the joint

index can be explicitly injected into motion features, allowing the network to not only learn the trajectory of each joint, but also incorporates the joint identities into this process. To encourage smoothness between the input and output trajectories, we also apply the residual scheme at the end of the decoding (see Figure 5.1 right).

5.2.4 Progressive Joint Propagation

The conventional transformer decoder works auto-regressively during inference, that is, conditioning each output word on previously generated outputs. In terms of pose prediction, we observe that human motion is naturally propagated sequentially based on the kinematic chains of body skeletons. For instance, a person may initiate movement of the left shoulder, which then drives the movement of the left elbow and eventually that of the left wrist.

Motivated by this, we propose to progressively express the 3D pose predictions in a similar manner. In particular, as shown in Figure 5.1 bottom-left, we treat the central eight joints as the seed joints, and estimate their future motions first, based on their historical motions. Next we sequentially propagate the joint predictions from center to periphery, according to the structural connectivity of the body skeleton. Figure 5.1 (right) depicts the details of the progressive decoding process: we iteratively predict the residual DCT coefficients of the joints, given the encoded context feature $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_J]$ and the auxiliary information $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_J]$ from the dictionary, where J is the number of joints. Mathematically, for the s^{th} progressive decoding, we formulate the computational process as:

$$\mathbf{X}_{in}^{(s)} = [\hat{\mathbf{X}}_p; \mathbf{X}_h^{(s)}], \quad (5.4)$$

$$\hat{\mathbf{X}}_{out}^{(s)} = \mathbf{X}_h^{(s)} + \text{Decoder}(\mathbf{X}_{in}^{(s)}, \mathbf{C}, \mathbf{F}^{(s)}), \quad (5.5)$$

$$\hat{\mathbf{X}}_p = [\hat{\mathbf{X}}_p; \hat{\mathbf{X}}_{out}^{(s)}], \quad (5.6)$$

where $\mathbf{X}_{in}^{(s)}$ denotes the input of the progressive decoder at stage s , which is a combination of the previously predicted joint sequence $\hat{\mathbf{X}}_p$ and the historical motion of the target joints $\mathbf{X}_h^{(s)}$. $\hat{\mathbf{X}}_{out}^{(s)}$ is the output of the s^{th} decoder, which summarizes the historical motions $\mathbf{X}_h^{(s)}$ and the generated residual DCT coefficients. $\mathbf{F}^{(s)}$ refers to the auxiliary information used for stage s , containing the guided future dynamics of the target joints at stage s . Note that each time we generate the estimation of

FIGURE 5.3: Query and reconstruction procedure in memory-based dictionary for joint j .

certain joints $\hat{\mathbf{X}}_{out}$, we merge them into the previously predicted joint sequence $\hat{\mathbf{X}}_p$ for the next iteration. In this way, we ensure that the estimation is propagated along the structural connectivity and the entire body prediction is constructed in the order of the kinematic chains via the progressive joint propagation. To further refine the full-body prediction, we add a total updating stage at the end of the progressive decoder.

5.2.5 Dictionary

One potential limitation of the encoder-decoder framework (*e.g.* the transformer, the RNN-based modeling) is that the decoder mainly focuses on one input sequence that is currently being processed while decoding. However, partial motions of many actions follow certain common patterns (*e.g.* walking feet, bending knees), which may appear in multiple videos with similar but not identical context features. To incorporate this generalized motion prior knowledge for better prediction quality, we design a memory-based dictionary to guide the motion prediction, inspired by the memory scheme [180–182] leveraged to preserve a knowledge base for comprehensive understanding. The dictionary is built to store the full spectrum of correspondences between the observed motion features and the representative future dynamics of each joint across different videos in training data. Note that

although the correspondences are mainly constructed for each joint, the global motion of the full body is also taken into account, due to the self-attention mechanism of the transformer encoder.

We propose to learn this dictionary via query and construction processes. This dictionary $\mathcal{D}$ is defined as:

$$\mathcal{D} = \left\{ \left(\mathbf{D}_j^{key}, \mathbf{D}_j^{value} \right) \mid j = 1, 2, \dots, J \right\}, \quad (5.7)$$

where $\mathbf{D}_j^{key}, \mathbf{D}_j^{value} \in \mathbb{R}^{N \times M}$ are the key and the value matrices respectively for joint j , N is the number of memory cells / clusters for each joint, and M is the dimension of a feature stored in a memory cell. As shown in Figure 5.3, the key matrix is used to score an observed motion query of each memory cell so as to better combine the value elements. Mathematically, given an encoded context motion feature $\mathbf{c}_j \in \mathbb{R}^M$ of joint j , the query process can be written as:

$$\mathbf{q}_j = \text{softmax} \left(\mathbf{D}_j^{key} \mathbf{c}_j \right), \quad (5.8)$$

where $\mathbf{q}_j$ is the query result in the memory network for joint j . Then we define the construction process as

$$\mathbf{f}_j = \left(\mathbf{D}_j^{value} \right)^T \mathbf{q}_j, \quad (5.9)$$

where $\mathbf{f}_j$ is the feature vector constructed by the memory network for joint j , representing the future dynamics summarized from the learned generalized motion space. In our implementation, we set N as 100 and M as 512.

5.2.6 Training Strategy

Since the construction of the dictionary relies on the context features from the transformer encoder, we first train the whole network without the dictionary module. Then we learn the dictionary and finetune the whole model subsequently. For the first stage, we employ the following loss function, which aims to minimize the differences between the predicted sequential poses converted from the DCT

coefficients and the corresponding ground truth:

$$L = \sum_{s=1}^S \lambda_s \sum_{j \in \mathcal{J}_s} \sum_{t=1}^{T+T_f} \mathcal{P}(\hat{x}_{j,t}, x_{j,t}^{GT}) \quad (5.10)$$

Here $\hat{x}_{j,t}$ refers to the prediction of the j^{th} joint in frame t and $x_{j,t}^{GT}$ is the corresponding ground truth. S is the number of the progressive decoding stages, $\mathcal{J}_s$ is the set of joints to be predicted at stage s , λ_s is the weight for each stage, T and T_f represent the length of observed frames and predicted frames respectively. $\mathcal{P}$ is a distance function that uses L_1 loss for joint angle representation and L_2 loss for 3D joint coordinates, both of which are typical representations in motion prediction literature. Following [11], we sum the errors over both future and observed time steps, to provide more signals for the learning process.

Having trained the transformer encoder and decoder, we next learn the dictionary module. Specifically, given the observed motion features $\mathbf{C}$, we seek to query for similar historical motion patterns and produce the auxiliary information $\mathbf{F}$ containing future dynamics for each joint. The auxiliary information is typically concatenated with the features generated from the decoder and sent into the final linear layer of the progressive decoder (see Figure 5.1 right), so as to produce the prediction of each joint. Formally, we train the dictionary by penalizing the difference between the produced joint predictions and the corresponding ground truth:

$$L_d = \sum_{j=1}^J \sum_{t=1}^{T+T_f} \mathcal{P}(\hat{x}_{j,t}, x_{j,t}^{GT}) \quad (5.11)$$

where J is the number of joints.

Finally, we finetune the whole model in an end-to-end manner, with the same loss function (Equation (5.10)) as proposed in the first training stage.

5.3 Experiments

5.3.1 Implementation details

In our experiments, we chose the conventional transformer proposed in [9], with 8 headers and 512 hidden units for each module. Both the encoder and the progressive decoder contain a stack of $K = 4$ identical layers. To accelerate the convergence, we applied the scheduled sampling scheme [183] during training, which randomly replaces part of the previous joint predictions with the ground truth in the input to the progressive decoder. The whole model was implemented within the PyTorch framework. For the first training stage described in Section 5.2.6, we set $\lambda_s = 1$, and trained for 40 epochs with the Adam optimizer [154]. The learning rate started from $5e-4$, with a shrink factor of 0.96 applied every two epochs. For the second stage, we learned the dictionary for 20 epochs with the learning rate of $5e-4$. Finally, the whole network was finetuned in an end-to-end manner, using a relatively small learning rate of $5e-5$. All experiments were conducted on a single NVIDIA Titan V GPU, with a batch size of 128 for both training and evaluation.

5.3.2 Datasets and Evaluation Metrics

We evaluated our method on two publicly available datasets: the Human3.6M dataset [184] and the CMU-Mocap dataset¹ for 3D human motion prediction.

Human3.6M: Following the standard setup in [10, 11, 92, 93, 112], the global rotations, translations and constant joints were excluded from our experiments. We down-sampled each sequence to 25 frames per second and applied the evaluation on subject 5 (S5), as proposed in [10, 11, 112].

CMU-Mocap: To show the generalization ability of our proposed method, we also evaluated our performance on the CMU mocap dataset (CMU-Mocap). Following [11, 112], we selected eight actions for evaluation, including “basketball”, “baseball”, “soccer”, etc. The data processing is the same as for Human3.6M.

¹Available at <http://mocap.cs.cmu.edu/>

TABLE 5.1: Short-term prediction results in Mean Angle Error (MAE) on Human3.6M for all actions. The best result is marked in bold.

	Walking				Eating				Smoking				Discussion				Directions				Greeting			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
zero-velocity [10]	0.39	0.68	0.99	1.15	0.27	0.48	0.73	0.86	0.26	0.48	0.97	0.95	0.31	0.67	0.94	1.04	0.39	0.59	0.79	0.89	0.54	0.89	1.30	1.49
Residual sup. [10]	0.28	0.49	0.72	0.81	0.23	0.39	0.62	0.76	0.33	0.61	1.05	1.15	0.31	0.68	1.01	1.09	0.26	0.47	0.72	0.84	0.75	1.17	1.74	1.83
convSeq2Seq [112]	0.33	0.54	0.68	0.73	0.22	0.36	0.58	0.71	0.26	0.49	0.96	0.92	0.32	0.67	0.94	1.01	0.39	0.60	0.80	0.91	0.51	0.82	1.21	1.38
AGED w/o adv [92]	0.28	0.42	0.66	0.73	0.22	0.35	0.61	0.74	0.30	0.55	0.98	0.99	0.30	0.63	0.97	1.06	0.26	0.46	0.71	0.81	0.61	0.95	1.44	1.61
AGED w/ adv [92]	0.22	0.36	0.55	0.67	0.17	0.28	0.51	0.64	0.27	0.43	0.82	0.84	0.27	0.56	0.76	0.83	0.23	0.39	0.63	0.69	0.56	0.81	1.30	1.46
Imitation [93]	0.21	0.34	0.53	0.59	0.17	0.30	0.52	0.65	0.23	0.44	0.87	0.85	0.23	0.56	0.82	0.91	0.27	0.46	0.81	0.89	0.43	0.75	1.17	1.33
GNN [11]	0.18	0.31	0.49	0.56	0.16	0.29	0.50	0.62	0.22	0.41	0.86	0.80	0.20	0.51	0.77	0.85	0.26	0.45	0.71	0.79	0.36	0.60	0.95	1.13
ours	0.17	0.30	0.51	0.55	0.16	0.29	0.50	0.61	0.21	0.40	0.85	0.78	0.19	0.54	0.89	0.94	0.22	0.39	0.62	0.69	0.34	0.58	0.94	1.12
	Phoning				Posing				Purchase				Sitting				Sitting Down				Taking Photo			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
zero-velocity [10]	0.64	1.21	1.65	1.83	0.28	0.57	1.13	1.37	0.62	0.88	1.19	1.27	0.40	1.63	1.02	1.18	0.39	0.74	1.07	1.19	0.25	0.51	0.79	0.92
Residual sup. [10]	0.23	0.43	0.69	0.82	0.36	0.71	1.22	1.48	0.51	0.97	1.07	1.16	0.41	1.05	1.49	1.63	0.39	0.81	1.40	1.62	0.24	0.51	0.90	1.05
convSeq2Seq [112]	0.59	1.13	1.51	1.65	0.29	0.60	1.12	1.37	0.63	0.91	1.19	1.29	0.39	0.61	1.02	1.18	0.41	0.78	1.16	1.31	0.23	0.49	0.88	1.06
AGED w/o adv [92]	0.23	0.42	0.61	0.79	0.34	0.70	1.19	1.40	0.46	0.89	1.06	1.11	0.46	0.87	1.23	1.51	0.38	0.77	1.18	1.41	0.24	0.52	0.92	1.01
AGED w/ adv [92]	0.19	0.34	0.50	0.68	0.31	0.58	1.21	1.34	0.47	0.78	1.01	1.07	0.41	0.76	1.05	1.19	0.33	0.62	0.98	1.10	0.23	0.48	0.81	0.95
Imitation [93]	0.54	1.05	1.40	1.56	0.27	0.55	1.16	1.41	0.54	0.78	1.07	1.14	0.29	0.48	0.87	1.04	0.34	0.68	1.01	1.14	0.17	0.37	0.60	0.72
GNN [11]	0.53	1.02	1.35	1.48	0.19	0.44	1.01	1.24	0.43	0.65	1.05	1.13	0.29	0.45	0.80	0.97	0.30	0.61	0.90	1.00	0.14	0.34	0.58	0.70
ours	0.46	0.90	1.2	1.37	0.19	0.44	1.07	1.3	0.38	0.64	1.13	1.21	0.27	0.44	0.78	0.96	0.27	0.54	0.88	0.97	0.13	0.33	0.60	0.74
	Waiting				Walking Dog				Walking Together				Average											
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400								
Residual sup. [10]	0.34	0.67	1.22	1.47	0.60	0.98	1.36	1.50	0.33	0.66	0.94	0.99	0.40	0.78	1.07	1.21								
Residual sup. [10]	0.28	0.53	1.02	1.14	0.56	0.91	1.26	1.40	0.31	0.58	0.87	0.91	0.36	0.67	1.02	1.15								
convSeq2Seq [112]	0.30	0.62	1.09	1.30	0.59	1.00	1.32	1.44	0.27	0.52	0.71	0.74	0.38	0.68	1.01	1.13								
AGED w/o adv [92]	0.31	0.64	1.08	1.12	0.51	0.87	1.21	1.33	0.29	0.51	0.72	0.75	0.32	0.62	0.96	1.07								
AGED w/ adv [92]	0.24	0.50	1.02	1.13	0.50	0.81	1.15	1.27	0.23	0.41	0.56	0.62	0.31	0.54	0.85	0.97								
Imitation [93]	0.25	0.53	0.96	1.19	0.46	0.80	1.12	1.31	0.19	0.41	0.61	0.64	0.31	0.57	0.90	1.02								
GNN [11]	0.23	0.50	0.91	1.14	0.46	0.79	1.12	1.29	0.15	0.34	0.52	0.57	0.27	0.51	0.83	0.95								
ours	0.21	0.48	0.86	1.08	0.40	0.75	1.05	1.23	0.14	0.32	0.52	0.55	0.25	0.49	0.83	0.94								

TABLE 5.2: Short-term prediction results in Mean Per Joint Position Error (MPJPE) on Human3.6M for all actions. The best result is marked in bold. A 3D suffix to a method indicates that the method was directly trained on 3D joint positions. Otherwise, the results were obtained by converting the joint angle to 3D positions. The best result is marked in bold.

	Walking				Eating				Smoking				Discussion				Directions				Greeting			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [10]	21.7	38.1	58.9	68.8	15.1	28.6	54.8	67.4	20.8	39.0	66.1	76.1	26.2	51.2	85.8	94.6	27.9	44.8	63.5	78.2	29.3	56.0	110.2	125.6
Residual sup. 3D [10]	23.8	40.4	62.9	70.9	17.6	34.7	71.9	87.7	19.7	36.6	61.8	73.9	31.7	61.3	96.0	103.5	36.5	56.4	81.5	97.3	37.9	74.1	139.0	158.8
convSeq2Seq [112]	21.8	37.5	55.9	63.0	13.3	24.5	48.6	60.0	15.4	25.5	39.3	44.5	23.6	43.6	68.4	74.9	26.7	43.3	59.0	72.4	30.4	58.6	110.0	122.8
convSeq2Seq 3D [112]	17.1	31.2	53.8	61.5	13.7	25.9	52.5	63.3	11.1	21.0	33.4	38.3	18.9	39.3	67.7	75.7	22.0	37.2	59.6	73.4	24.5	46.2	90.0	103.1
GNN [11]	11.1	19.0	32.0	39.1	9.2	19.5	40.3	48.9	9.2	16.6	26.1	29.0	11.3	23.7	41.9	46.6	11.2	23.2	52.7	64.1	14.2	27.7	67.1	82.9
GNN 3D [11]	8.9	15.7	29.2	33.4	8.8	18.9	39.4	47.2	7.8	14.9	25.3	28.7	9.8	22.1	39.6	44.1	12.6	24.4	48.2	58.4	14.5	30.5	74.2	89.0
Ours	9.6	18.0	33.1	39.1	9.1	19.5	40.2	48.8	7.2	14.2	24.7	29.7	9.0	22.9	43.4	51.6	9.3	22.0	51.6	63.2	15.4	30.7	71.8	82.8
Ours 3D	7.9	14.5	29.1	34.5	8.4	18.1	37.4	45.3	6.8	13.2	24.1	27.5	8.3	21.7	43.9	48.0	11.1	22.7	48.0	58.4	13.2	28.0	64.5	77.9
	Phoning				Posing				Purchase				Sitting				Sitting Down				Taking Photo			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [10]	28.7	50.9	88.0	99.7	30.5	59.4	118.7	144.7	33.3	58.2	85.4	93.7	32.6	65.2	113.7	126.2	33.0	64.1	121.7	146	21.2	40.3	72.2	86.2
Residual sup. 3D [10]	25.6	44.4	74.0	84.2	27.9	54.7	131.3	160.8	40.8	71.8	104.2	109.8	34.5	69.9	126.3	141.6	28.6	55.3	101.6	118.9	23.6	47.4	94.0	112.7
convSeq2Seq [112]	22.4	38.4	65.0	75.4	22.4	42.1	87.3	106.1	28.4	53.8	82.1	93.1	24.7	50.0	88.6	100.4	23.9	39.9	74.6	89.8	18.4	32.1	60.3	72.5
convSeq2Seq 3D [112]	17.2	29.7	53.4	61.3	16.1	35.6	86.2	105.6	29.4	54.9	82.2	93.0	19.8	42.4	77.0	88.4	17.1	34.9	66.3	77.7	14.0	27.2	53.8	66.2
GNN [11]	13.5	22.5	45.2	52.4	11.1	27.1	69.4	86.2	20.4	42.8	69.1	78.3	11.7	27.0	55.9	66.9	11.5	25.4	53.9	65.6	8.3	15.8	38.5	49.1
GNN 3D [11]	11.5	20.2	37.9	43.2	9.4	23.9	66.2	82.9	19.6	38.5	64.4	72.2	10.7	24.6	50.6	62.0	11.4	27.6	56.4	67.6	6.8	15.2	38.2	49.6
Ours	12.8	24.1	43.6	51.6	8.4	25.3	68.9	88.7	20.2	43.5	66.2	74.9	10.6	24.8	50.4	59.1	11.5	25.3	53.9	69.2	7.1	14.8	39.2	48.9
Ours 3D	10.8	19.6	37.6	46.8	8.3	22.8	65.6	81.8	18.5	38.1	61.8	69.6	9.5	23.9	49.8	61.8	11.2	29.9	59.8	68.4	6.3	14.5	38.8	49.4
	Waiting				Walking Dog				Walking Together				Average											
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Residual sup. [10]	24.9	50.0	96.5	114.0	53.8	90.9	134.6	156.9	19.7	38.2	62.9	72.3	27.9	51.6	88.9	103.4	20.5	40.5	119.9	141.8	29.8	59.7	115.2	135.5
Residual sup. 3D [10]	29.5	55.0	109.0	129.0	60.5	119.1	160.8	188.3	23.5	38.5	61.0	70.4	30.8	57.9	95.3	110.9	24.9	44.9	129.9	159.9	30.8	60.9	129.9	155.5
convSeq2Seq [112]	24.9	50.2	101.6	120.0	56.4	94.9	136.1	156.3	21.1	38.5	61.0	70.4	24.9	44.9	84.9	97.9	24.9	44.9	119.9	149.9	29.8	59.7	115.2	135.5
convSeq2Seq 3D [112]	17.9	36.5	72.9	90.7	40.6	74.7	116.6	138.7	15.0	29.9	54.9	63.8	19.6	37.8	68.1	80.2	17.9	36.5	72.9	90.7	14.9	27.5	53.5	65.7
GNN [11]	12.1	27.5	67.3	85.6	35.8	63.6	106.7	126.8	11.7	23.5	46.0	53.5	13.5	27.0	54.2	65.0	12.1	27.5	67.3	85.6	14.9	27.5	67.3	85.6
GNN 3D [11]	9.5	15.5	22.0	25.2	10.5	18.5	28.5	35.3	10.5	18.5	28.5	35.3	11.5	20.5	31.0	38.5	9.5	15.5	22.0	25.2	10.5	18.5	28.5	35.3
Ours	9.2	23.6	62.0	79.6	29.9	60.3	104.3	127.2	10.2	22.0	44.5	52.9	11.9	26.1	53.2	64.5	9.2	23.6	62.0	79.6	14.9	27.5	67.3	85.6
Ours 3D	8.4	21.5	53.9	69.8	22.9	50.4	100.8	119.8	8.7	18.3	34.2	44.1	10.7	23.8	50.0	60.2	8.4	21.5	53.9	69.8	14.9	27.5	67.3	85.6

(a) “Walking” in Human3.6M

(b) “Soccer” in CMU-Mocap

FIGURE 5.4: (a) Qualitative comparison of long-term prediction on Human 3.6M dataset. From top to bottom, we show the ground truth, the results of Residual sup.[10], GCN [11] and our method. (b) Qualitative analysis for the impact of the dictionary. We see that adding the dictionary facilitates more descriptive future dynamics.

5.3.3 Comparison with the State-of-the-art Methods

For fair comparison, we report both short-term (10 frames in 400 milliseconds) and long-term predictions (25 frames in 1 second) for the two datasets, given the input of consecutive 10-frame human poses.

Results on Human3.6M: For short term predictions, we evaluated our results under both MAE (Table 5.1) and MPJPE (Table 5.2) protocols, in comparison to state-of-the-art baselines [10, 11, 92, 93, 112]. As previously mentioned, for MPJPE we can either directly use 3D coordinates or convert angles to 3D joint locations. As can be seen, our proposed method consistently outperformed all the state-of-the-art methods on most actions for both MAE and MPJPE protocols. The improvement is more obvious when measuring with the MPJPE metric, for which the best performance was achieved when directly using 3D joint locations during training. Moreover, we would like to point out that a high error in angle space (*e.g.* Phoning Action under MAE protocol) does not necessarily generate worse results in 3D (Phoning under MPJPE protocol). This can be explained by the inherent ambiguity of the angle representation, since two different sets of angles

TABLE 5.4: Long-term prediction of 3D joint positions in MPJPE on Human 3.6M dataset. Our method using 3D coordinates yields the best performance.

	Walking		Eating		Smoking		Discussion		Average	
milliseconds	560	1000	560	1000	560	1000	560	1000	560	1000
Residual sup. [10]	79.4	91.6	82.6	110.8	89.5	122.6	121.9	154.3	93.3	119.8
Residual sup. 3D [10]	73.8	86.7	101.3	119.7	85.0	118.5	120.7	147.6	95.2	118.1
convSeq2Seq [112]	69.2	81.5	71.8	91.4	50.3	85.2	101.0	143.0	73.1	100.3
convSeq2Seq 3D [112]	59.2	71.3	66.5	85.4	42.0	67.9	84.1	116.9	62.9	85.4
GNN [11]	55.0	60.8	68.1	79.5	42.2	70.6	93.8	119.7	64.8	82.6
GNN 3D [11]	42.3	51.3	56.5	68.6	32.3	60.5	70.5	103.5	50.4	71.0
Ours	51.8	58.7	59.3	76.5	40.3	76.8	82.6	107.7	58.5	79.9
Ours 3D	36.8	41.2	58.4	67.9	29.2	58.3	74.0	103.1	49.6	67.6

FIGURE 5.5: Average 3D position error of each predicted frame for all actions on Human 3.6M dataset.

can generate the same 3D human pose. Based on this observation, for the following experiments, we mainly report our results under the MPJPE metric, using the 3D coordinates for training.

Besides the short term predictions, we also compared our results with the state-of-the-art methods [10, 11, 112] in long-term scenarios. For fair comparison, we report our results for 4 main classes used in the previous work, including the “Walking”, “Eating”, “Smoking” and “Discussion” actions under MPJPE evaluation. As shown in Table 5.4, similar to the short-term results, our results surpassed all other state-of-the-art methods, reducing the average errors to 49.6mm in 560ms and 67.6mm in 1000ms predictions when directly training and evaluating with the 3D joint locations.

For qualitative analysis, we provided visual comparisons with the state-of-the-art approaches [10, 11] for long-term (Figure 5.4 (a)) scenario, which further underscored how our method generates more realistic results, matching the ground truth

TABLE 5.5: Influence of the spatial temporal explorations and the progressive-decoding strategy on 4 actions of Human3.6M. For fair comparison, we exclude the dictionary module for all models.

	Walking				Eating				Smoking				Discussion				Average			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
temporal	28.9	47.2	69.7	77.8	17.2	31.6	58.3	69.0	18.8	35.7	65.4	76.1	27.9	53.1	82.0	87.9	23.2	41.9	68.9	77.7
non-progressive spatial-temporal	10.5	17.1	31.9	35.7	10.1	21.2	40.7	47.5	8.6	15.9	26.5	30.4	10.6	24.1	47.5	51.3	9.9	19.5	36.6	41.2
progressive spatial-temporal (proposed)	8.3	15.1	30.3	35.2	8.8	19.3	39.0	46.1	7.1	14.0	24.9	28.1	8.9	22.1	44.3	49.1	8.3	17.6	34.6	39.6

TABLE 5.6: Impact of the propagating directions on Human3.6 M dataset under the MPJPE protocol. The outward direction performs in the proposed central-to-peripheral extension while the inward direction contrastly propagate the predictions from the outside to inside body.

	Walking				Eating				Smoking				Discussion				Average			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Inward Propagation	8.9	15.8	30.5	34.9	8.9	19.5	38.9	46.7	7.6	15.0	25.0	29.2	9.4	23.0	45.3	49.9	8.7	18.3	34.9	40.2
Outward Propagation(proposed)	7.9	14.5	29.1	33.5	8.4	18.1	37.4	45.3	6.8	13.2	24.1	27.5	8.3	21.7	43.9	48.0	7.8	16.8	33.6	38.5

TABLE 5.7: The impact of the dictionary module on Human3.6M dataset under the MJMPE metric.

	Walking				Eating				Smoking				Discussion				Average			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
w/o dictionary	8.3	15.1	30.3	35.2	8.8	19.3	39.0	46.1	7.1	14.0	24.9	28.1	8.9	22.1	44.3	49.1	8.3	17.6	34.6	39.6
w/ dictionary(proposed)	7.9	14.5	29.1	33.5	8.4	18.1	37.4	45.3	6.8	13.2	24.1	27.5	8.3	21.7	43.9	48.0	7.8	16.8	33.6	38.5

better.

Results on CMU-Mocap: Table 5.3 compared the performance of our approach with the previously reported results [10, 11] on the CMU-Mocap dataset. For fair comparison, all methods were directly trained with 3D joint coordinates and evaluated under the MPJPE protocol. It can be seen that compared with the state-of-the-art methods, our approach achieved the best results on average and over most of the action classes.

5.3.4 Ablation Study

Advantages of spatio-temporal correlation & joint propagation. We first quantify the importance of leveraging both spatial and temporal correlations and assess the effectiveness of the progressive-decoding strategy. For fair comparisons, we excluded the dictionary from the model and ablated our proposed method (progressive spatio-temporal) with the following baselines: **1) temporal:** We used the straightforward way of applying the transformer network, which treats a pose at each time step as a “word” in machine translation task and sequentially generates the pose prediction of each frame. Here the temporal information is directly captured by the transformer network without using DCT encoding; **2)**

non-progressive spatio-temporal: We used DCT to encode the temporal information of each joint and fed the sequential joints into the transformer network, so as to capture the spatial and temporal dependencies. Here each joint represented by DCT coefficients is considered as a token in the transformer network. However, the decoder generates the whole body predictions at one step, without progressively producing the results.

As shown in Table 5.5, compared with the temporal baseline, exploiting both the spatial and temporal correlations (non-progressive spatio-temporal) considerably improved the performance by a large margin, reducing the average MPJPE error from 77.7 mm to 41.2 mm in 400ms prediction. This result can be further enhanced by applying our proposed progressive-decoding strategy, dropping the MPJPE error to 39.6mm in 400ms prediction. Moreover, as illustrated in Figure 5.5, the 3D errors accumulated much faster with the temporal model than the spatio-temporal approaches, and the proposed progressive joint propagation consistently outperformed the non-progressive counterpart across all time steps.

Impact of the propagating direction. Despite the overall effectiveness of progressive joint propagation, we wanted to investigate how the propagation direction impacts the results. To address this, we employed the progressive-decoding in two directions: the outward (proposed) direction that propagates from the body center to the periphery, and the inward (opposite) direction that propagates from outside to the center. As shown in Table 5.6, the outward propagation yielded superior performance, indicating the benefit of guiding joint extension with the more stable motion cues from the center body.

Impact of using the dictionary. We examined the impact of the dictionary quantitatively and qualitatively. As presented in Table 5.7, adding the dictionary consistently reduced the 3D errors among the four main action classes on the Human 3.6M dataset, which quantitatively shows the effectiveness of dictionary module. To gain more insight into what the dictionary has learned and how the dictionary enhances the prediction quality, in Figure 5.4 (b), we qualitatively compared our method with or without the dictionary. As can be seen, when adding the dictionary for general motion guidance, we produce more plausible and descriptive future dynamics, such as “smooth running” after kicking a ball when playing

soccer.

5.4 RunTime Analysis

We compare the time latency for predicting 1000ms on CMU dataset under a single TitanX in Table 5.8. (pd = progressive decoder and dict = dictionary.) We see that our model is promising for real-time scenarios with a time latency of 26ms for 1000ms prediction during inference. We also agree there is a tradeoff between effectiveness and efficiency. For example, the model is faster (16ms instead of 26ms) but less effective (96.9mm instead of 93.2mm).

Method	Train Run time (ms)	Inference Run time (ms)	MPJPE (mm)
Ours (w/o pd w/o dict)	51	16	96.9
Ours (w/o dict)	68	23	94.7
Ours (proposed)	77	26	93.2

TABLE 5.8: Run Time analysis for the proposed 3D pose prediction method.

5.5 Conclusion

In this chapter, we presented a novel approach for human motion prediction, which leverages the transformer-based architecture to simultaneously capture the long-range temporal correlations and spatial dependencies. To fully exploit the kinematic chains of body skeletons, we deployed a progressive strategy, which explicitly decomposes the future joint motion predictions into progressive steps, performed in a central-to-peripheral manner according to the structural connectivity. Moreover, in order to further enable a generalized full-spectrum human motion space across all videos in training data, a memory-based dictionary was proposed to provide auxiliary information to enhance the prediction quality. We see that this work achieves good results for motion prediction task, especially under the evaluation metric of minimizing the differences between "ground truth" motions and generated motions. However, one limitation of this work is that it mainly focuses on minimizing the reconstruction error between the ground truth and generated motion sequences, while less considering motion diversity and human-likeness, which

are also significant for realistic generation. To better address this problem, in the next chapter, we would like to introduce a uniform framework for different motion synthesis tasks, which not only cares about the "accuracy" of the generated series, but also targets at producing plausible and realistic results.

Chapter 6

Investigating Unified 3D Human Motion Synthesis Model

In this chapter, we address the more generalized problem of 3D motion synthesis that covers the tasks of motion prediction, completion, interpolation, and spatial-temporal recovery. Since these tasks have different input constraints and various fidelity and diversity requirements, most existing approaches only cater to a specific task or use different architectures to address various tasks. Here we propose a unified framework based on the Conditional Variational Auto-Encoder (CVAE), where we treat any arbitrary input as a masked motion series.

6.1 Introduction

Generating realistic and plausible human body animations with specified actions has been a fundamental task for the animation industry for decades [79, 185]. To synthesize smooth and natural motions, traditional methods [100, 101] rely heavily on dense pose specifications, which are time-consuming and expensive to obtain.

To overcome this, in recent years many data-driven approaches have been researched for processing human pose sequences, focusing mainly on a motion prediction task [10, 89, 90, 186] that aims to generate future actions from a limited number of observed initial frames. They commonly adopt recurrent neural network (RNN) architectures to predict future frames sequentially, with new ones dependent only

FIGURE 6.1: Here we propose a unified framework for various 3D motion synthesis tasks, which can be considered as a generalization of several existing problems such as prediction, completion, interpolation, and spatial-temporal recovery.

on previously generated frames. Although performing well in motion prediction, these approaches are not directly suited for generalizing to other motion synthesis tasks such as motion completion, interpolation, and spatial-temporal recovery, as shown in Figure 6.1, for which both forward and backward dependencies should be exploited.

On the other hand, generative models have been used in recent approaches [12, 13, 91, 104, 105, 111] to synthesize plausible motions. However, they are either restricted to one type of motion synthesis or use different approaches to address various tasks of Figure 6.1. This is mainly because the different motion synthesis tasks have different goals and expectations. For example, the motion prediction task might expect some diversity in the output since a short initial motion can lead to many possible future motions, while the spatial-temporal recovery task might expect a faithful recovery of the ground-truth motion. Moreover, it may be highly desirable for a user to be able to influence the action type of the generated series. Few existing solutions can fulfill all these requirements in a unified motion synthesis model, especially in terms of generalization to the different motion synthesis tasks while taking into account both fidelity and diversity. Furthermore, the ability to flexibly incorporate semantic guidance such as action labels is highly desirable.

This provides the impetus for us to find a unified architecture for various 3D motion synthesis tasks, aimed at generating realistic and meaningful results. To deal with

FIGURE 6.2: Schematic overview of our proposed network architecture.

various input conditions within a single model, we uniformly treat any arbitrary input as a masked motion series. The visible parts are considered as the input condition or constraint, while the masked regions are the sites targeted for automatic generation. To accommodate diverse possible solutions that are consistent with the given observed frames, we propose a framework incorporating a Conditional Variational Auto-Encoder (CVAE) for estimating a latent distribution of the missing regions, from which we can sample and synthesize multiple plausible results. To encourage congruency between estimated and ground truth distributions, we introduce two parallel but linked branches during training. As seen in Figure 6.2, the bottom branch uses ground truth to obtain the prior distribution of the missing regions and rebuild the original motion series. The upper branch, acting as the inference branch, takes the visible constraints to estimate the conditional latent distribution and samples diverse results from this distribution.

In addition to the CVAE-based framework for generalized motion synthesis, we also allow more precise manipulation of the motion styles. In particular, we propose an Action-Adaptive Modulation (AAM) to subsume the given semantics into the generation process. To further enhance the realism and global consistency of the generated series, we introduce a cross-attention mechanism between encoder and decoder features to capitalize on the relationships between input and output poses irrespective of temporal distance.

In summary, the main contributions of this chapter are:

- We propose a unified CVAE-based framework to handle various motion synthesis tasks such as motion prediction, completion, interpolation, and spatial-temporal recovery that have different input constraints and different fidelity and diversity requirements.

- We introduce an Action-Adaptive Modulation (AAM) that is able to control semantic motion styles of the generated series.
- We design a cross-attention mechanism that exploits long-term context information to enhance the realism and global consistency of synthesized sequences.

We conducted quantitative and qualitative assessments on the widely-used Human3.6M and CMU-Mocap datasets. Experiments show that our approach performs better than the existing pose synthesis methods, and is able to generate realistic and plausible motion series conditioned on various input constraints.

6.2 Proposed Method

Given an arbitrarily masked pose sequence $\mathbf{X}_i \in \mathbb{R}^{T \times K}$, where T denotes the length of a sequence and K is the number of parameters describing each pose, our goal is to recover the missing regions $\hat{\mathbf{X}}_u$ and generate the full consecutive series $\hat{\mathbf{X}}_g = \{\mathbf{X}_i, \hat{\mathbf{X}}_u\}$. As stated earlier, this problem can be considered a generalization of the several existing motion synthesis tasks shown in Figure 6.1.

Unlike most existing motion prediction methods [10, 12, 89, 93] that resort to a temporal modeling architecture such as RNN, we instead propose a CVAE-based framework with two parallel branches during training. As seen in Figure 6.2, the bottom branch is a reconstructive path that utilizes the ground truth to get the latent distribution of the missing regions and rebuilds the original motion series from this distribution. Conversely, the upper branch is a generative path that estimates the latent distribution conditioned on the visible constraints, from which to sample and generate plausible results. To further encourage a precise manipulation of the motion styles, we explicitly integrate the action labels into the decoding stage with the help of the proposed Action-Adaptive Modulation (AAM), followed by an action classifier. As the network parameters are adaptive to the action labels, the semantic meaning of the synthesized series can be manipulated accordingly. Finally, in order to synthesize realistic motions with global context consistency, we employ a cross-attention mechanism to exploit the distant relations between the encoder and decoder features. We discuss individual components next.

FIGURE 6.3: Illustration of the encoder and decoder architectures, which facilitate the effective process and consolidation of the features across scales. For clarity, we also depict a building block (bottom dashed box) showing the detailed architecture of each decoding stage.

6.2.1 CVAE-based Statistical Framework

Given the observed regions $\mathbf{X}_i$ in a pose series, we attempt to synthesize a plausible and realistic pose sequence $\hat{\mathbf{X}}_g = \{\mathbf{X}_i, \hat{\mathbf{X}}_u\}$ without losing the prior knowledge of the input key frames. To do so, we utilized a CVAE[156]-based framework, which estimates a parametric distribution of the unseen regions over a latent space, from which we can sample the latent vector $\mathbf{z}_u$ to generate the missing parts $\mathbf{X}_u$. Formally, this involves a variational lower bound of the conditional log-likelihood of the observation:

$$\begin{aligned} \log p(\mathbf{X}_u | \mathbf{X}_i) \geq & -\mathbf{KL}(q_\psi(\mathbf{z}_u | \mathbf{X}_u) || p_\phi(\mathbf{z}_u | \mathbf{X}_i)) \\ & + \mathbb{E}_{q_\psi(\mathbf{z}_u | \mathbf{X}_u)} [\log(p_\theta(\mathbf{X}_u | \mathbf{z}_u, \mathbf{X}_i))], \end{aligned} \quad (6.1)$$

FIGURE 6.4: Action-Adaptive Modulation (AAM) is applied to the normalization layers in the decoding stage.

where $\mathbf{KL}$ is the Kullback-Leibler divergence, $\mathbf{z}_u$ is the sampled latent vector, $q_\psi(\mathbf{z}_u|\mathbf{X}_u)$ is the posterior sampling function, $p_\phi(\mathbf{z}_u|\mathbf{X}_i)$ is the conditional prior, and $p_\theta(\mathbf{X}_u|\mathbf{z}_u, \mathbf{X}_i)$ is the likelihood. Note that the distributions q_ψ , p_ϕ and p_θ can be parameterized by deep neural networks, and we refer readers to the supplementary file for the detailed derivations.

One issue with Eq. (6.1) is that the second term only takes the sampling from $q_\psi(\mathbf{z}_u|\mathbf{X}_u)$ during training, which is encoded by the ground truth of the unseen regions $\mathbf{X}_u$. This may not be optimal for the synthesis during testing, where we only have the visible regions $\mathbf{X}_i$. To mitigate the gap between training and testing, inspired by [156], we modify Eq. (6.1) by setting the sampling from both $q_\psi(\mathbf{z}_u|\mathbf{X}_u)$ and $p_\phi(\mathbf{z}_u|\mathbf{X}_i)$:

$$\begin{aligned} \log(p(\mathbf{X}_u|\mathbf{X}_i)) \geq & \mu \{ -\mathbf{KL}(q_\psi(\mathbf{z}_u|\mathbf{X}_u)||p_\phi(\mathbf{z}_u|\mathbf{X}_i)) \} + \\ & \mu E_{q_\psi(\mathbf{z}_u|\mathbf{X}_u)}[\log(p_\theta(\mathbf{X}_u|\mathbf{z}_u, \mathbf{X}_i))] + \\ & (1 - \mu) E_{p_\phi(\mathbf{z}_u|\mathbf{X}_i)}[\log(p_\theta(\mathbf{X}_u|\mathbf{z}_u, \mathbf{X}_i))]. \end{aligned} \quad (6.2)$$

where $0 \leq \mu \leq 1$ is a tradeoff parameter.

Network Architecture. Figure 6.2 gives an overview of our CVAE-based statistical framework with two parallel paths, each of which consists of an encoder and a decoder and the decoder networks share identical weights. In particular, for the upper path (also the test path), the partially visible pose sequence $\mathbf{X}_i$ is used to infer the latent distribution $p_\phi(\mathbf{z}_u|\mathbf{X}_i)$ of the unseen sites, from which we can sample the latent vector $\mathbf{z}_u$ and generate the plausible full pose series. For the lower path, the latent distribution $q_\psi(\mathbf{z}_u|\mathbf{X}_u)$ is encoded from the ground truth of the unseen regions $\mathbf{X}_u$. By combining the features extracted from the visible regions and the sampled latent vector containing information of the missing parts, the goal of this path is reconstructing the original pose sequence $\mathbf{X}_g = \{\mathbf{X}_i, \mathbf{X}_u\}$. Moreover, to ensure that the synthesized data fit in the training set distribution, both paths are deployed with an adversarial learning network to facilitate high-quality generation.

6.2.2 Action-Adaptive Modulation

When synthesizing human body animations with the given constraints, a typically desired capability would be: *can we manipulate the motion styles of the generated series with certain semantic guidance such as action labels?* To tackle this issue, inspired by [187–189] that adjust the activation in normalization layers for image style transformation, we propose to deploy an Action-Adaptive Modulation (AAM) during the decoding stage (as shown in Figure 6.3), followed by an action classifier to enhance the distinctive action-related features. The detailed design of the AAM can be found in Figure 6.4, where the modulation parameters γ and β of the normalization layers are learned from the given action labels. To further make use of the sequential order, we additionally insert a positional embedding into the action labels, assigning each frame with a unique encoded value.

Specifically, let $\mathbf{h}^i$ be the activations of the i -th layer of a deep convolutional network for a batch of N samples. $\mathbf{L} \in \mathbb{D}^T$ is the semantic label of each sample, where $\mathbb{D}$ is a set of integers denoting the one-hot action labels. T^i is the length of the sequence in the layer, and C^i is the number of channels in the activation map. The modulated action value at site $(n \in N, c \in C^i, t \in T^i)$ is given by:

$$\gamma_{c,t}^i(\text{PE}(\mathbf{L})) \frac{h_{n,c,t}^i - \mu_{n,c}^i}{\sigma_{n,c}^i} + \beta_{c,t}^i(\text{PE}(\mathbf{L})), \quad (6.3)$$

FIGURE 6.5: Network Architecture of the AttenNet in Figure 6.3. We first perform self-attention to the decoder features to capture long-term dependencies. Then we apply cross attention to enhance the consistency between the encoder and decoder features.

where $h_{n,c,t}^i$ is the activation at the site before normalization, PE is the positional embedding function following [9]. $\gamma_{c,t}^i(\text{PE}(\mathbf{L}))$ and $\beta_{c,t}^i(\text{PE}(\mathbf{L}))$ are the learned modulation parameters of the normalization layer. $\mu_{n,c}$ and $\sigma_{n,c}$ are the mean and standard deviation of the activation in channel c of sample n .

6.2.3 Cross-Attention Mechanism

One potential limitation for the current framework is that it mainly relies on the convolutional architecture to propagate local signals progressively through the data. This approach, however, does not fully exploit the long-range dependencies of the distant features, as well as the consistency between the given visible regions and the synthesized poses. To address this issue, inspired by the Transformer network [9] that employs global attention to model long-term dependencies, we apply a cross-attention mechanism between the encoder and decoder features across scales (see AttenNet in Figure 6.3 with its detailed architecture in Figure 6.5). Specifically, we first perform a multi-head self-attention to capture the long-term dependencies

between frames. Then we introduce cross attention between the encoder and decoder features, to enhance the correlations between the input constraints and the output series.

6.2.4 Training

The following losses are used in training.

Distributive Regularization. In Eq. (6.2), the KL divergence term can be interpreted as regularizing the learned conditional prior $p_\phi(\mathbf{z}_u|\mathbf{X}_i)$ to the posterior distribution $q_\psi(\mathbf{z}_u|\mathbf{X}_u)$. In particular, to simplify the computation, we define both $q_\psi(\mathbf{z}_u|\mathbf{X}_u)$ and $p_\phi(\mathbf{z}_u|\mathbf{X}_i)$ as Gaussian distributions, and the loss is given by

$$\mathcal{L}_{KL}^g = -\text{KL}(q_\psi(\mathbf{z}_u|\mathbf{X}_u)||p_\phi(\mathbf{z}_u|\mathbf{X}_i)) \quad (6.4)$$

To enhance smooth training, we further use a traditional VAE [190] to model the missing regions as arising from a smooth Gaussian prior $p(\mathbf{z}_u)$:

$$\mathcal{L}_{KL}^p = -\text{KL}(q_\psi(\mathbf{z}_u|\mathbf{X}_u)||p(\mathbf{z}_u)) \quad (6.5)$$

where $p(\mathbf{z}_u) = \mathcal{N}(0, \sigma^2(m)I)$, which allows for greater latent prior variance with the increasing number of the missing regions m .

Matching Loss. Analogous to the likelihood term in Eq. (6.2), we encourage accurate sequence reconstruction for the reconstructive path (the bottom path in Figure 6.2) with an L1 loss, given by

$$\mathcal{L}_m^r = \|\hat{\mathbf{X}}_g^{rec} - \mathbf{X}_g\|_1 \quad (6.6)$$

where $\hat{\mathbf{X}}_g^{rec}$ is the reconstructed pose series and $\mathbf{X}_g$ is the ground truth of the whole sequence. On the other hand, for the generative path (the upper path in Figure 6.2) we want to accommodate diversity and thus only enforce the appearance matching of the visible regions between the generated sequence $\hat{\mathbf{X}}_g^{gen}$ and the corresponding ground truth $\mathbf{X}_g$, using

$$\mathcal{L}_m^g = \|\mathbf{M} \odot (\hat{\mathbf{X}}_g^{gen} - \mathbf{X}_g)\|_1 \quad (6.7)$$

FIGURE 6.6: Qualitative examples of our proposed model for different motion-related tasks on Human3.6M and CMU-Mocap datasets. Gray poses are the pre-defined input frames (or partial body of some frames), while the red & blue skeleton sequences are the synthesized poses. Note that the input constraints can be flexibly set to arbitrary positions with varying densities.

where $\mathbf{M}$ denotes the visible mask of the input sequence.

Adversarial Loss. To facilitate high-quality generation, we further incorporate two discriminators D_A , D_B to respectively judge whether the generated and reconstructed series fit in with the dataset distribution. Inspired by [191], we apply a mean feature matching loss to enhance the reconstruction accuracy:

$$\mathcal{L}_{ad}^r = \|D_B(\hat{\mathbf{X}}_g^{rec}, \mathbf{L}) - D_B(\mathbf{X}_g, \mathbf{L})\|_2 \quad (6.8)$$

where $D_B(\cdot)$ is the output feature maps of the discriminator D_B and $\mathbf{L}$ is the corresponding action label. For the adversarial loss in the generative path, we adopt LSGAN [116, 192] to improve realism:

$$\mathcal{L}_{ad}^g = [D_A(\hat{\mathbf{X}}_g^{gen}, \mathbf{L}) - 1]^2. \quad (6.9)$$

Classification Loss. A classification loss $\mathcal{L}_c$ is introduced to facilitate the action-based motion synthesis. In particular, the classification network is pre-trained by the ground truth pose series. To train our model, we apply L1 norm to regularize

the classification scores of the generated and reconstructed results:

$$\mathcal{L}_c = \|CL(\hat{\mathbf{X}}_g^{gen}) - CL(\mathbf{X}_g)\|_1 + \|CL(\hat{\mathbf{X}}_g^{rec}) - CL(\mathbf{X}_g)\|_1 \quad (6.10)$$

where $CL(\cdot)$ denotes the output scores of the classification network.

Overall Loss. Combining all the above losses, we obtain the overall loss function $\mathcal{L}$:

$$\begin{aligned} \mathcal{L} = & \lambda_{KL}(\mathcal{L}_{KL}^g + \mathcal{L}_{KL}^p) + \lambda_{ad}(\mathcal{L}_{ad}^r + \mathcal{L}_{ad}^g) \\ & + \lambda_m(\mathcal{L}_m^r + \mathcal{L}_m^g) + \lambda_c \mathcal{L}_c \end{aligned} \quad (6.11)$$

where λ_* are the tradeoff parameters. In our implementation, we set $\lambda_{KL} = 20$, $\lambda_{ad} = 1$, $\lambda_m = 20$, and $\lambda_c = 1$.

6.3 Experiments

6.3.1 Implementation Details

We implemented our method with the PyTorch framework. Following [116, 189], we applied Spectral Norm [193] to stabilize the adversarial training. During training, we set a learning rate of 10^{-4} , with a mini-batch size of 128 samples using the Adam optimizer [154]. The length of the pose series T is set as 128. Four heads are used in cross-attention mechanism. Note that our model can also be applied to data without action labels by removing the Action-Adaptive-Modulation (AAM) and the action classifier from the proposed network, and not using action labels for the discriminator. Please refer to the supplementary for more details of our implementation.

6.3.2 Datasets

We evaluated our method on two publicly available datasets: the Human3.6M dataset [184] and the CMU-Mocap dataset ¹.

¹Available at <http://mocap.cs.cmu.edu/>

FIGURE 6.7: Diverse actions generated from the same initial pose. By setting different semantic guidance, our model generates a variety of semantically meaningful results given the same input frame.

FIGURE 6.8: Visualization example of comparisons on Human 3.6M dataset. From top to bottom, we show the result of HP-GAN [12], CSGN [13], our method and ground truth motions.

Human3.6M. In our experiments, in order to reduce redundant frames and encourage large motion variations, we subsample the video frames to 10 frames per second. The action classes we select are “Direction”, “Sitting”, “Sitting Down”, “Walking”, “Taking Photos”, “Smoking” and “Eating”. Following the standard setting in [3, 57, 97, 113, 194], we take 5 subjects (S1, S5, S6, S7, S8) for training and 2 subjects (S9 and S11) for testing. The global translations and constant joints are excluded from our experiments.

CMU-Mocap. To show the generalization ability of our proposed method, we also evaluated our performance on the CMU mocap dataset (CMU-Mocap). In our experiments, we selected 6 action classes for evaluation, including “Basketball”, “Jumping”, “Washwindow”, etc. We down-sample each sequence to 25 frames per second. The data processing is the same as that for Human3.6M.

TABLE 6.1: Classification Accuracy (%) of different action classes on Human 3.6M using a classifier independently trained with ground truth 3D pose series. Our method with both action guidance and Action-Adaptive Modulation (AAM) considerably improves the classification accuracy by a large margin compared with the other two baselines.

Data	Sitting	Phoning	Walking	Takingphoto	Direction	SittingDown	Smoking	Eating	Average
Ours w/o action	0.28	0.45	0.82	0.10	0.30	0.87	0.32	0.47	0.45
Ours w/ action w/o AAM	0.41	0.62	0.91	0.51	0.63	0.90	0.57	0.66	0.65
Ours w/ action w/ AAM	0.85	0.93	0.99	0.89	0.81	0.99	0.97	0.81	0.91

6.3.3 Evaluation Metrics

We propose two ways to quantitatively evaluate the synthesized samples. Specifically, for completing sequences with pre-defined past and future frames, we assume that one of our 50 synthesized samples will be close to the original ground truth, and select the single sample with the minimum mean per joint position error (MPJPE) of the masked places. Practically, we apply a center mask of 30 frames to each sequence for the completion assessment.

Conversely, for long-range generations with very few initial input frames, since the sequences tend to have diverse possible futures in totally different trends, it is not appropriate to directly compare the synthesized results against the one ground truth. Instead, we mainly focus on whether the results are adequately realistic compared with real data, as well as diverse enough to generate multiple plausible solutions. In particular, extending from the evaluation protocols of the image/video generation problem [111, 116–118], we use three evaluation metrics: the Frechet Inception Distance (**FID**) [119] that measures the statistical distance between the real and synthesized data in the feature space, the Inception Score (**IS**) [120] that analyzes class probabilities for each generated sequence over all classes, and a **Diversity** measure that estimates the feature-based standard deviation of the multiple generated outputs with the same input condition. (Please refer to Section 2.5.4 for more details about the three metrics.)

Apart from the above metrics to evaluate the generation quality, we also want to examine whether the synthesized samples are actually residing in the same motion styles with the given action labels. To this end, we employed the pre-trained ST-GCN network [121] to compute the classification accuracy of each action class. Note that the higher the recognition accuracy, the better the generated samples follow the class-specific motion patterns in the real data.

TABLE 6.2: Quantitative results on Human3.6M with FID / IS / Diversity metrics to evaluate the long-term generation and MPJPE to assess the completion accuracy. The best results are marked in bold.

Method	FID _g (↓)	FID _a (↓)	IS(↑)	Diversity(↑)	MPJPE(↓)
HP-GAN [12]	272.3	365.2	3.48	0.05	—
Two-stage [111]	327.8	453.6	1.33	0.11	154.9
CSGN [13]	136.8	264.5	4.35	0.17	185.2
ours w/o action	88.2	192.3	6.12	0.26	115.6
ours w/ action	62.5	111.3	6.87	0.19	97.3

TABLE 6.3: Quantitative results on CMU-Mocap dataset. The best results are marked in bold.

Method	FID _g (↓)	FID _a (↓)	IS(↑)	Diversity(↑)	MPJPE(↓)
HP-GAN [12]	188.5	214.2	2.36	0.01	—
Two-stage [111]	386.4	426.9	1.02	0.02	162.2
CSGN [13]	146.7	223.5	3.75	0.04	188.3
ours w/o action	86.2	162.3	4.32	0.08	126.7
ours w/ action	84.6	97.7	4.67	0.05	108.6

TABLE 6.4: Impact of the cross attention mechanism on Human3.6M. For fair comparison, action labels are leveraged for both methods.

Method	FID _g (↓)	FID _a (↓)	IS(↑)	Diversity(↑)	MPJPE(↓)
ours w/o cross attention	71.3	120.1	6.43	0.18	104.4
ours w/ cross attention	62.5	111.3	6.87	0.18	97.3

6.3.4 Results of Pose Series Generation

Quantitative Comparisons. We compared our method with the state-of-the-art approaches [12, 13, 111] on both Human3.6M and CMU-Mocap datasets. Since our model can be optionally implemented with semantic guidance, we carried out experiments with two variants of our method: a) **ours w/o actions** that removes the AAM and classifier modules, and does not use action labels for the discriminator; b) **ours w/ action** that leverages the action labels for semantic manipulation. The compared methods include the RNN-based HP-GAN [12], the convolutional graph-based method CSGN [13] that transforms random vectors from Gaussian processes to pose series, and the two-stage generation model [111] that utilizes a single-to-sequence strategy for motion synthesis. The quantitative results, measured by FID_a / FID_g / IS / Diversity / MPJPE for both the generation and

completion tasks, are summarized in Tables 6.2 and 6.3. Since HP-GAN does not support the motion completion task, we only report the completion performance (MPJPE) of the Two-stage and CSGN models.

As seen in the tables, for both the completion and the long-range sequence generation, our method (**ours w/o action**) significantly surpasses the previous leading approaches on both Human3.6M and CMU-Mocap. Moreover, adding the action labels (**ours w/ action**) further improves the generation realism (FID/ IS) and the completion accuracy (MPJPE), since the generated sequences are more likely to follow the motion styles of the given action labels, leading to better consistency with the real data distributions. In addition, we notice that compared with **ours w/ action**, our non-action method (**ours w/o action**) achieves higher diversity scores. This is expected, as the generation diversity can be reduced by constraining with a specific motion style. For instance, a standing initial pose may lead to multiple possible future motions, such as walking, sitting, and running. If we specify the future action as walking, the possibilities will be limited accordingly.

We may also be curious about how our model compares with the transformer-based architecture proposed in Chapter 5 for motion prediction task. For better analysis, we also show our motion prediction MPJPE result with the reported results on the existing papers [10] and the one proposed in Chapter 5, following their evaluation settings. Compared with other work, here we do not directly take the ground truth supervision for the upperbranch (which is also the inference path of our motion generation method). The results in Figure 6.9 show that our method is comparable with them.

Qualitative Results. For qualitative analysis, we first provided some visual examples of our non-action method for various motion-related tasks on both Human3.6M and CMU-Mocap datasets. As can be seen in Figure 6.6, our model is able to produce realistic and plausible pose series, matching well with the miscellaneous input constraints.

Figure 6.7 gives a few visual results of our action-guided method with the same initial pose but different action labels. We can see that, by setting different semantic guidance, our method produces plausible future dynamics in totally different but semantically meaningful trends.

FIGURE 6.9: MPJPE comparison of motion prediction task among different methods.

We also provide a visualization example to show how our result provides a higher-quality result compared with other leading approaches. As seen in Figure 6.8, the RNN-based HP-GAN led to a frozen state during long-term prediction, while CSGN occasionally generated unnatural poses due to their random sampling method. Conversely, our method is able to produce plausible motions without freezing during long-range prediction.

6.3.5 Ablation Study

Impact of Action-Adaptive Modulation. To evaluate our proposed Action-Adaptive Modulation (AAM) for motion style manipulation, we conducted action classification experiments with the following baselines: a) **ours w/o action**: our proposed method without using action labels; b) **ours w/ action w/o AAM**: directly taking the concatenation of the action label and the latent features as the input to the decoder without leveraging AAM; c) **ours w/ action w/ AAM**: our proposed method that utilizes AAM for semantic manipulation. The results are presented in Table 6.1. Compared with baseline-a, leveraging the action labels (baseline-b) unsurprisingly improved the classification accuracy from 45% to 65%. Moreover, using our proposed AAM further improved the accuracy to 91%, demonstrating the effectiveness of our proposed AAM for semantic manipulation.

Impact of the cross-attention mechanism. We also examined the impact of the cross-attention mechanism. As shown in Table 6.4, our method with the cross attention mechanism consistently outperformed the version without the cross attention in all the evaluation metrics, clearly demonstrating its effectiveness.

6.4 Runtime Analysis

We also analyse the time complexity of run and inference for our pose synthesis model on Human3.6M dataset with a single TITANX in Table 6.5. Note that we didn't quantitatively compare their performance since the evaluation metrics we used (such as FID/IS) should be compared for the same output frames. We also note that this model takes more time than chapter 5 since the generation sequences are much longer (64 frames vs 25 frames) in the proposed motion synthesis model.

Method	Train Run time (ms)	Inference Run time (ms)
Ours (64 frames output)	97	31
Ours (128 frames output)	123	52

TABLE 6.5: Run Time analysis for the proposed 3D pose synthesis method.

6.5 Conclusion

In this chapter, we have proposed a unified CVAE-based model to handle various 3D motion synthesis tasks. Unlike existing methods, our framework enables automatic motion synthesis with flexible input constraints. To further manipulate the motion style of the generated series, we designed an Action-Adaptive Modulation (AAM) to propagate the semantic guidance through the whole sequence. We also introduced a cross-attention mechanism to improve realism and global consistency. We see that this method provides superior performance compared with the existing motion prediction tasks. To further improve the generation realism, we suggest adding physical constraints for post-processing, which can help reduce some jittering cases of the generated motion series.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we study 3D hand/human pose estimation, 3D motion prediction, and a more generalized 3D motion synthesis. The conclusions of this thesis are summarized in the following four aspects.

Firstly, for 3D hand pose estimation, one target of our thesis is to alleviate the burden of 3D joint labeling. To achieve this, we present a weakly supervised solution, which can leverage unlabeled real-world datasets by adapting fully-annotated synthetic datasets with the aid of low-cost depth images.

Secondly, for video-based 3D pose estimation, motivated by the effectiveness of incorporating spatial dependencies and temporal consistencies, we introduce a novel graph-based method for 3D pose estimation from a short sequence of extracted 2D joint locations. In particular, domain knowledge of the human hand (body) configurations is explicitly incorporated into the graph convolutional operations to meet the specific demand of the 3D pose estimation. Furthermore, we introduce a local-to-global network architecture, which is capable of learning multi-scale features for the graph-based representations.

Besides 3D pose estimation, we also take a further step to the highly-related and important task, 3D motion prediction. To fully exploit both spatial and temporal correlations, we propose a novel transformer-based architecture, with a progressive

decoding strategy to get more robust results and a memory module to store the commonly-used motion patterns.

Finally, we attempt to address the generalized problem of 3D motion synthesis that covers the tasks of motion prediction, completion, interpolation, and spatial-temporal recovery. Since these tasks have different input constraints and various fidelity and diversity requirements, most existing approaches only cater to a specific task or use different architectures to address various tasks. This provides the impetus for us to find a unified architecture for various 3D motion synthesis tasks, aimed at generating realistic and meaningful results. To deal with various input conditions within a single model, we uniformly treat any arbitrary input as a masked motion series, and design a Conditional Variational Auto-Encoder (CVAE) model for estimating a latent distribution of the missing regions.

7.2 Future Work

In this section, we propose two interesting topics to be explored in future research.

7.2.1 Understanding Hand-Object Interactions

In this thesis, we assume that the hand is not interacting with objects when doing pose estimation. However, in real scenarios, hand-object interactions are even more common for many applications. Therefore, it is of great importance to further explore how to get accurate hand-object interactions from RGB/RGB-D cameras.

In this area, Choi et al. [195] collaboratively classifies the grasp types and orientation of the hand from depth images and further constrains a pose space using these estimations. Hasson et al.[61] tried to reconstruct both hand and object mesh together from an RGB image. In the future, we will also investigate how to recognize hand-object interactions and estimate hand/object pose, which will be challenging while having a larger prospect.

7.2.2 Human Action Understanding

As there are more accurate 3D pose estimation work, it is practical to see real-time robust pose tracking in daily applications. This provides a fundamental base for higher-level human action understanding. Can we leverage the accurate pose estimation to understand human intentions, *e.g.*, predicting their future actions? Specifically, future research may combine action recognition, detection, and pose estimation/prediction together, where our methods in this thesis could be adopted in these tasks.

7.2.3 Self-supervised Learning

Recently, there are many self-supervision explorations in computer vision, which can greatly reduce the burden of labeling and provide better solutions for large-scale dataset. In the future, it is worth exploring self-supervised learning in pose-related tasks, as 3D labeling remains one of the biggest challenges in this field. For instance, we can combine the contrastive learning mechanism [196] with our pose estimation methods to further learn a better latent space with large amount of unlabeled data. Furthermore, it may also be interesting to estimate the parametric mesh model (such as SMPL [70] or MANO [197]) to first predict a rough result of 3D human pose, and then gradually refine it by reducing the differences between the projected 2D joints or silhouettes from original images.

Author's Publications

Journal Articles

- **Yujun Cai**, Liuhao Ge, Jianfei Cai, Nadia Magnenat Thalmann and Junsong Yuan. “3D Hand Pose Estimation Using Synthetic Data and Weakly Labeled RGB Images”, in *IEEE Trans. on Pattern Analysis and Machine Intelligence* (T-PAMI 2020).

Conference Proceedings

- **Yujun Cai**, Yiwei Wang, Yiheng Zhu et al. “A Unified 3D Human Motion Synthesis Model via Conditional Variational Auto-Encoder”, in *Proc. Int’l Conf. on Computer Vision* (ICCV 2021).
- **Yujun Cai**, Lin Huang, Yiwei Wang, et al. “Learning Progressive Joint Propagation for Human Motion Prediction”, in *Proc. European Conf. on Computer Vision* (ECCV 2020).
- **Yujun Cai**, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, Nadia Magnenat Thalmann, “Exploiting Spatial-temporal Relationships for 3D Pose Estimation via Graph Convolutional Networks”, in *Proc. Int’l Conf. on Computer Vision* (ICCV 2019).
- **Yujun Cai**, Liuhao Ge, Jianfei Cai, Junsong Yuan, “Weakly-supervised 3D Hand Pose Estimation from Monocular RGB Images”, in *Proc. European Conf. on Computer Vision* (ECCV 2018 **Oral**).
- Yiwei Wang, Wei Wang, **Yujun Cai**, Bryan Hooi, Beng Chin Ooi. “Detecting Implementation Bugs in Graph Convolutional Network based Node

Classifiers”. in *International Symposium on Software Reliability Engineering* (ISSRE 2020)

- Chi Zhang, **Yujun Cai**, Guosheng Lin and Chunhua Shen, “DeepEMD: Few-Shot Image Classification with Differentiable Earth Mover’s Distance and Structured Classifiers”, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2020 **Oral**).
- Yiwei Wang, Wei Wang, Yuxuan Liang, **Yujun Cai**, Juncheng Liu, Bryan Hooi. “NodeAug: Semi-Supervised Node Classification with Data Augmentation.”, in *26th SIGKDD Conference on Knowledge Discovery and Data Mining* (SIGKDD 2020).
- Yiwei Wang, Wei Wang, Yuxuan Liang, **Yujun Cai**, Bryan Hooi. “Progressive Supervision for Node Classification.”, in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (ECML-PKDD 2020).
- Lihao Ge, **Yujun Cai**, Junwu Weng and Junsong Yuan, “Hand PointNet: 3D Hand Pose Estimation using Point Sets”, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR 2018 **Spotlight**).

Bibliography

- [1] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016. [xiii](#), [xiv](#), [15](#), [22](#), [24](#), [28](#), [30](#), [32](#), [36](#), [37](#), [44](#), [45](#), [46](#), [47](#), [48](#), [54](#)
- [2] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *International Conference on Computer Vision*, 2017. [xiii](#), [xiv](#), [10](#), [14](#), [16](#), [19](#), [20](#), [22](#), [24](#), [25](#), [27](#), [28](#), [29](#), [30](#), [32](#), [36](#), [37](#), [38](#), [39](#), [46](#), [47](#), [48](#), [53](#), [54](#), [70](#), [71](#)
- [3] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. *ECCV, Springer*, 12, 2018. [xiv](#), [xvii](#), [10](#), [17](#), [45](#), [50](#), [51](#), [52](#), [53](#), [55](#), [56](#), [70](#), [71](#), [106](#)
- [4] Srinath Sridhar, Franziska Mueller, Michael Zollhoefer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. URL <http://handtracker.mpi-inf.mpg.de/projects/RealtimeH0/>. [xiv](#), [15](#), [36](#), [47](#), [48](#), [51](#), [54](#), [55](#)
- [5] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, 2017. URL <http://handtracker.mpi-inf.mpg.de/projects/OccludedHands/>. [xiv](#), [15](#), [36](#), [47](#), [49](#), [51](#), [54](#), [55](#)
- [6] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 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, 2014. [xv](#), [15](#), [62](#), [69](#), [74](#)
- [7] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. A hand pose tracking benchmark from stereo matching. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 982–986. IEEE, 2017. [xv](#), [62](#), [69](#), [74](#)
- [8] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In

- Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. [xv](#), [16](#), [74](#)
- [9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. [xv](#), [78](#), [80](#), [87](#), [102](#)
 - [10] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017. [xv](#), [12](#), [17](#), [77](#), [87](#), [88](#), [89](#), [90](#), [91](#), [95](#), [98](#), [109](#)
 - [11] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9489–9497, 2019. [xv](#), [13](#), [17](#), [78](#), [82](#), [86](#), [87](#), [88](#), [89](#), [90](#), [91](#)
 - [12] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1418–1427, 2018. [xvi](#), [12](#), [14](#), [96](#), [98](#), [106](#), [108](#)
 - [13] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4394–4402, 2019. [xvi](#), [14](#), [96](#), [106](#), [108](#)
 - [14] James M Rehg and Takeo Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 16–22. IEEE, 1994. [2](#), [9](#), [19](#)
 - [15] Ying Wu and Thomas S Huang. Capturing articulated human hand motion: A divide-and-conquer approach. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 606–611. IEEE, 1999.
 - [16] Ying Wu and Thomas S Huang. View-independent recognition of hand postures. In *cvpr*, page 2088. IEEE, 2000. [2](#), [19](#)
 - [17] Cem Keskin, Furkan Kırac, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *European Conference on Computer Vision*, pages 852–863. Springer, 2012. [2](#), [10](#), [19](#)
 - [18] 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*, volume 1, page 5, 2017. [10](#)

- [19] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12): 2821–2840, 2013.
- [20] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.
- [21] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3593–3601, 2016. [2](#), [10](#), [19](#)
- [22] Shan Lu, Dimitris Metaxas, Dimitris Samaras, and John Oliensis. Using multiple cues for hand tracking and model refinement. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–443. IEEE, 2003. [9](#)
- [23] Björn Stenger, Arasanathan Thayananathan, Philip HS Torr, and Roberto Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1372–1384, 2006. [9](#)
- [24] 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 *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2088–2095. IEEE, 2011. [9](#)
- [25] Robert Wang, Sylvain Paris, and Jovan Popović. 6d hands: markerless hand-tracking for computer aided design. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 549–558. ACM, 2011. [9](#)
- [26] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 2017. [9](#)
- [27] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011. [10](#), [53](#), [71](#)
- [28] 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.

- [29] Hui Liang, Junsong Yuan, Daniel Thalmann, and Zhengyou Zhang. Model-based hand pose estimation via spatial-temporal hand parsing and 3d fingertip localization. *The Visual Computer*, 29(6-8):837–848, 2013. [10](#)
- [30] Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3456–3462. IEEE, 2013. [10](#)
- [31] 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. [17](#), [82](#)
- [32] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proc. European Conf. Comput. Vis.*, 2018.
- [33] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Real-time 3d hand pose estimation with 3d convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4):956–970, 2019. [10](#)
- [34] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015. [10](#), [19](#)
- [35] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)*, 35(4):143, 2016.
- [36] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3325–3333, 2015.
- [37] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Trans. Graph.*, 38(4):49, 2019. [10](#)
- [38] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE Transactions on Multimedia*, 15, 2016. [10](#)
- [39] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Generated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, June 2018. URL

- <https://handtracker.mpi-inf.mpg.de/projects/GANeratedHands/>. 10, 16, 19, 20, 25, 29, 32, 47, 53, 54
- [40] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2018. 10, 16, 19, 20, 32, 46, 53, 55, 71
- [41] Umar Iqbal, Pavlo Molchanov, Thomas Breuel Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proc. Eur. Conf. Comput. Vis.*, pages 118–134, 2018. 16, 48, 53, 54
- [42] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *Applications of Computer Vision (WACV), 2018 IEEE Winter Conference on*, pages 436–445. IEEE, 2018. 32, 53, 71
- [43] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+o: Unified egocentric recognition of 3d hand-object poses and interactions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4511–4520, 2019.
- [44] 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. *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.
- [45] Yujun Cai, Liuhao Ge, Jianfei Cai, Nadia Magnenat-Thalmann, and Junsong Yuan. 3d hand pose estimation using synthetic data and weakly labeled rgb images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 10, 17
- [46] 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. 10
- [47] Endri Dibra, Thomas Wolf, Cengiz Oztireli, and Markus Gross. How to refine 3d hand pose estimation from unlabelled depth data? In *3D Vision (3DV), 2017 International Conference on*, pages 135–144. IEEE, 2017. 10, 21, 35
- [48] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7792–7801, 2019. 10
- [49] Zhe Zhang, Chunyu Wang, Weichao Qiu, Wenhui Qin, and Wenjun Zeng. Adafuse: Adaptive multiview fusion for accurate human pose estimation in the wild. *International Journal of Computer Vision*, pages 1–16, 2020. 10

- [50] 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. [10](#)
- [51] Abdolrahim Kadkhodamohammadi, Afshin Gangi, Michel de Mathelin, and Nicolas Padoy. A multi-view rgb-d approach for human pose estimation in operating rooms. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 363–372. IEEE, 2017. [10](#)
- [52] Timo Von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer Graphics Forum*, volume 36, pages 349–360. Wiley Online Library, 2017. [10](#)
- [53] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. [10](#)
- [54] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014. [10](#), [11](#)
- [55] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017. [10](#), [11](#), [69](#)
- [56] Sungheon Park, Jihye Hwang, and Nojun Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *European Conference on Computer Vision*, pages 156–169. Springer, 2016. [11](#)
- [57] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017. [11](#), [17](#), [69](#), [106](#)
- [58] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *IEEE International Conference on Computer Vision*, 2017. [11](#), [21](#), [23](#), [25](#), [27](#), [39](#), [43](#), [69](#)
- [59] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. [11](#), [17](#), [59](#), [69](#), [82](#)

- [60] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *European Conference on Computer Vision*, pages 69–86. Springer, 2018. [11](#), [59](#), [67](#), [69](#)
- [61] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 11807–11816, 2019. [11](#), [114](#)
- [62] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10833–10842, 2019. [12](#), [60](#), [82](#)
- [63] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1067–1076, 2019. [11](#), [54](#)
- [64] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10965–10974, 2019.
- [65] Christian Zimmermann, D Ceylan, J Yang, B Russell, Max Argus, and Thomas Brox. Freihand: Dataset for markerless capture of hand pose and shape from single rgb images. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.
- [66] Shreyas Hampali, Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Ho-3d: A multi-user, multi-object dataset for joint 3d hand-object pose estimation. *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.
- [67] Xiong Zhang, Qiang Li, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. *Proc. IEEE Int. Conf. Comput. Vis.*, 2019. [11](#), [54](#)
- [68] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *arXiv preprint arXiv:2012.13392*, 2020. [11](#)
- [69] Dominik Kulon, Haoyang Wang, Riza Alp Güler, Michael Bronstein, and Stefanos Zafeiriou. Single image 3d hand reconstruction with mesh convolutions. *Proc. Brit. Mach. Vis. Conf.*, 2019. [12](#)
- [70] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. [12](#), [115](#)
- [71] Jun Kai Vince Tan, Ignas Budvytis, and Roberto Cipolla. Indirect deep structured learning for 3d human body shape and pose prediction. 2017. [12](#)

- [72] Hsiao-Yu Fish Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5242–5252, 2017.
- [73] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 459–468, 2018. [12](#)
- [74] Hao Zhu, Xinxin Zuo, Sen Wang, Xun Cao, and Ruigang Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4491–4500, 2019. [12](#)
- [75] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020. [12](#)
- [76] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5442–5451, 2019. [12](#)
- [77] Ankur Gupta, Julieta Martinez, James J Little, and Robert J Woodham. 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2601–2608, 2014. [12](#), [77](#)
- [78] Hema Koppula and Ashutosh Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *International conference on machine learning*, pages 792–800, 2013.
- [79] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *ACM SIGGRAPH 2008 classes*, pages 1–10. 2008. [95](#)
- [80] Nikolaus F Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):2–2, 2002.
- [81] Wei Zhang Lingyu Duan Tianjiao Li, Jun Liu. hard-net: Hardness-aware discrimination network for 3d early activity prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. [12](#), [77](#)
- [82] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities for reactive robotic response. In *IROS*, page 2071. Tokyo, 2013. [12](#), [77](#)
- [83] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016. [12](#), [77](#)

- [84] Haifeng Gong, Jack Sim, Maxim Likhachev, and Jianbo Shi. Multi-hypothesis motion planning for visual object tracking. In *2011 International Conference on Computer Vision*, pages 619–626. IEEE, 2011. [12](#), [77](#)
- [85] Matthew Brand and Aaron Hertzmann. Style machines. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192, 2000. [12](#)
- [86] Vladimir Pavlovic, James M Rehg, and John MacCormick. Learning switching linear models of human motion. In *Advances in neural information processing systems*, pages 981–987, 2001. [12](#)
- [87] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007. [12](#)
- [88] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pages 458–466. IEEE, 2017. [12](#), [13](#)
- [89] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. [12](#), [13](#), [77](#), [95](#), [98](#)
- [90] Judith Butepage, Michael J Black, Danica Kragic, and Hedvig Kjellstrom. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6158–6166, 2017. [95](#)
- [91] Zhenyi Wang, Ping Yu, Yang Zhao, Ruiyi Zhang, Yufan Zhou, Junsong Yuan, and Changyou Chen. Learning diverse stochastic human-action generators by learning smooth latent transitions. *arXiv preprint arXiv:1912.10150*, 2019. [12](#), [14](#), [96](#)
- [92] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 786–803, 2018. [13](#), [17](#), [87](#), [88](#), [89](#)
- [93] Borui Wang, Ehsan Adeli, Hsu-kuang Chiu, De-An Huang, and Juan Carlos Niebles. Imitation learning for human pose prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7124–7133, 2019. [13](#), [17](#), [77](#), [87](#), [88](#), [89](#), [98](#)
- [94] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5308–5317, 2016. [13](#)

- [95] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135, 2018. [17](#), [69](#), [79](#), [82](#)
- [96] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7144–7153, 2019. [13](#)
- [97] Jun Liu, Henghui Ding, Amir Shahroudy, Ling-Yu Duan, Xudong Jiang, Gang Wang, and Alex Kot Chichung. Feature boosting network for 3d pose estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2019. [60](#), [69](#), [82](#), [106](#)
- [98] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021, 2017.
- [99] Jun Liu, Amir Shahroudy, Gang Wang, Ling-Yu Duan, and Alex C Kot. Skeleton-based online action prediction using scale selection network. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1453–1467, 2019. [13](#)
- [100] Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 491–500, 2002. [13](#), [95](#)
- [101] Maxim Likhachev, Geoffrey J Gordon, and Sebastian Thrun. Ara*: Anytime a* with provable bounds on sub-optimality. *Advances in neural information processing systems*, 16:767–774, 2003. [95](#)
- [102] Jianyuan Min, Yen-Lin Chen, and Jinxiang Chai. Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics (TOG)*, 29(1):1–12, 2009. [13](#)
- [103] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*, pages 474–489. Springer, 2020. [13](#)
- [104] Ping Yu, Yang Zhao, Chunyuan Li, Junsong Yuan, and Changyou Chen. Structure-aware human-action generation. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020. [14](#), [96](#)
- [105] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. A recurrent variational autoencoder for human motion synthesis. In *28th British Machine Vision Conference*, 2017. [96](#)

- [106] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno-Noguer. Human motion prediction via spatio-temporal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7134–7143, 2019.
- [107] Rui Zhao, Hui Su, and Qiang Ji. Bayesian adversarial human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6225–6234, 2020.
- [108] Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Learning character-agnostic motion for motion retargeting in 2d. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. [14](#)
- [109] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020. [14](#)
- [110] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. [14](#)
- [111] Haoye Cai, Chunyan Bai, Yu-Wing Tai, and Chi-Keung Tang. Deep video generation, prediction and completion of human action sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018. [14](#), [18](#), [96](#), [107](#), [108](#)
- [112] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, 2018. [17](#), [77](#), [82](#), [87](#), [88](#), [89](#), [90](#)
- [113] 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 International Conference on Computer Vision*, pages 2272–2281, 2019. [17](#), [79](#), [82](#), [106](#)
- [114] Shijian Lu Meng Hwa Er Alex C. Kot Siyuan Yang, Jun Liu. Collaborative learning of gesture recognition and 3d hand pose estimation with multi-order feature analysis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
- [115] Yao Wang Zhipeng Fan, Jun Liu. Adaptive computationally efficient network for monocular 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. [17](#)
- [116] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019. [18](#), [104](#), [105](#), [107](#)

- [117] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.
- [118] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. [18](#), [107](#)
- [119] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. [18](#), [107](#)
- [120] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. [18](#), [107](#)
- [121] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. [18](#), [60](#), [64](#), [107](#)
- [122] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graph.*, 33(5):169, 2014. [19](#)
- [123] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3786–3793, 2014. [19](#)
- [124] Hui Liang, Junsong Yuan, and Daniel Thalmann. Resolving ambiguous hand pose predictions by exploiting part correlations. *IEEE Trans. Circuits Syst. Video Technol.*, 25(7):1125–1139, 2015.
- [125] Romer Rosales, Vassilis Athitsos, Leonid Sigal, and Stan Sclaroff. 3d hand pose reconstruction using specialized mappings. In *Proc. IEEE Int. Conf. Comput. Vis.*, volume 1, pages 378–385, 2001. [19](#)
- [126] Jun Liu, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):3007–3021, 2018. [19](#)
- [127] Junwu Weng, Chaoqun Weng, Junsong Yuan, and Zicheng Liu. Discriminative spatio-temporal pattern discovery for 3d action recognition. *IEEE Trans. Circuits Syst. Video Technol.*, 2018.

- [128] Junwu Weng, Chaoqun Weng, and Junsong Yuan. Spatio-temporal naive-bayes nearest-neighbor (st-nbnn) for skeleton-based action recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4171–4180, 2017. [19](#)
- [129] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE Trans. Multimedia*, 15(5):1110–1120, 2013. [19](#)
- [130] Yidan Zhou, Jian Lu, Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. Hbe: Hand branch ensemble network for real-time 3d hand pose estimation. In *Proc. Eur. Conf. Comput. Vis.*, pages 501–516, 2018. [19](#)
- [131] Xiaokun Wu, Daniel Finnegan, Eamonn O’Neill, and Yong-Liang Yang. Handmap: Robust hand pose estimation via intermediate dense guidance map supervision. In *Proc. Eur. Conf. Comput. Vis.*, pages 246–262. Springer, Cham, 2018.
- [132] Qi Ye and Tae-Kyun Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. In *Proc. Eur. Conf. Comput. Vis.*, pages 801–817, 2018.
- [133] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Heloir, and Didier Stricker. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *Proc. Int. Conf. 3D Vis.*, pages 110–119, 2018.
- [134] Peiyi Li, Haibin Ling, Xi Li, and Chunyuan Liao. 3d hand pose estimation using randomized decision forest with segmentation index points. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 819–827, 2015.
- [135] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4957–4965, 2016.
- [136] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4663–4672, 2018. [19](#)
- [137] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 824–832, 2015. [19](#)
- [138] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4866–4874, 2017. [19](#)

- [139] Hui Liang, Junsong Yuan, and Daniel Thalmann. Egocentric hand pose estimation and distance recovery in a single rgb image. In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*, pages 1–6. IEEE, 2015. [19](#)
- [140] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2242–2251. IEEE, 2017. [20](#)
- [141] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Domain transfer for 3d pose estimation from color images without manual annotations. In *Proc. Asian Conf. Comput. Vis.*, pages 69–84. Springer, 2018. [20](#)
- [142] Denis Tome, Christopher Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *CVPR 2017 Proceedings*, pages 2500–2509, 2017. [21](#), [24](#), [39](#)
- [143] Hashim Yasin, Umar Iqbal, Bjorn Kruger, Andreas Weber, and Juergen Gall. A dual-source approach for 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.
- [144] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. In *European Conference on Computer Vision*, pages 365–382. Springer, 2016.
- [145] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016. [21](#)
- [146] 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. [21](#), [25](#), [35](#), [43](#), [46](#)
- [147] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. [24](#), [27](#), [38](#), [46](#), [70](#)
- [148] Xiao Sun, Bin Xiao, Shuang Liang, and Yichen Wei. Integral human pose regression. *arXiv preprint arXiv:1711.08229*, 2017. [24](#), [38](#)
- [149] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *CVPR*, volume 2, page 6, 2017. [24](#), [39](#)
- [150] Ross Girshick. Fast r-cnn. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 1440–1448. IEEE, 2015. [25](#), [42](#)

- [151] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016. [25](#), [46](#)
- [152] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456, 2015. [25](#), [46](#)
- [153] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. [25](#)
- [154] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [27](#), [46](#), [87](#), [105](#)
- [155] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 4068–4076. IEEE, 2015. [27](#), [43](#)
- [156] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015. [39](#), [40](#), [99](#), [100](#)
- [157] M Oberweger, P Wohlhart, and V Lepetit. Generalized feedback loop for joint hand-object pose estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. [43](#)
- [158] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand key-point detection in single images using multiview bootstrapping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. [47](#)
- [159] Chen Qian, Xiao Sun, Yichen Wei, Xiaou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1106–1113, 2014. [53](#), [71](#)
- [160] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 9877–9886, 2019. [53](#)
- [161] Xianjie Chen and Alan L Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in neural information processing systems*, pages 1736–1744, 2014. [59](#)
- [162] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3073–3082, 2016. [59](#)

- [163] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. *arXiv preprint arXiv:1811.11742*, 2018. 59, 69, 71
- [164] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017. 59
- [165] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 60
- [166] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 64
- [167] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *arXiv preprint arXiv:1211.0053*, 2012. 60, 63
- [168] 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. 61, 65
- [169] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997. 63
- [170] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016. 64
- [171] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 66
- [172] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 International Conference on 3D Vision (3DV)*, pages 506–516. IEEE, 2017. 69
- [173] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2602–2611, 2017. 69
- [174] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7307–7316, 2018. 69

- [175] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018. 69
- [176] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Gnerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018. 71
- [177] 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. 78, 81
- [178] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf, 2018.
- [179] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 78
- [180] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015. 79, 84
- [181] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406, 2016. 79
- [182] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10685–10694, 2019. 84
- [183] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015. 87
- [184] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 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, 2013. 87, 105
- [185] Brian F Allen and Petros Faloutsos. Evolved controllers for simulated locomotion. In *International Workshop on Motion in Games*, pages 219–230. Springer, 2009. 95

- [186] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *International Conference on Learning Representations*, 2018. [95](#)
- [187] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. [101](#)
- [188] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.
- [189] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020. [101](#), [105](#)
- [190] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. [103](#)
- [191] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. [104](#)
- [192] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. [104](#)
- [193] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. [105](#)
- [194] Haoshu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. *arXiv preprint arXiv:1710.06513*, 2017. [106](#)
- [195] Chiho Choi, Sang Ho Yoon, Chin-Ning Chen, and Karthik Ramani. Robust hand pose estimation during the interaction with an unknown object. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3123–3132, 2017. [114](#)
- [196] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. [115](#)

-
- [197] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):1–17, 2017. [115](#)